Institutions of Participation: A Nested Case Study of Company Participation in the Eclipse Foundation, Community, and Business Ecosystem

Steven M. Muegge
Bachelor of Engineering, McMaster University, 1995
Master of Engineering, Carleton University, 2004

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Management

Carleton University
Ottawa, Ontario

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-87757-9

Our file Notre référence
ISBN: 978-0-494-87757-9

Abstract

The competitive strategies of commercial firms increasingly include participation in *business ecosystems* – networked systems of innovation and production that differ in important ways from markets and organizational hierarchies. A growing body of management scholarship examines technology platforms controlled by for-profit keystone companies; however, less is known about business ecosystems anchored around community-developed technology platforms and not-for-profit foundations. This doctoral thesis is a nested multilevel case study of company participation in the Eclipse Foundation, the Eclipse open source developer community, and the Eclipse vendor-neutral business ecosystem. Through the conceptual lens of the *Institutional Analysis and Design* (IAD) framework, it examines the institutional structures, the benefits to companies from participation, and the institutional characteristics that promote and enable company participation. Contributions include (1) an emergent theory of participation linking a community-developed platform, the individuals of a meritocratic developer community, the companies of a business ecosystem, and a governance foundation through shared institutional structures and resource-dependence relationships, (2) answers to a set of research questions about business ecosystem organization and company participation, and (3) a rich descriptive and explanatory case of a field setting not previously documented in the literature. These contributions collectively provide a deeper understanding of how business ecosystems attract participation; this is relevant to researchers in strategy, organization studies, entrepreneurship and software engineering, as well as top management teams, entrepreneurs, software developers, and policy-makers.

Chapter headings

Table of contents

List of figures

List of tables

1 Introduction

1.1 Motivation for this research

Management scholars, from Schumpeter to Chandler to the present day, have long argued that the business firm is the locus of innovation in the modern economy. Teece (1996, p. 193) writes: "In market economies, the business firm is clearly the leading player in the development and commercialization of new products and processes." Yet increasingly, high-impact innovation is occurring outside of the boundaries of business firms, in systems that practitioners are calling business ecosystems (Moore, 1993, 1996, 1998, 2006; Iansiti & Levien, 2004a, 2004b; Chesbrough & Appleyard, 2007; Iyer & Davenport, 2008; West & Wood, 2008; Isckia, 2009; Li, 2009).

Business ecosystems are a practitioner-driven phenomenon where organizations and individuals within a business ecosystem typically self-identify as an ecosystem, both in their own internal discourse and in the brand identity they convey to others (e.g., Milinkovich, 2008a; Lerner & Schankerman, 2010). Examples of business ecosystems examined by management researchers include the Google ecosystem of content providers, advertisers, innovators and consumers that utilize Google's search and advertising platform (Iyer & Davenport, 2008), the "Wintel" personal computer ecosystem of independent software vendors (ISVs) anchored around software from Microsoft and hardware from Intel (Moore, 2006, pp. 48-58), Cisco's ecosystem of networking technology firms (Li, 2009), the Amazon ecosystem of e-commerce and cloud computing (Isckia, 2009), and the Symbian ecosystem of system integrators, component vendors, ISVs, mobile service providers, enterprise software developers, and others whose business depends on the Symbian mobile operating system (West & Wood, 2008). Companies within a business ecosystem are said to "co-evolve capabilities around a new innovation: they work cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations" (Moore, 1993, p. 76). Ecosystem discourse is particularly evident around information and communication technology (ICT) platforms (Gawer & Cusumano, 2002, 2008; West, 2003; Baldwin & Woodard, 2009) such as personal computers and software, mobile telephony and mobile computing devices, web services delivered over the Internet, and console gaming systems (Moore, 2006; Gawer, 2009b).

Business ecosystems are relevant to management scholarship for at least three reasons. First, as envisioned by practitioners, business ecosystems differ from other established management concepts. They cut across traditional industry boundaries (Moore, 2006), and although they share some qualities with, for example, value chains (Porter, 1985), value nets (Brandenburger & Nalebuff, 1996), value networks (Christensen & Rosenbloom, 1995), value constellations (Vanhaverbeke & Cloodt, 2006), communities of practice (Wenger & Snyder, 2000), business groups (Yiu et al. 2005), network organizations (Powell, 1990), and strategic alliances (Culpan, 2009), they differ from each of these constructs in deep and fundamental ways (Iansiti & Levien, 2004a; Moore, 2006). Second, particularly for ICT businesses, the business ecosystem concept is well entrenched in practice – in the mental models for framing management problems, in metaphors for making sense of a complex world and the vocabulary for conveying that understanding to others, and in the enacted theory that guides managerial decision-making and action. Scholars have long argued that manager mental models, metaphors, sense-making and enacted theory are directly relevant to research programs that endeavour to describe, explain and predict aspects of the manager's world (e.g., Cyert & March, 1963; Weick, 1989, 1995; Senge, 1990; Morgan, 2006). Third, business ecosystems offer a fundamental theoretical and empirical challenge to long-established scholarly views of business strategy that associate competitive advantage with control of resources by firms (Wernerfelt, 1984; Barney, 1991; Peteraf, 1993; Amit & Shoemaker, 1993) or advantageous positioning of firms within an industry or geographical industry cluster (Porter, 1980, 1985, 1990). Chesbrough & Appleyard (2007) associate business ecosystems with what they call *open strategy*, where items that were of central importance in traditional business strategy, such as ownership, entry barriers, switching costs, and intra-industry rivalry are of secondary importance to a new set of items, not yet well understood, that are absent or under-appreciated in traditional business strategy. If, as Teece (2007, 2009) argues, business ecosystems are increasingly the new context for opportunity discovery, business model shaping, and competition, new management theories or significant rethinking of extant theories are needed.

Despite the relevance of business ecosystems to management scholarship, much of the existing literature is practitioner-focused, in the form of either practitioner books (Moore, 1996; Iansiti & Levien, 2004a) or articles in practitioner journals (e.g., Moore, 1993; Iansiti & Levien, 2004b). Researchers have established a small base of descriptive case studies (Iyer & Davenport, 2008; West & Wood, 2008; Isckia, 2009; Li, 2009), and made progress describing the structure and properties of business ecosystems, identifying some competitive strategies and advantages enjoyed by ecosystem participants, and partly explaining how economic production in ecosystems differs from production in markets or firms. The extant literature has emphasized technology platforms controlled by *for-profit keystone companies*, and sought to identify the most lucrative strategies for *platform leaders* (Gawer & Cusumano, 2002) – firms in a privileged position to control the platform and capture attractive profits – or firms aspiring to become platform leaders (Gawer & Cusumano, 2008).

The Eclipse ecosystem – anchored around the Eclipse platform of open source software tools and frameworks – is a prominent example of a *vendor neutral* business ecosystem[1], with *shared governance* so that no particular firm is inherently favoured by the governance rules, and broad *company participation* so that governance is shared among firms as a matter of practice. The keystone organization is a membership-based not-for-profit foundation, and the shared technology platform is community-developed. Despite a small but growing literature about the business ecosystems surrounding vendor-controlled platforms, *little is known about vendor neutral business ecosystems and community-developed platforms, or how these settings attract active participation from organizations and individuals*. These gaps are the central motivations for this research.

[1] "Vendor-neutrality" is built into the opening paragraph of the Eclipse bylaws *(Bylaws of the Eclipse Foundation*, 2003, 2008) examined in chapter 7. The term originates in technical standards for interoperability between equipment produced by different vendors (e.g., Quarterman & Wilhelm, 1992; Rymer, 1993; Moriconi et al. 1997). It is a common concept in the scholarly literature on technical standards, as well as the related practitioner literatures on computer engineering, software, open systems architecture, communication networks, and certification, where a "vendor-neutral standard" refers to a technology or specification that is not controlled by any one dominant vendor. In this context, a "vendor neutral business ecosystem" refers to the properties of the business ecosystem surrounding a vendor neutral platform. Chapter 9 of this document is a detailed study of vendor neutral governance at Eclipse.

This research differs from prior research in at least three important ways[2]: (1) selection of an unusual, interesting, and under-studied field setting, (2) a unit of analysis encompassing a broader context and range of phenomena, and (3) an institutional perspective not previously employed within this area of study.

1.2 Field setting

This research is a nested multilevel case study of company participation in the Eclipse business ecosystem. The Eclipse ecosystem is deeply interconnected with the Eclipse Foundation, the Eclipse technology platform, and the Eclipse open source developer community – a system referred to here as the *Eclipse field setting*. The scope of data collection and analysis includes all four components of the Eclipse field setting in order to examine ecosystem participation within its broader context[3].

Eclipse is one of few business ecosystems to successfully transition a technology platform from the privileged control of a for-profit keystone company to vendor neutral governance and broad community participation (West & O'Mahony, 2008; West, 2010). Formed in 2001 around open source software assets donated by and previously controlled by IBM (West, 2003; Capek et al. 2005; O'Mahony et al. 2005a, 2005b)[4], the Eclipse Foundation presently comprises 171 member organizations, 975 individual committers, and more than 120 active projects; more than eleven thousand individuals have

[2] Section 1.2 elaborates on the Eclipse field setting and the scope of data collection and analysis; sections 1.4 and 1.5 elaborate on the institutional perspective and its role in the research design and method.

[3] The emergent explanation for company participation in the Eclipse ecosystem involves the interaction of all four systems. Chapter 7 examines the interconnectedness of the Eclipse Foundation, platform, developer community, and ecosystem. Chapter 8 examines individual participation in the open source software projects that comprise the Eclipse developer community. Chapter 9 examines the governance structures of the Eclipse Foundation. Chapter 10 examines ecosystem participation by companies.

[4] The *Eclipse Consortium*, originally led by IBM, re-organized in 2004 as the not-for-profit *Eclipse Foundation* (Capek et al. 2005; O'Mahony et al. 2005b). Section 7.2 examines the history of Eclipse.

contributed at least one software patch to an Eclipse project[5]. Eclipse software assets are community-developed open source software that can be freely obtained, used, modified, and redistributed by anyone[6]. Website analytics indicate more than one million Eclipse software downloads from the Eclipse website per month, and Eclipse Foundation staff estimate that between four and six million users employ an Eclipse-based integrated development environment (IDE) for building software[7].

The Eclipse software platform is comprised of modular extensible frameworks for building software and a family of tools and runtimes built on those frameworks (des Rivieres & Wiegand, 2004; see also chapter 7). The most well-known Eclipse tool is the Eclipse Java IDE – often called the dominant IDE for software developed in the Java programming language (Geer, 2005; Goth, 2005). McAffer et al. (2010a, p. 5) describe the Eclipse software platform as follows:

> *Underneath the IDE is a generic tooling platform that supports a wide range of tools for languages and systems, from Java to C to Python to Web technologies to data manipulation and reporting... Under the tooling platform is the Eclipse RCP [Rich Client Platform]. This is a generic platform for running applications. The Eclipse IDE happens to be one such application.*

[5] Metrics on Eclipse membership are from presentations at the Eclipse Annual General Meeting at EclipseCon in Santa Clara California in March 2011. Estimates on Eclipse adoption are from public blog posts written in 2010 in by Eclipse Foundation staff. For-profit *member organizations* pay annual membership dues to the Eclipse Foundation at various membership levels, ranging from US$5k to US$500k. Some membership levels require other commitments, such as a minimum number of active full-time software developers. *Committers* are software developers with access privileges to make changes directly to software assets in the software code repository – they are the core inner circle of the development community that maintains and extends the Eclipse software. Anyone can submit a software "patch", but only a committer can review and commit that contribution to the software repository.

[6] Open source software is discussed in section 3.6 of the literature review, and the Eclipse platform is examined closely in chapter 7. The *Eclipse Public License* (EPL), the primary license for Eclipse software assets, is approved by the *Free Software Foundation* (FSF; http://www.fsf.org) as *free software* compliant with the *Free Software Definition* (FSD; Stallman, 1996a), and by the *Open Source Initiative* (OSI, http://www.opensource.org) as *open source software* complaint with the *Open Source Definition* (OSD; http://www.opensource.org/docs/osd).

[7] An *integrated development environment* (IDE) is a software application used by software developers to build and test software. Its various interconnected tools together simplify the tasks of developing software.

The Eclipse modular plug-in architecture extends deeper. Under the RCP is the Eclipse Equinox runtime platform – a standards-based framework for running component-based software (McAffer et al. 2010b). Through well-defined extension points and application programming interfaces (APIs)[8], software developers can use Eclipse tools to create new plug-in components to extend Eclipse tools and frameworks in new ways.

Eclipse is structured to deliberately encourage companies to incorporate Eclipse software assets into their own in-house software and commercial products (O'Mahony et al. 2005a). Examples of Eclipse-based software applications include the IBM Rational, WebSphere, and Lotus product lines (McAffer et al. 2010a, pp. 7-10) and the Actuate product line of business intelligence software (Clenahan, 2009). The software development tools for creating "apps" on the Android, Blackberry, and Symbian mobile platforms are built on Eclipse (Cloninger, 2010), as is the Maestro software controlling the *Spirit* and *Opportunity* rovers on the NASA *Mars Exploration Rover Mission* (Norris, 2010). Google manages its enterprise software infrastructure – one of the largest in the world – with customized Eclipse-based tools deployed to thousands of engineers (Konigsberg, 2010).

The rationale for selecting Eclipse as the focus for this research is three-fold: Eclipse is *unusual*, it is *interesting* to researchers and practitioners, and its business ecosystem has *not been formally studied* by organization, strategy, or innovation scholars. Selection of Eclipse is consistent with advice in the methods literature to examine cases that are *exceptional* (Siggelkow, 2007) and *exemplary* (Yin, 2003b, p. 13), and where

[8] Application programming interfaces (APIs) are specified rules for one software program to call another.

individuals and organizations have "high experience [with] the phenomena under study" (Pettigrew, 1990). The following paragraphs elaborate on each rationale.

First, Eclipse is one of few business ecosystems to successfully transition a technology platform from company-sponsorship to vendor neutrality and to achieve community participation comparable to the most successful open source communities (West & O'Mahony, 2008). The *Mozilla Foundation* (O'Mahony & Raj, 2007) and the *Symbian Foundation* (West & Wood, 2008) also attempted similar transitions, but neither achieved broad company participation comparable to Eclipse (West & Wood, 2008; West 2010)[9]. In this sense, Eclipse is *unusual.*

Second, Eclipse is perceived as an exemplar and role model by others interested in forming new business ecosystems (West & Wood, 2008; West, 2010). In addition to successfully transitioning to vendor neutrality and achieving high participation, it has also achieved high levels of success according to the metrics commonly cited in the business ecosystems literature (Smith & Milinkovich, 2007)[10], it produces software that is technically advanced and widely used, it has enough history (2001-present) to appear sustainable, and its platform has been successively refactored and repackaged to address

[9] The core Mozilla platform software continues to be developed primarily by employees of Mozilla Corporation. Symbian later restructured back to a vendor-controlled platform (Nokia Corporation press release, November 8 2010), with Nokia recently announcing that Symbian was no longer central to its long-term ecosystem strategy (Nokia Corporation press release, February 11 2011).

[10] Iansiti & Levien (2004a) draw on analogy with biological ecosystems to propose three measures of ecosystem health: (1) *productivity*, (2) *robustness* to external shocks, and (3) *niche creation.* Robustness is not directly observable and difficult to assess from available data; a possible proxy is *diversity of participants,* an observable property conceptually associated with robustness (Norbert et al. 2008). By these dimensions, Eclipse is *productive* with regards to output (more than 120 active projects), large and *diverse* with regards to *participation* (171 member organizations, 975 individual committers, thousands of companies companies embedding Eclipse into products and applications, and millions of users), and active with respect to *niche creation* of new initiatives and industry verticals. Metrics are from presentations at the Eclipse Annual General Meeting at EclipseCon 2011 (March) and the 2010 *Eclipse Community Survey.*

a broader range of applications, from an IDE to a tools platform to a rich client platform (and most recently) to a runtime platform. The Eclipse Foundation deliberately nurtures both an open source development community and a commercial ecosystem, and appears to successfully manage any tensions. For these and other reasons, Eclipse is *interesting* to researchers and practitioners. In *BusinessWeek Online*, West (2010, p. 7) argues that for companies seeking to form new ecosystems, "the best role model is Eclipse."

Third, the Eclipse ecosystem has *not been formally studied* by management scholars in this way. Much is written about Eclipse technology and community, but little scholarly writing examines the Eclipse ecosystem and the relationship between the ecosystem and community[11]. An Eclipse ecosystem case study contributes to the small but growing literature of descriptive and explanatory business ecosystems cases (e.g., Iyer & Davenport, 2008; West & Wood, 2008; Li, 2009; Isckia, 2009).

In addition to this three-fold rationale, there are several practical benefits to studying the Eclipse field setting, including extensive publicly available data, an open culture conducive to scholarly inquiry, and availability of and access to key informants.

Prior research has examined business ecosystems, open source developer communities, open source platforms, and governance foundations in more limited contexts. The expanded scope of this research – with all four components of the Eclipse field setting examined within a single research design – focuses attention on interactions *between* subsystems that were at the boundaries or outside the scope of prior studies.

[11] Section 7.3 reviews the extant literature on Eclipse, comprised of four streams: technical articles about the Eclipse platform and products, studies of the Eclipse open source community, software engineering studies examining Eclipse source code, and IBM's open source strategy and the decision to spin out Eclipse.

1.3 Research questions

There are two broad guiding research questions addressed by this research: (1) What is the Eclipse ecosystem? (2) Why do companies participate?

What is the Eclipse ecosystem? More specifically, what are the institutions that structure participation and interaction between individual and economic actors? How does the Eclipse ecosystem relate to the Eclipse open source developer community, to the Eclipse platform of open source software and complementary assets, and to the Eclipse Foundation? Are there tensions between the community and ecosystem? If so, how are tensions managed and resolved? Can the Eclipse ecosystem, as experienced and understood by practitioners, be specified as an operationalized management construct?

Why do companies participate in the Eclipse Ecosystem? Companies have numerous alternatives for research and development (Chesbrough et al. 2006; Culpan, 2009), yet many organizations choose to be Eclipse members and to contribute to community software development. Broad company participation is implicit in vendor neutrality, for without participation by companies, shared governance would remain *de facto* control by a sponsoring company. What are the motives, incentives, and benefits for company participation? For example, is company participation in Eclipse motivated by *purposeful inflows and outflows* through the porous boundaries of company R&D processes (Chesbrough, 2003a, 2006; Chesbrough et al. 2006), by *user innovation* (von Hippel, 1988, 2005) where Eclipse developers are also the most demanding consumers of their own output and therefore best positioned to innovate, or by the potential benefits of intentional and unintentional knowledge spillovers from *cumulative innovation* (Allen, 1983; Scotchmer, 2006; Fiona & O'Mahony, 2007)? Do companies participate for *pooled R&D* (West & Gallagher, 2006), as a *private-collective* innovation model (von Hippel & von Krogh, 2003) with *selective revealing* (Henkel, 2006), and/or for access to *user-to-user assistance* (Lakhani & von Hippel, 2003; Franke & Shah, 2003)? Can Eclipse be usefully understood as a self-governed *knowledge commons* (Hess & Ostrom, 2007) that motivates collective action without coercive regulation or imposition of exclusive property rights? Is participation motivated by access to *complementary assets* (Teece, 1986) which are otherwise difficult or expensive to obtain? Are there different *categories of participation*, perhaps related to *how* companies participate or to the *extent* of participation? Does the answer to why companies participate differ across categories?

Bringing together this question with the previous one, what is it about the Eclipse ecosystem that enables, promotes, and sustains participation? Which of the various institutional aspects of the socio-technical structure are most important, or is participation driven by the interaction between multiple aspects and categories of company participation and motivation? Are the factors that motivate companies to join the same as or are they different from the factors that keep companies engaged and sustain participation over time?

A full understanding of the answers to these guiding questions would contribute useful theory to several areas of management scholarship, including innovation, open innovation, platforms, modularity and design, interorganizational networks, communities, anti-trust law and competition policy, and the scholarly study of knowledge commons.

1.4 Contribution

The scholarly management literature has examined business ecosystems through the conceptual lenses of *industry structure and technology platforms* (Baldwin & Clark, 2000; Messerschmitt & Szyperski, 2003; Gawer, 2009a; Baldwin & Woodard, 2009; Cusumano, 2010), *systems of open innovation* (Chesbrough & Appleyard, 2007; West & Wood, 2008), *innovation communities* (Lynn et al. 1996; von Burg & Kenny, 2003; Miles et al. 2005; von Hippel, 2005; West & Lakhani, 2008; Snow et al. 2009), and *innovation networks* (Iansiti & Levien, 2004a; Iyer et al. 2006; Weiss & Gangadharan, 2010). These perspectives are complementary; each brings into sharp focus some aspects of the business ecosystem that are unaddressed or under-addressed by other perspectives. The *institutional perspective* employed in this research expands our conceptual apparatus and provides an additional vantage point on the business ecosystem as a set of rules, norms, and enforcement characteristics that structure interactions and participation.

The unique theoretical contributions of this research include the following:

(1) a clear conceptual and operational distinction between the business ecosystem and innovation community constructs, and a systems-level explanation that links the ecosystem and community through resource flows, interdependent institutions, shared governance, and a shared platform with the properties of a *knowledge commons* (Hess & Ostrom, 2003);

(2) extensions to a theory of participation in open source software projects and other community-developed platforms: participation is driven not only by an *architecture of participation* (O'Reilly, 2005a) with high *modularity* and *option value* (Baldwin

& Clark, 2006) and a *transparent and accessible open source community* (West & O'Mahony, 2008), but also by interdependent *institutions of participation* and *keystone actions* that motivate resource flows from economic actors in a business ecosystem to development activities within the developer community;

(3) theoretical insights enriching both the literature on commons governance and the literatures on business ecosystems and community-developed platforms, obtained by joining these literatures through the Institutional Analysis and Design (IAD) framework (Ostrom, 1990, 2005; Ostrom & Hess, 2007);

(4) insights for institutional theories of management (Scott, 2008), and how aspects of an institutional perspective can enrich and complement other management perspectives, such as resource dependence (Pfeffer & Salancik, 1978) and platform modularity (Baldwin & Clark, 2000).

Other contributions from close study of the Eclipse field setting include the following:

(5) an institutional description of the Eclipse ecosystem using the IAD framework;

(6) a list of possible tensions between community and ecosystem stakeholders, and the mechanisms by which the Eclipse Foundation manages and harnesses tensions;

(7) typologies of entry strategies employed by Eclipse member companies to join Eclipse open source software projects, of ways in which companies contribute code to projects, and of ways in which companies benefit from participation;

(8) a rich descriptive and explanatory case study of the Eclipse field setting – a system not previously documented in the scholarly management literature.

1.5 Overview of research design

This research is intended as the foundation of a multidisciplinary mixed-method research program in technology innovation management for the principal researcher, research colleagues, and graduate students, for the next decade.

The research design is a *nested explanatory case study* (Yin, 2003a) of an *exceptional* (Siggelkow, 2007) and *exemplary* case (Yin, 2003b, p. 13). Theory-building is *inductive*, *multilevel*, and *multidisciplinary*. It is *inductive* in the sense of not testing hypotheses or models deduced from prior theory, but rather developing the components of theorizing – tentative propositions and explanations, and discovery of anomalies and new categories (Weick, 1989; 1995) – and building theory to account for observations and patterns in the data (Poole et al. 2000; Locke, 2007; Eisenhardt & Graebner, 2007; Van de Ven, 2007). It is *multilevel* in the sense of data collection and analysis attending to multiple organizational entities residing in nested arrangements (Hitt et al. 2007) – individuals, collectives of individuals, business organizations, and collectives of business organizations. It is *multidisciplinary* in the sense of drawing on several scholarly disciplines, including strategic management, organization theory, institutional economics, and analogy with natural ecology to study and better understand phenomena that cross traditional disciplinary boundaries.

The *Institutional Analysis and Design* (IAD) framework (Ostrom, 1990, 2005; Ostrom & Hess, 2007) is the central organizing framework guiding data collection[12].

[12] The role of prior literature in this study is consistent with recommendations of Yin (2003a) and Eisenhardt (1989). In the early stages of this research, prior literature lightly guided data collection and theorizing by suggesting tentative *a priori* constructs, measures, and associations between constructs; the researcher sought to maintain *theoretical sensitivity* to abandon or revise notions that did not fit the data and emerging theory (Glaser & Strauss, 1967, p. 46; see also Glaser, 1978). Glaser & Strauss (1967, p. 3) write: "Of

IAD is a comparative institutional perspective useful for previous case study research on shared governance and collective action in other domains (Ostrom, 2009; Poteete et al. 2010). It focuses attention on three broad categories of institutional variables: (1) *underlying factors* (the *rules-in-use*, attributes of the *community*, and attributes of the *resource*), (2) the *action arena* (of *actors* and an *action situation*), and (3) *outcomes*.

Insights drawn from three other literatures sharpen the IAD categories by focusing attention more narrowly on *a priori* constructs within each IAD element, and suggesting possible measures of constructs and associations between constructs. The first of those literatures is extant research on *business ecosystems*, rooted in an ecological image of interconnected modern business that focuses attention on identification of ecosystem constituents, the roles available to ecosystem constituents, the relationships between constituents, and the characteristics of the platform. The second literature is a *nexus* of six streams of empirically-grounded management research: (1) innovation, (2) open innovation, (3) platforms, (4) modularity and design, (5) interorganizational networks, and (6) community. Each stream informs some aspect of empirical inquiry into business ecosystems. The third and final literature examines a heuristic argument, prevalent in practitioner discourse, that systems which are most successful at attracting community contributions posses an *architecture of participation* (O'Reilly, 2005a). The practitioner literature on architectures of participation suggests possible links between

course, the researcher does not approach reality as a *tabula rasa*. He must have a perspective that will help him see relevant data and abstract significant categories from his scrutiny of the data." In later stages of the research, the literature – both similar and conflicting – was enfolded back into the study to compare against emergent theory. Eisenhardt (1989, p. 544, emphasis in original) writes: "The juxtaposition of conflicting results forces researchers into a more creative, framebreaking mode of thinking than they might otherwise be able to achieve. The result can be deeper insight into both the emergent theory *and* the conflicting literature, as well as sharpening of the limits to generalizability of the focal research."

constructs, particularly rules-in-use, and the incentives for participation. Chapters 2, 3, and 4, respectively, survey the relevant findings from each of these three literatures.

Governance and participation within the Eclipse field setting are examined at three levels of analysis:

(1) *Projects*, selected by theoretical sampling from the population of Eclipse open source software development projects.

(2) *Polycentric governance* structures, including the Eclipse Foundation board of directors, Eclipse Foundation staff, and cross-project governing councils.

(3) *Companies,* selected by theoretical sampling from the population of companies involved in the selected projects.

Data sources include direct observation of participants and participant communications, extensive archival data, and interviews with individual participant informants at multiple levels of analysis. Data collection included a longitudinal component of direct observation of project and community communication channels for the six-month period from September 2010 to February 2011. The archival component examined records back to the launch of the Eclipse Consortium in November 2001 up to the present day, with close attention to the the eighteen-month period from June 2009 to November 2010. The interview component comprised thirty semi-structured interview sessions with twenty individual informants.

Figure 1 situates the three nested levels of analysis within the top-level case of the entire Eclipse field setting, adapting the visual notation of Yin (2003a, figure 2.5, p. 50).

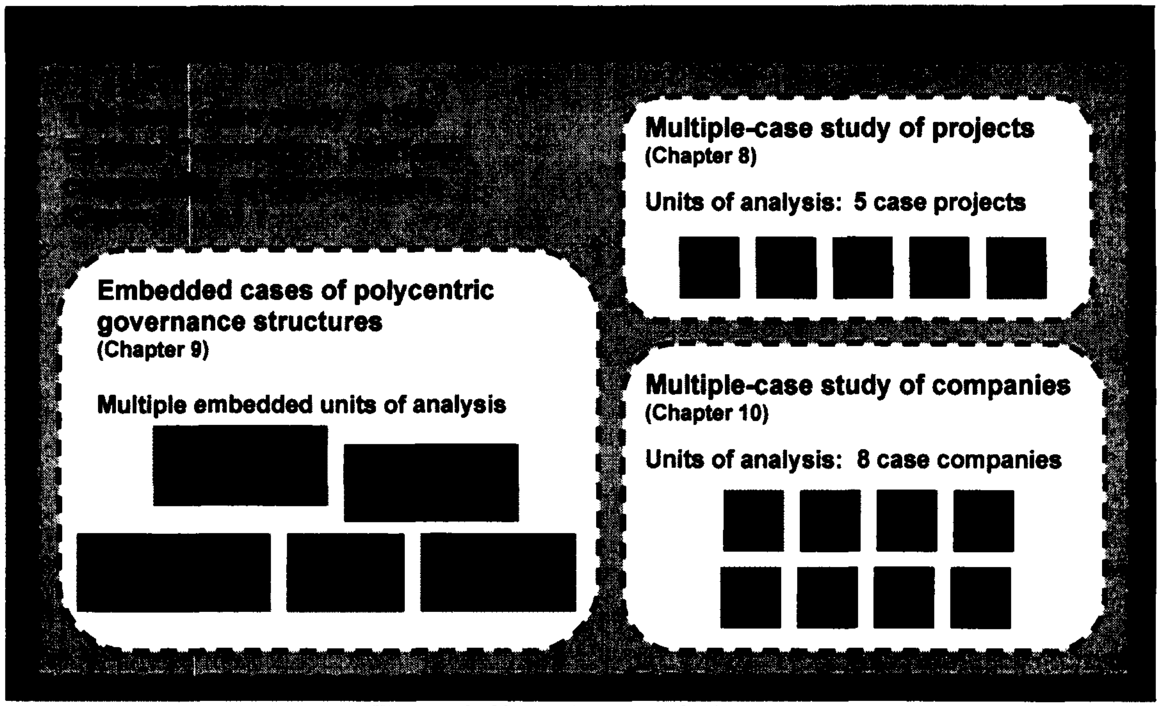

Multiple-case study of projects
(Chapter 8)
Units of analysis: 5 case projects
Embedded cases of polycentric governance structures
(Chapter 9)
Multiple embedded units of analysis
Multiple-case study of companies
(Chapter 10)
Units of analysis: 8 case companies

Figure 1: Nested research design

Through the lens of the IAD framework, each nested level corresponds to a different *action arena* of governance and participation. At the project level, individuals in the Eclipse open source developer community make decisions about day-to-day participation in activities to maintain and extend the Eclipse platform. At the company level, companies make decisions about participation in the Eclipse business ecosystem, membership in the Eclipse Foundation, allocation of development resources to Eclipse projects and the Eclipse development community, and contributions of assets to the Eclipse platform. At the governance level, actions by these same individual and organizational actors, interacting within various interconnected governance structures, collectively shape and reshape the operational rules guiding action at other levels.

By providing "a metatheoretic language … [of] *universal* elements that any relevant theory would need to include" (Ostrom, 2005, p. 28), the IAD framework offers a guide to data collection and a flexible conceptual foundation for multilevel inductive theorizing and constructing emergent theory[13]. It links the project, company, and governance action arenas without assuming or imposing any particular theory of action.

This research design is strongly influenced by the nested design of Kim Clark's breakthrough studies of Japanese manufacturing processes (Lenfle & Baldwin, 2007), Chesbrough's (2003a) seminal work on open innovation (examining innovations spun out from Xerox research labs), the rich descriptive cases from the early years of organization studies (e.g., Selznick, 1949; Blau, 1955; Lipset et al. 1956; Perrow, 1961), and arguments that "Intensive case studies and other forms of qualitative research may be especially useful for understanding multilevel phenomena" (Hitt et al. 2007, p. 1393).

1.6 Organization of this document

The body of this document is organized as twelve chapters, each structured into sections and subsections. The order of major topics follows what Yin (2003a, p. 152) calls a "linear-analytic" structure, beginning with reviews of relevant prior literature, then proceeding to the research method, the findings from the data collected, and then the conclusions. The contents of each chapter are described briefly below.

Chapter 2, "Business ecosystems as metaphor, label, and analogy," distinguishes between three conceptually different ways in which the words "business ecosystem" have been employed together in scholarly writing and practitioner discourse, identifies

[13] Chapter 5 elaborates on the role of the IAD framework in building an emergent theory of participation.

ambiguities and contradictions, and reviews the small management literature that directly addresses business ecosystems as a distinct research topic. The objective is clarity and precision about the business ecosystem construct and its constituent elements.

Chapter 3, "Business ecosystems as a nexus of management literatures," selectively surveys the salient contributions from six empirically-grounded streams of management research: (1) innovation, (2) open innovation, (3) platforms, (4) modularity, (5) interorganizational networks, and (6) communities. Inductive theory-building in later chapters draws on insights from each of these peripheral literatures.

Chapter 4, "Business ecosystems as architectures of participation," surveys arguments from the practitioner literature on the nature of systems that are designed for user participation, and the nascent scholarly literature beginning to build a theory of participation by operationalizing this collection of participation heuristics.

Chapter 5, "Institutional Analysis and Design (IAD)," introduces the IAD framework and situates it within management scholarship. It briefly surveys how IAD has been previously employed to better understand effective governance of collective action around common-pool natural resources, and recent efforts to extend IAD-based research on commons governance into the "new knowledge commons" of scholarly communication, open source software and Internet goods.

Chapter 6, "Research design," specifies the over-all approach and the specific methods for data collection and analysis at the top-level case and each of the three nested levels of data collection and analysis: project, governance, and ecosystem.

The presentation of research results follows a "theory-building" structure (Yin, 2003a, p. 154): each of chapters 7, 8, 9, and 10 develop part of a theoretical argument, and chapter 11 brings these arguments together to a synthetic theoretical explanation of company participation in the Eclipse ecosystem.

Chapter 7, "Eclipse Foundation, platform, community, and ecosystem," introduces the components of the Eclipse field setting and develops the conceptual framework to be refined and sharpened in subsequent sections. It examines the Eclipse platform of open source software and complementary assets, the Eclipse community of individuals who maintain and extend the platform, the Eclipse ecosystem of companies providing commercial offers that utilize, extend, and complement the Eclipse platform, and the Eclipse Foundation. It includes a rigorous and systematic survey of previously published research examining Eclipse.

Chapter 8, "Projects", examines individual participation in Eclipse projects and develops an IAD representation of the Eclipse developer community as an institution.

Chapter 9, "Foundation governance," examines the polycentric governance structures of the Eclipse Foundation and develops an institutional representation of Eclipse governance using the conceptual apparatus of the IAD framework.

Chapter 10, "Company participation," examines company participation in the Eclipse ecosystem and contribution of company resources to the Eclipse community and platform. It develops an IAD representation of the Eclipse business ecosystem, and theorizes about the motivations for company participation and the ways in which the institutional structures of Eclipse relate to those motivations.

Chapters 11 and 12 are discussion and conclusions, respectively.

2 Business ecosystems as metaphor, label, and analogy

The *business ecosystem* is an increasingly prevalent concept in managerial writing and practitioner discourse. However, different individuals conceptualize the ecosystem in different ways and for different purposes, resulting in conceptual differences that are a potential source of confusion and can hamper comparability. This chapter makes those differences explicit and provides a vocabulary for further discussion[14]. It identifies three conceptually different ways in which practitioners and scholars employ the ecosystem concept: (1) as a metaphor for shaping managerial thinking and action, (2) as a label to classify or describe a category of management phenomena, or (3) as an analogy for theory-building. Drawing on management research on language and metaphor, it explores the ambiguities, contradictions, and implications of the ecosystem concept, and develops a conceptual framework of the different ways in which the ecosystem concept is employed in scholarly writing and practitioner discourse. It argues that ecosystem constructs in the literature may differ along at least four dimensions: (1) conceptual space, (2) anchor point, (3) boundaries, and (4) elements.

2.1 Business ecosystems in management discourse

Table 1 summarizes three ways in which the expression "business ecosystem" has been employed in management discourse. The earliest writing on business ecosystems argued that modern business has similarities to ecology, thus an ecological metaphor of management may be able to help managers make better decisions. As these arguments

[14] A previous version of this chapter (Muegge, 2011) was presented to the Technology and Innovation Management division at the 2011 annual meeting of the Academy of Management. This chapter benefits from the constructive comments from three anonymous reviewers and from participants of the session.

became established in management discourse, some practitioners and researchers adopted "ecosystem" as a label to describe management phenomena. In parallel, a body of management scholarship – sometimes overlapping with early writing on ecological metaphors – began applying ecological methods, tools and concepts with varying degrees of rigour to developing management theory and investigating management practice.

Table 1: Business ecosystem as metaphor, label, and analogy

Category	Description	Examples
Metaphor	Argues that an ecological approach to management can help managers make sense of their environment and make better decisions.	Moore (1993, 1996) Iansiti & Levien (2004a, 2004b)
	The business ecosystems metaphor focuses manager attention on coevolution, interconnectedness, shared outcomes, cooperative as well as competitive interactions, networks of relationships, and niches.	
Label	"Business ecosystem" is a label to identify, classify or describe something. There are at least three forms of "something":	
	2a) A synonym for an existing management concept: (i) environment; (ii) supply chain; (iii) network	Tian et al. (2008) West & Wood (2008)
	2b) An organizational environment with specific (ecological) properties such as strong selection effects, rapid change, and constant threats.	Teece (2009)
	2c) A distinct organizational form: a mode of organizing innovation and production that differs from markets and organizational hierarchies.	Moore (2006)
Analogy	Argues that theory developed for biological ecosystems can be a source for inspiration, constructs, explanations, and strong management theory with predictive power and prescriptions for management practice: e.g.,	Peltoniemi (2006)
	• network structures and roles (hub positions; keystones, dominators, specialized niche players; ecosystem niches)	Iansiti & Levien (2004a, 2004b)
	• business ecosystem properties (health, diversity)	Iansiti & Richards (2006)
	• approaches to conducting research and building theory:	
	• network analysis	Sterman (2000, 2001)
	• systems dynamics	Holland (1992)
	• evolutionary models; selection processes	Miller & Page (2007)
	• complex adaptive systems	Norberg et al. (2008)

Each of these three conceptual forms is discussed and critiqued in its own section, following a brief and selective survey of natural ecology and biological ecosystems.

2.2 Natural ecology and biological ecosystems

The field of natural ecology – particularly at the level of analysis of the biological ecosystem – is the *base domain* (Grant & Oswick, 1996) for the business ecosystem metaphor and analogy. This section provides a brief survey of that base domain as an intellectual foundation for discussion of business ecosystems. The main source material is the writing and work of Eugene P. Odum (Odum, 1997; Odum & Barrett, 2005) and Howard T. Odum (Odum, 1971; Brown, 2004), early pioneers in the field of ecology (Golley, 1993). The volume of scholarship on biological ecosystems is vast; only major themes salient to ecosystem metaphors and analogies are discussed here.

The word "ecology" derives from the Greek root "oikos" meaning "household" – thus ecology is the study of the household (Odum, 1997, pp. xiii, 32). More formally, it is the science of the relationships between organisms and their environments (*American Heritage Dictionary of the English Language*, 2009; Odum & Barrett, 2005). The same Greek root is shared with "economics", the management of the household, but ecology differs from economics in currency (energy rather than money), capital (natural rather than economic), resource use (circular rather than linear), system regulation (limited by carrying capacity rather than unlimited exponential expansion), and desired outcomes (sustainability rather than growth).

Odum (1997, p. 38) describes the ecosystem as "a key middle level between you as an individual and the world in which you live." It is not a "catch-all" word for everything, but rather a specific technical term for an organized ecological unit. Odum (1997, p. 43) writes:

> *The ecosystem level is the logical level around which to organize theory and practice in ecology because it is the lowest level in the ecological hierarchy ... that is* complete *– that is, has all the components necessary for function and survival over the long term. For the same reason, we are hearing more about ecosystem management these days – we are moving up from trying to deal with the components separately, one at a time, to managing the system as a whole.*

Ecological historians credit Tansley (1935), a botanist, with first proposing the ecosystem as a higher-level systems concept (Golley, 1993; Odum & Barrett, 2005).

As with organization scholarship, ecology recognizes an extended hierarchy of different levels of analysis (Simon, 1973). Table 2 summarizes an eleven-level framework adapted mainly from Odum & Barrett (2005) with some additional explanation from Odum (1997). There are many alternative variants of this framework (e.g., Odum, 1997, p. 30) sharing similar features.

Table 2: Ecological levels of organization (adapted from Odum & Barrett, 2005)

Level	**Description**
Ecosphere	All living organisms of Earth interacting with the physical environment as a whole.
Biome	A large regional system characterized by an identifying landscape aspect, often a vegetation type – e.g., the Temperate Deciduous Forest biome, or the Continental Shelf Ocean biome. (A biome is distinct from a *region,* which is an area that may contain more than one biome – e.g., the Appalachian Mountains region, or the Pacific Coast region).
Landscape	A heterogeneous area composed of cluster of interacting ecosystems – e.g., a watershed.
Ecosystem	**The biotic community and the abiotic (nonliving) environment functioning together. (The ecosystem is the lowest level of the ecological hierarchy that is "complete" – i.e., with all the components necessary for function and survival over the long term)**
Community	All the populations occupying an area.
Population	Groups of individuals of one kind of organisms – typically an interbreeding species. (In the plural, *populations* can refer to multiple groups linked by common ancestry or habitat, e.g., bird populations or plant populations).
Organism	(The organism is the highest level of the ecological hierarchy exhibiting homeostasis – set-point controls that regulate steady state properties within acceptable limits. Levels above the organism exhibit homeorhesis – a more complex process maintaining pulsing states)
Organ system	
Organ	
Tissue	
Cell	

Hierarchy in ecology is more than a convenient rank-order classification: "It is a holistic approach to understanding and dealing with complex situations, and is an alternative to the reductionist approach of seeking answers by reducing problems to lower-level analysis" (Odum & Barrett, 2005, pp. 6-7). As components combine to produce higher levels of organization, new *emergent properties* appear that were not previously evident at lower levels. Odum (1997) provides two illustrative examples. The properties of water are quite different from those of its gaseous components, oxygen and hydrogen. Likewise, when certain algae and coelenterate animals evolve together to produce a coral, an efficient nutrient cycling mechanism results that enables a coral reef to maintain a high rate of productivity in waters with a low nutrient content.

Ecology draws extensively on cybernetics (Wiener, 1948) and general systems theory (Bertalanffy, 1950, 1968). The organism level (and all biotic levels below the organism) exhibit *homeostasis* – positive and negative feedback loops with set-point controls that regulate steady state properties within acceptable limits. Levels above the organism exhibit *homeorhesis* – a more complex cybernetic process without set-point controls that maintains pulsing states rather than steady states. Ecologists argue that this difference in cybernetic regulatory process is a fundamental qualitative difference between the organism level (and below) and higher levels of the hierarchy. Odum & Barrett (2005, p. 6) warn against "anthropomorphizing" higher levels of the ecological hierarchy as if they were organisms: "Failure to recognize this difference in cybernetics has resulted in much confusion about the realities of the balance of nature." There are close parallels in organizational scholarship: Scott & Davis (2007, pp. 87-106) examine the contribution of cybernetics and general systems theory, and MacKechnie & Donnelly-Cox (1996, p. 51) warn about the dangers of anthropomorphizing organizations.

Ecologists study ecological systems by creating models – a "simplified formulation that mimics a real-world phenomenon so that complex situations can be comprehended and predictions made" (Odum, 1997, p. 38). The standard building blocks are *forcing functions* of outside energy sources or causal forces driving the system, *state variables* of system properties, *flow pathways* showing where energy or material transfers connect state variables and forcing functions, *interaction functions* where forces and properties interact, and *feedback loops* where an output loops back to influence an upstream component (Brown, 2004).

For *biotic* (i.e. living) components, ecological classification is one of function, not species (Odum, 1997, p. 48). For example, the two major biotic components are autotrophs (self-nourishing *producers*) and heterotrophs (other-nourishing *consumers*). Green plants are *autotrophs* that manufacture food from sunlight and simple inorganic substances by the process of photosynthesis. Fungi and animals are *heterotrophs* that utilize, rearrange, and decompose the complex material synthesized by autotrophs. The network of autotrophic and heterotrophic components is called a *food web*. Ecologists further subdivide heterotrophs according to the source of their energy: *herbivores* which feed on plants, *carnivores* or *predators* which feed on other animals, *omnivores* which feed on both plants and animals, and *saprovores* which feed on decaying materials. An *ecological niche* is the ecological role of a species in its community.

The *abiotic* (i.e. non-living) parts of the input environment consists of *energy* *flows*, *material cycles*, and *physical factors*. Energy flows from external sources, through the biotic community and its food web, and out of the ecosystem as the heat, organic matter, and organisms produced in the system. Materials can be used over and over without loss of utility, and in well-ordered ecosystems, many of these materials cycle back and forth between abiotic and biotic components. Physical factors such as climate (e.g., temperature, rainfall, and humidity), the physicochemical nature of soil and water (e.g., salinity and acidity/alkalinity), and the underlying geological substrata determine the kinds of organisms present, how those organisms are organized into communities, and how well they are able to utilize available energy and resources.

Ecologists have developed a precise language for discussing species structure and diversity within an ecosystem community. Odum (1997) writes: "A characteristic and consistent feature of natural communities is that they contain *a comparatively few species that are common* (represented by large numbers of individuals or a large biomass) and a *comparatively large number of species that are rare* at any given place and time" (p. 56). The few common species in a particular community grouping are called *ecological dominants*. Species that exert some kind of controlling influence, whether or not they are dominants, are called *keystone species*. "In the aggregate, rare species have an appreciable impact, and they determine the diversity of the community as a whole. Should conditions become unfavorable for the dominants, rarer species adapted to or tolerant of the new conditions may increase in abundance and take over vital functions" (p. 57). Ecosystem "*redundancy* (repetition) and *resilience stability* (the ability recover rapidly from disturbance) are both enhanced by the presence of many different species in the landscape... Whether a high species diversity increases *resistance stability* – the ability of the ecosystem to remain the same (stable) in the face of disturbance – is a question much debated by ecologists" (p. 66; see also Holling, 1973; Peterson et al., 1998; Gunderson, 2000). Recent field evidence suggests that diversity does increase resistance, but the details are complex and research continues to better understand resilience (Tilman et al. 1996; Tilman, 2000; Tilman et al. 2006; Folk et al. 2004, 2006; Fischer et al. 2006; Ives & Carpenter, 2007; Norbert et al. 2008; Levin & Lubchenco, 2008; Hoey & Bellwood, 2009). The introduction of a new ecological role may be stabilizing or destabilizing depending on how species function within a community. In general, regions with a single dominant species – a *monoculture* – are vulnerable to pests and weeds, and require high levels of inputs of energy and materials to maintain, while *polycultures* of multiple species may produce higher yields per unit area. Two components of diversity are (1) *richness* or *variety* (expressed as the *number* of kinds per unit space) and (2) *relative abundance* (apportionment of individuals among kinds). Ecologists use species structure tables, various diversity indices, and dominance-diversity plots to quantitatively and graphically assess and compare ecosystems (Odum, 1997, ch. 3, pp. 43-77) according to different forms of diversity, including species diversity, genetic diversity, and landscape diversity[15].

Community ecology examines the interaction between different populations, and the dynamics of population change over time. The effect that one species may have on the population growth and well-being of another species may be negative (-), positive (+),

[15] The May 11 2000 issue of *Nature* (volume 405, issue number 6783) has a collection of six review articles on the topic of biodiversity (e.g., Tilman, 2000) that provide an accessible introduction to this literature.

or neutral (0). The five most important combinations are (1) *competition* (--) where both populations inhibit or have a negative effect on each other, often related to scarcity of resources; (2) *predation* (+-) where the effect is positive for the predator and negative for the prey; (3) *parasitism* (-+) where the effect is negative for the host and positive for the parasite; (4) *commensalism* (+0) where the commensal species benefits and the other is not affected; and (5) *cooperation* (++) where both populations benefit from optional interaction, or *mutualism* (++) where both populations benefit from interaction essential to the survival of both partners. Odum (1997, pp. 200-201) writes:

> *Mutualism, the ultimate in cooperation, is extremely widespread and important. Many pairs or groups of species live together for mutual benefit as obligatory partners (neither can live alone). Indirectly, many mutualisms also benefit the ecosystem as a whole...*
>
> *Mutualism often involves two species that are different taxonomically, each of which has vital "goods or services" that the other needs. Microorganisms that digest cellulose and other resistant plat residues are often mutualistic with animals that do not have the necessary enzymes for this. Two example [are] ruminants and their rumen bacteria and termites and their intestinal flagellates... In both cases the microorganisms break down cellulose into fats and carbohydrates the animals can use, and the hosts provide the microorganisms with a place to live and protection against competitors or predators.*

Other examples of mutualism in the natural world include African acacia trees that provide homes for the ants that protect the tree from defoliating herbivorous insects, mycorrhizal fungi that enhance a plant's ability to extract nutrients from soil in exchange for some of the plant's photosynthate, and lichen – a composition of mutualistic algae and fungi that is so tightly co-evolved that botanists find it convenient to consider the association a single species.

This section concludes with an evocative description of the complex biological interactions that occur at coral reefs, excerpted from Odum (1997, p. 258):

> *Coral reefs can prosper in nutrient-poor waters because of water flow and a large investment in mutualism. Coral is a plant-animal superorganism, since algae, called zooanthellae, grow in the tissues of the animal polyp. The animal component gets its vegetables from the algae growing in its belly, and obtains its meat by extending its tentacles at night to fish for zooplankton in the water flowing past its limestone house – which the colony builds by deposing calcium carbonate from raw material that is plentiful in the ocean. The plant component of this partnership gets protection and nitrogen and other nutrients from the animal component.*
>
> *Because we humans must learn to prosper in a world of declining resources, the coral ecosystem serves as an example of how to efficiently retain, use, and recycle resources.... However, like a complex and energetic city, a finely tuned coral reef is neither resistant nor resilient to perturbations such as pollution or a rise in water temperature. In recent years the world's coral reefs have shown signs of stress that may be early warnings of global warming and oceanic pollution. An early sign of stress is the "bleaching" that occurs when the symbiotic algae leave the coral animal. If the mutualism is not restored, the coral slowly dies of starvation.*

Interactions within coral reefs are a natural segue to an ecosystem metaphor of business.

2.3 Metaphor

The first way in which "business ecosystem" is used in management discourse is as a *metaphor for shaping management thinking and action.* Management theory and practice has long drawn on biological images of organizations. Examples include the organic organizational structures of Burns & Stalker (1961), the natural systems perspective of Scott (1981), and the living company of de Geus (1997). The theoretical perspectives of evolutionary economics (Nelson & Winter, 1982) and organizational ecology (Hannan & Freeman, 1977; Hannan et al. 2007) are both anchored in strong biological and ecological metaphors, albeit at different levels of analysis.

A seminal 1993 *Harvard Business Review* article by James F. Moore introduced the term "business ecosystem" into popular management parlance. Moore (1993) argues for an *ecological* approach to management, where the modern business is viewed not as a member of single industry, but rather part of a *business ecosystem* that crosses a variety of industries. Moore sees clear parallels between businesses and biological systems, including the work of anthropologist Gregory Bateson on coevolution of species and biologist Steven Jay Gould on the collapse and re-establishment of natural ecosystems confronting radical environmental change. The article received the 1993 McKinsey Award for the best article published in *Harvard Business Review* that year, bringing attention from corporate executives and other management practitioners. Moore expanded these arguments in a follow-up book (Moore, 1996), describing the business ecosystem as "an economic community supported by a foundation of interacting organizations and individuals – the organisms of the business world" (p. 26). The ecosystem includes customers, suppliers, competitors, and other stakeholders, who "coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies" (p. 26)[16]. Iansiti & Levien (2004a, p. 9) further expand on Moore's ecological image:

> *We are not arguing here that industries* are *ecosystems or even that it makes sense to organize them* as if they were, *but that biological ecosystems can serve as a source of vivid and useful terminology as well as provide specific and powerful insights into the different roles played by firms.*

Moore (1993, 1996) and Iansiti & Levien (2004a, 2004b) collectively argue for the efficacy of an ecosystem metaphor of management – that managers who think of their companies as members of a business ecosystem can manage more effectively than managers who do not.

[16] In later writing, Moore (2006) treats the business ecosystem as a distinct organizational form rather than an ecological metaphor. For the purposes of this research, Moore (1993) and Moore (1996) are situated here in the category of metaphor, while Moore (2006) is situated in category of label. See also footnote 18.

A thick scholarly literature argues that metaphor is pervasive and fundamental to management theory and practice; a new and effective metaphor can therefore be a high-impact contribution. Morgan (2006, p. 4) writes: "All theories of organization and management are based on implicit images or metaphors that lead us to see, understand, and manage organizations in distinctive yet partial ways.... The use of metaphor implies *a way of thinking* and *a way of seeing* that pervade how we understand our world generally." Metaphor is ultimately rooted in a cognitive rather than merely linguistic process (Grant & Oswick, 1996, p. 1; Lakoff & Johnson, 1980); its surface expression is a linguistic phenomenon of images and words found in language, but its deeper expression is "a primal generative process that is fundamental to the creation of human understanding and meaning in all aspects of life" (Morgan, 1996, p. 228). The literature on metaphor collectively argues that managers' dominant metaphors shape how they perceive information and thus how they formulate problems (see also Newell & Simon, 1972; Simon, 1996; Reed, 2007), and how they form expectations of cause and effect (Senge, 2006). Metaphors exhibit taken-for-granted assumptions that "may remain not only unquestioned, but also beyond conscious awareness" (Morgan, 1980, p. 605).

"The essence of metaphor is understanding and experiencing one kind of thing in terms of another" (Lakoff & Johnson, 1980, p. 5). In the technical terms of Grant & Oswick (1996), the business ecosystem metaphor transfers information about the *base domain* of natural ecology and biological ecosystems to the *target domain* of modern interconnected business as a way for managers to make sense of their world. The following two examples illustrate the transfer of experience between base and target domains. The first example is from Moore (1993, p. 76):

> *Think of a prairie grassland that is succeeded by strands of conifers, which in turn evolve into a more complex forest dominated by hardwoods. Business ecosystems condense out of the original swirl of capital, customer interest, and talent generated by a new innovation, just as successful species spring from the natural resources of sunlight, water, and soil nutrients.*

A second example from Iansiti & Levien (2004b, p. 76) draws parallels between a keystone species of a biological ecosystem and a keystone organization of a business ecosystem:

> *An example of a biological keystone is the sea otter, which helps regulate the coastal ecosystem of the Pacific Northwest by consuming large numbers of sea urchins. Left unchecked, sea urchins overgraze a variety of invertebrates and plants, including kelp, which in turn support a food web that is the engine of near-shore productivity. The decline of the sea otter population in the nineteenth and twentieth centuries, when they were trapped for their fur, had a profoundly negative impact on a wide variety of coastal fish and other organizations. Like keystones in business networks, sea otters represent only a small part of the biomass of their community but exert tremendous influence. Note, too, that, as in business ecosystems, some individual members of the community – the sea urchins that get eaten by the otters – suffer as a result of the keystone's behavior, but the community as a whole benefits.*

The ecosystem metaphor brings some aspects of management into sharp focus. Moore (1993, 1996) emphasizes *interconnectedness* and *co-evolution*. Within a business ecosystem, companies work both cooperatively and competitively. As with the fungi and algae that together constitute lichen, companies may achieve better results cooperatively that would be impossible working alone. Within the larger business environment, several business ecosystems may compete with one another for survival and dominance of an area: "Superficially, competition among business ecosystems is a fight for market share. But below the surface, these new competitive struggles are fights over who will direct the future" (Moore, 1993, p. 85). Iansiti & Levien (2004a) emphasize the *networked* aspects of modern business: "More than any other type of network, a biological ecosystem provides a powerful analogy for understanding a business network" (p. 8). Specific features of biological ecosystems – structure, niches, relationships between members, kinds of connections, and roles of members – "suggest important analogies for understanding business networks" (p. 9). For Iansiti & Levien, the central features of the ecosystem metaphor are *shared fate*, and *collective behaviour*. Each member of a business ecosystem ultimately shares in the outcomes of the network as a whole, regardless of that member's apparent strength (2004b, p. 69). It is thus in the interest of all members to attend not only to the success of their own business, but also to the health of the larger business ecosystem.

The scholarly literature warns that metaphors can mislead if they are inappropriate or too specific: "All depends on whether the similarities the metaphor captures are significant or superficial" (Simon, 1996, p. 173). Metaphor always produces one-sided insights; in highlighting certain interpretations it tends to force others into a background role. It always creates distortions; evocative images create "constructive falsehoods" that, if taken too literally, become absurd. Any particular metaphor has "blind spots," which may be revealed by other metaphors: "Reality has a tendency to reveal itself in accordance with the perspectives through which it was engaged" (Morgan, 2006, p. 339), however, the "insights of one metaphor can often help us overcome the limitations of another" (Morgan, 2006, p. 342).

Various criticisms raised against ecological metaphors of business fall into one of two categories: that the metaphor goes too far, or that the metaphor does not go far enough. Criticisms in the first category argue that similarities between ecology and business are not large; thus an ecological metaphor may be at best unhelpful to managers, and potentially misleading. Criticisms in the second category are wary of imprecision, ambiguous language, and unclear or unstated logic; they argue for a stricter and more direct application of ecological theory to business. Both categories are discussed below.

The field of industrial ecology frequently debates the validity and appropriateness of ecological metaphors to business studies. Ecologist John Harte warns against treating ecosystems as a blueprint for business (*California Management Review*, 2001, p. 19): "The more you get to know how ecosystems work, the more you realize that they are not workable templates for human decision making." Natural evolved systems have properties that are undesirable in comparison to man-made systems that are deliberately designed and engineered to meet user requirements. Natural systems that are effective on the global scale may be very wasteful on a local scale; for example, eggs and young have very low survival rates in many species. *Ecosystem succession* – an important process in ecology – is driven by "catastrophes" of fires and floods, reminiscent of Schumpeter's (1942) creative destruction. "Natural and ecological and evolutionary processes in no way promote equity or justice. There is ... no regulatory body of forces that can protect the weak and unfortunate" (p. 19). To summarize the concerns from industrial ecology: "Applied to human society, what nature does is impractical for us to copy on a global scale, and wasteful, unstable, and sometimes even unethical for us to try to mimic on a local scale" (p. 27). Similarly, the Free Software Foundation (FSF) argues that it is a mistake to describe the free software community, or any human community, as an "ecosystem" because that word implies the absence of ethical judgment. Stallman (Free Software Foundation, 2010; see also Stallman, 1996b) writes:

> *The term "ecosystem" implicitly suggests an attitude of nonjudgmental observation: don't ask how what should happen, just study and explain what does happen. In an ecosystem, some organisms consume other organisms. We do not ask whether it is fair for an owl to eat a mouse or for a mouse to eat a plant, we only observe that they do so. Species' populations grow or shrink according to the conditions; this is neither right nor wrong, merely an ecological phenomenon.*
>
> *By contrast, beings that adopt an ethical stance towards their surroundings can decide to preserve things that, on their own, might vanish — such as civil society, democracy, human rights, peace, public health, clean air and water, endangered species, traditional arts...and computer users' freedom.*

To summarize the first category of criticism, an ecological metaphor can mislead if taken too seriously because business systems differ from biological systems in important ways: managers of companies have a moral responsibility to act ethically, they compete in ways unlike the predator-prey relationships between populations, and are bound by a regulatory framework of anti-trust and other laws that govern action.

The second category of criticism argues that managers have not taken the metaphor far enough. Okey (2004) charges that Iansiti & Levien (2004b) "intentionally misuse the ecosystem analogy with the justification that everyone else in the business community also misuses it," and that this misuse "has profound implications for the world's economies and ecologies, not just to individual businesses" (p. 132). Okey commends a broader systems view of business, but criticizes the article for leaving out half of the ecosystem – the natural capital of the physical and social environment[17].

[17] In the technical language of ecology, the analysis of Iansiti & Levien (2004a, 2004b) attends to the biotic components of the ecosystem with little attention to the abiotic components – the matter and energy of the environment. Thus, Okey argues (quite correctly) that a more appropriate ecological metaphor would be the "business community" (all economic organizations occupying an area) rather than the "business ecosystem" (the biotic community of economic organizations *and* the nonliving environment functioning together). Iansiti & Levien (2004b, p. 76) pleaded justification: "our use of the term 'ecosystem' is

Barnet (2006) criticizes Iansiti & Levien for recommending relaxed anti-trust policy, while considering only business firms within their ecology: "This ecology does not include the natural environment, communities, consumers, or employees.... They leave out all the other stakeholders that traditionally get crushed when firms are allowed to pursue self-interest under a stifled regulatory regime" (p. 90). In summary, the second category of criticism warns that an ecological metaphor can mislead if applied with insufficient rigour and precision, particularly if omitting important ecological components from the picture.

Likewise, advocates of the business ecosystem metaphor candidly acknowledge its limitations. Moore (1993, pp. 85-86) writes:

> *It's precisely in the role of conscious direction that a strictly biological metaphor is no longer useful. Business communities, unlike biological communities of co-evolving organisms, are social systems. And social systems are made up of real people who make decisions; the larger patterns are maintained by a complex network of choices, which depend, at least in part, on what participants are aware of.... [I]f you change the ideas in a social system, you change the system itself.*

Iansiti & Levien (2004a) identify three important differences between business networks and biological ecosystems that limit the generality of the business ecosystem metaphor: (1) business ecosystems are concerned with innovation while biological ecosystems are not; (2) although members of business ecosystems and biological ecosystems compete with one another, members of a business ecosystem are mobile and exercise freedom to move from one ecosystem to another; and (3) members of a business ecosystem are intelligent actors that make strategic choices.

probably closer to the biological term 'community.' We follow others in choosing ecosystem, rather than the generic-sounding community, because it clearly signals that we are discussing a complex system and that we are working with a biological analogy." Okey rejects this justification and calls on managers to use ecological terms in an ecologically correct way. See also the authors' rejoinder (Iansiti & Levien, 2004c).

Neither category of criticism is damning: both have merit, and consistent with the scholarly literature on metaphor, they highlight *limitations* – the boundaries beyond which the metaphor is unhelpful – rather than fatal weaknesses. In summary, there are several important ways in which business differs from biological ecosystems: evolved biological ecosystems lack an ethical or moral component, and can be inefficient, wasteful, and unstable, in comparison to man-made designed systems. Furthermore, an ecological metaphor can be misleading if applied in an incomplete way.

Morgan (2006) argues that effective managers adopt multiple metaphors and a plurality of perspectives. This is effective in management practice because different metaphors give rise to different theories of organization and management, and an understanding of that process can help managers master the strengths and limitations of different viewpoints, and use this knowledge to become more effective leaders and managers. Morgan (2006, p. xii) writes: "The ultimate challenge is not to be seduced by the power or attractiveness of a single metaphor – old or new – so much as to develop an ability to integrate the contributions of different points of view" (p. xii). According to Morgan (pp. 3-4), "Skilled leaders and managers... are aware that new insights often arise as one approaches situations from 'new angles' and that a wide and varied reading can create a wide and varied range of action possibilities. Less effective managers and problem solvers, however, seem to interpret everything from a fixed standpoint. As a result, they frequently hit blocks they cannot get around; their actions and behaviours are often rigid and inflexible." Morgan's arguments for the pervasiveness and influence of metaphor, and the effectiveness of a plurality of perspectives, support the assertions of

Moore (1993, 1996) and Iansiti & Levien (2004a, 2004b) that an ecological approach to management, as a complement to other perspectives and within appropriate bounds, can be a contribution to management practice.

In summary, the business ecosystem metaphor focuses manager attention at a system level of analysis above the individual company – a level that includes the company as well as the organizations, institutions, and individuals that impact it, with boundaries that cross traditional industry sectors. As with the populations of species comprising a biological ecosystem, the members of a business ecosystem co-evolve their capabilities and roles, and achieve collective outcomes that would be impossible alone. Properties of the ecosystem are emergent, in the sense that they are not discernible by examining the ecosystem components in isolation. In comparison to other metaphors of business and management, the ecosystem metaphor brings into sharp focus the notions of ecosystem health and diversity, shared fate and interconnectedness, collective behaviour, a wider range of possible relationship types between companies (especially commensalism, cooperation, and mutualism), a network structure of relationships rather than linear chain of suppliers and customers, feedback loops and time lags between cause and effect, competition between rival ecosystems, simultaneous value creation and value capture, and the existence of roles and specialized niches. The ecosystems metaphor implies *systems thinking* (Senge, 2006; Atwater et al. 2008) with cyclical feedback loops, varying time delays between actions and outcomes, and complex interactions, rather than the linear event-driven thinking more common in management practice. Just as any single metaphor has the potential to mislead and distort, the possible blind spots of the business ecosystems metaphor include the ethical or moral component of managerial decision-making, and the agency and mobility of companies to exit an ecosystem or join multiple ecosystems. As with all metaphors of management, the business ecosystem metaphor may be most useful in combination with other perspectives.

2.4 Label

The second way in which "business ecosystem" is used in management discourse is as a *label to classify or describe*. There are at least three distinct and qualitatively different ways in which the ecosystem label has been applied in the literature. First, the ecosystem label has been applied loosely by some authors and practitioners as a synonym for an established management concept, such as an organization's environment, supply chain, or interorganizational network. Second, the ecosystem label has been applied by some authors to identify a business environment with specific ecological properties, such as strong selection effects, rapid change, and constant threats, and distinguish those environments from other environments without those properties. Third, and most important for this research, the ecosystem label has been applied by Moore (2006) and others to distinguish a specific organizational form from its alternatives.

The first form of label substitutes the word "ecosystem" as a direct drop-in replacement of an established concept, sometimes employing both the new and established labels synonymously and concurrently. This has been observed for at least three established management concepts: (I) the business *environment,* (ii) the *supply chain,* and (iii) the *network* of social or contractual business relationships. One or more illustrative examples are provided here for each concept. At a recent board of directors meeting, a partner at an investment firm remarked to the author in a personal correspondence that “ecosystem is the new word for environment.” Consequently, that partner's company had revised their documents to use the word “ecosystem” everywhere that they had previously used the word “environment.” No difference in meaning or in thinking was implied: company documents recommended the same approaches and tools for environmental and industry sector analysis, but re-labeled them as ecosystem analysis. The question posed to an entrepreneur became “tell us about your ecosystem” rather than “tell us about your environment.” Tian et al. (2008) is an example of substituting the term “business ecosystem” for the term “supply chain.” The article develops a sophisticated framework for mathematically modeling the interactions of buyers and sellers in resource exchange relationships in a business-to-business service platform. The object of study is a supply chain, albeit a complex supply chain with regards to the number and properties of interactions. West & Wood (2008) is an example of substituting the term “business ecosystem” for the term “value network” (Norman & Ramirez, 1993; Amit & Zoff, 2001) – the inter-organizational network construct of the open innovation networks literature (Vanhaverbeke, 2006; Vanhaverbeke & Cloodt, 2006; Maula et al. 2006). West & Wood (2008, p. 3) write: “Here we use the two terms [ecosystem and value network] interchangeably, but favour 'ecosystem' in the explanation because it is the one used by the subjects in our study.” Iyer et al. (2006) is a second example, employing network analysis techniques to examine relationships between firms in the software sector (defined by sector industry code 7372), and designating that network of firm relationships as an “ecosystem” (p. 42).

The second form of label indicates an organizational environment with specific ecological properties. This appears to be the sense in which Teece (2007, 2009) adopts the term in his recent writing on dynamic capabilities. Teece (2009, p. 16) writes:

> *The 'environmental' context recognized for analytical purposes is not that of the industry, but that of the business 'ecosystem' – the community of organizations, institutions, and individuals that impact the enterprise and the enterprise's customers and supplies. The relevant community therefore includes complementors, suppliers, regulatory authorities, standard-setting bodies, the judiciary, and educational and research institutions. It is a framework that recognizes that innovation and its supporting infrastructure have major impacts on competition... [T]he ecosystem [is] the organizing paradigm for assessing developments in the environment.*

According to Teece, dynamic capabilities (and thus the business ecosystem construct) are most important in environments that display four characteristics: (1) the environment is open to international commerce and fully exposed to the threats associated with rapid technological change; (2) technical change is systematic: multiple inventions must be combined to create products and/or services that address customer needs; (3) there are well-developed markets for the exchange of component goods and services; and (4) the business environment is characterized by poorly developed markets in which to exchange technological and managerial know-how. “These characteristics can be found in large sectors of the global economy and especially in high-technology sectors” (p. 5). Under these circumstances, the foundations for success differ from operational effectiveness, neoclassical economics, and positioning-school business strategy. The “rules of the game” by which competitive force play out is “the result of coevolution and complex interaction between what might be thought of as (business) ecosystem participants” (2009, p. 11). In this framework, the business ecosystem is the environment in which the search activities for sensing and shaping new business opportunities occur (p. 13), where selection processes determine outcomes (p. 15), and to which successful business models must fit and adapt (p. 24). Many of the arguments are largely consistent with those of Moore (1993, 1996, 1998, 2006), with the important difference that Teece's ecosystem is anchored around a focal firm.

The third form of label is Moore's (2006) argument that business ecosystems are a distinct organizational form – a mode of organizing economic production that differs from markets and organizational hierarchies[18]. Moore (2006, p. 31) writes:

> *For more than sixty years, markets and hierarchies have dominated our thinking about economic organization... A third form, the ecosystem organizational form, has now become so important in practice that it should be accorded equal recognition in theory and policy-making. Markets, hierarchies and ecosystems are the three pillars of modern business thinking.*

Moore (2006, p. 33) defines business ecosystems as "intentional communities of economic actors whose individual business activities share in some large measure the fate of the whole community." The central concept is *co-evolution* – reciprocal interactions among technologies, business processes, products and services, market mechanisms, firm and ecosystem structures, and policy and regulation. "The activities of business ecosystems set the agenda for 'co-evolution' of markets and hierarchies and their outputs" (p. 32). "Just as the firm internalized markets under the visible hand of the entrepreneur, the ecosystem form internalizes systems of firms and the markets that connect them

[18] The argument that business ecosystems are a distinct organizational form (Moore, 2006) is logically different from the ecological metaphor of Moore (1993, 1996). See also footnote 16. Moore (1998) is a transition between these two perspectives, but the central argument of the paper is different: that the *E-form organization*, focusing on innovation and the creation of a business ecosystem of complementors, is challenging the M-form multidivisional vertically-integrated firm of Chandler (1962, 1982). E-form organizations exercise ecosystem leadership across the total range of ecosystem issues, and coordinate cross-ecosystem activities across multiple business ecosystems.

under the guiding hands of community leaders" (p. 73). Moore (pp. 73-74) clarifies the differences between markets, hierarchies, and ecosystems:

> *Markets facilitate transactions for goods. Hierarchies facilitate control over activities that produce goods. Ecosystems facilitate coordination of innovation in goods and the activities that produce them, as well facilitate managed co-evolution of the complex web of markets and hierarchies themselves. What differs in the three forms of organization are the ideal relationships that are sought and the levels of analysis used to judge performance. In markets, the ideal is perfectly transparent transactions for contributions, i.e. goods and services. In hierarchies, the ideal is perfect control of tasks. In business ecosystems, the ideal is perfect co-evolution of innovation across a multitude of contributors.*

Moore argues that there were business ecosystems even in the earliest days of the modern corporation, but their management received no attention. The organizational challenges of the day were elsewhere – primarily the efficiencies of men and machines. Enduring focus on yesterday's management problems remains deeply entrenched in neoclassical economics, operations management, business strategy, and assumptions underlying economic policy and anti-trust law, but the challenges today are different (p. 68):

> *Our challenge as a society today is to promote and combine creativity in order to solve major economic and social problems. The ecosystem organizational form grapples with this challenge. The business ecosystem organizational form is an important public good in a world of fast-moving technologies, globalization and the potential inclusion of vast numbers of contributors, and complex, multidimensional economic and social needs.*

Awareness of ecosystems, and their active management by stakeholders and policy-makers, is a recent development, with theory lagging practice: "Business ecosystems surround, permeate, and reshape markets and hierarchies. Managers establish business ecosystems to coordinate innovation across complementary contributions arising within multiple markets and hierarchies" (p. 32). According to Moore, markets, hierarchies, and ecosystems are three pillars of modern business thinking and should be accorded equal recognition in theory and policy-making.

Moore also proposes a second view of the ecosystem construct that offers complementary insights. In this second view, business ecosystems "can also be conceived as a network of interdependent niches that in turn are discovered and occupied by organizations. These niches can be said to be more or less open, to the degree to which they embrace alternative contributors. One of the most exciting ideas in business today is that business ecosystems can be 'opened up' to the entire word of potential contributors and creative participants" (p. 34). This can contribute to social welfare. "The fundamental public good offered by business ecosystems is the taking of a challenge that requires coordination and finding a way to take it out of a firm and bring in more participants. The business ecosystem organizational form solves the problem of how to open up and extend a framework of modularity and participation" (p. 48). "Volunteerism is part of every business ecosystem, but ... some use it much more than others. The most striking current example of the strategic use of volunteerism is in [open source software], based on the peer production of intellectual goods" (p. 45).

Moore's ecosystem exists separately from any particular organization. A company chooses voluntarily to enter or exit an ecosystem. Within the ecosystem, companies compete with one another on the basis of continuous innovation, creating a steep innovation trajectory that Moore describes as an "innovation treadmill" (p. 54):

> *Companies in a business ecosystem are under constant threat of competition and commoditization. The only way they can sustain themselves is by aggressively improving their contributions. The concept of a company as an aggressive continuous innovator is a far cry from the traditional concept of the firm, where scale and operating efficiency are essential, and improvement nice to have.*

Companies participating in ecosystems are spared three expensive problems: (1) simple stable interfaces permit exclusive focus on the problems within a particular domain, (2) reciprocal commitments permit focus on a narrower range of outcomes, and (3) costs of stewardship are shared.

This section has sought to clarify the different ways in which the "business ecosystem" label has been applied to management phenomena. A key message is that the label has been applied to a range of phenomena bearing a family resemblance (Wittgenstein, 1953) but differing in particulars; scholars and practitioners discussing "ecosystems" without further clarifying their subject are not necessary discussing the same thing. Christensen & Raynor (2003b) warn that concepts applied too broadly or without clear categories risk becoming empty buzzwords or management fads. Likewise, Bierstedt (1950, p. 730) warns that the more things a term can be applied to, the less precise is its meaning, and Dahl (1957, p. 210) warns that "a Thing to which people attach many labels with subtly or grossly different meanings... is probably not a Thing at all but many Things"[19].

2.5 Analogy

The third way in which "business ecosystem" is used in management discourse is as an *analogy to natural ecology*. This is related to metaphor, but with a stronger claim directed to a different audience: not only can the ecosystem metaphor inform the mental models of managers and thus usefully impact management practice, but also the extensive body of theory and empirical research on biological ecosystems can directly

[19] Bierstedt (1950) and Dahl (1957) were both writing about power, but their insights on the broader implications of ambiguous labels are directly applicable here.

inform management scholarship. In this view, theory of biological ecosystems is an analogous source for inspiration, constructs, explanations, and strong management theory with predictive power and prescriptions for practice. It posits an isomorphism between biological and social systems that is deeper and more fundamental than the similarity implied by metaphor[20]. Analogy with biological ecosystems can suggest areas to which management researchers could pay close attention. Three possible areas of useful analogy are network structure and roles, ecosystem properties (particularly health and diversity), and approaches to theory-building.

Biological networks often evolve one or more members that are highly connected to others and/or are disproportionately influential on the ecosystem as a whole (Iansiti & Levien, 2004a, 2004b), and populations of species occupy ecological niches classified by similarly of function or role. By analogy, this focuses attention on network structure of the business ecosystem; attention to network structure is not new, but the contribution of the ecosystem analogy is a set of categories with prescriptive implications, and useful adaptations of those categories to address the concerns of management researchers and practitioners. Iansiti & Levien (2004a, 2004b) adapt language from ecology[21] to propose

[20] Nonetheless, the difference between metaphor and analogy as described here is one of degree rather than kind, and classification of edge cases can be arbitrary. Likewise, the distinction between management practice and management theory is far from absolute. Management practice benefits from good theories that shape manager mental models (Senge, 2006; Christensen & Raynor, 2003a, 2003b), and good theory often has strong prescriptive implications (Christensen & Carlile, 2009; Bazerman, 2005). Arthur (2000) succinctly joined these two concepts by describing theory as "metaphors with entanglements." Astley & Zammuto (1992, p. 443) argue that the theory-practice link is actually one of language: "Organizational scientists should be viewed not as engineers offering technical advice to managers but as providers of conceptual and symbolic language for use in organizational discourse."

[21] The meaning of these terms, as defined by Iansiti & Levien (2004b), differs from the meaning in the ecology base domain. Odum (1997) used the term "ecological dominant" to refer to the few common species in community grouping, and the term "keystone species" to refer to species that exert controlling influence. A particular species could be a dominant or a keystone, both or neither. This is typical of management adaptations and is central to the criticisms of some ecologists, such as those Okey (2004).

that firms occupying influential *hub positions* (i.e., network nodes that are highly connected to other nodes) can adopt either a *keystone* role or a *dominator* role. *Keystones* exercise leadership to their own benefit, but also to the benefit of other ecosystem members. Keystones create platforms of services, tools, or technologies that other members of the ecosystem can use to enhance their own performance. *Dominators* instead adopt the short-term tactic of maximum value extraction, without attending to ecosystem health. Iansiti & Levien (2004a, p. 102) argue that recognition of the keystone role is a unique contribution of the ecosystem metaphor: "The idea of a keystone is virtually absent form existing theories of managing." The title of their book, *The Keystone Advantage* (2004a), is indicative of their central argument: that advantage accrues to keystone companies that exercise leadership and promote the health of the ecosystem. Other ecosystem members not occupying hub positions become *niche players* that pursue specialization strategies to provide differentiation from others. Iansiti & Levien treat each niche player company as unique; other management researchers have found it useful to group companies pursuing similar strategies into *niches* (e.g., Iyer & Davenport, 2008) – the business ecosystem analogue of ecological niches.

A second area of useful analogy is ecosystem properties. Ecologists argue that the properties of a ecosystem are emergent: just as the properties of water differ greatly from the properties of hydrogen and oxygen, the properties of an ecosystem are not discernible by examining the properties of ecosystem members, or the units of interest at any lower level of analysis (Simon, 1962, 1973, 1996). Ecologists attend to *ecosystem redundancy* (repetition), *resilience stability* (the ability to recover rapidly from a disturbance), and *resistance stability* (the ability to remain the same), and devote much effort to understanding antecedent factors, particularly species diversity. Iansiti & Levien (2004a, 2004b; Iansiti & Richards, 2006) propose an analogous multidimensional property of business ecosystem *health* with three dimensions: (1) productivity, (2) robustness, and (3) niche creation. *Productivity* is consistently transforming the raw materials of innovation into lower costs and new products; a simple but effective measure is return on invested capital. *Robustness* is the capability to survive disruptions; a simple if crude measure is the survival rates of ecosystem members – either over time or relative to comparable ecosystems. *Niche creation* is about exhibiting variety and diversity; a possible measure is the extent to which emerging technologies are being applied as new businesses and products. Ecologist have developed various ways of conceptualizing and measuring diversity through indices, tables, and graphic displays (Odum 1997). These may have useful analogs in business ecosystems; Page (2007) and Enayat et al. (2009) are early investigations of diversity.

A third area of potentially useful analogy is the approach and the underlying assumptions of building useful theory to explain ecosystems. Ecologists attend closely to material and energy flows, with cyclical feedback loops between variables, time delays, and complex interactions. The analogue in management practice, as noted previously in the discussion on metaphor, is what Senge (2006) calls "systems thinking" (Forrester, 1971; Ackoff, 1981). The analogue in management theory formalizes systems thinking into specific mathematical techniques of quantitative analysis and formal modelling through systems of differential equations (Sterman, 2001) or numerical simulation (Davis et al. 2007). *Complex adaptive systems* (Miller & Page, 2007; Holland, 1992) are a second potentially useful approach as a way of thinking about emergence and the evolution of designs and ecosystems over time. Axelrod & Cohen (1999) highlight the processes of *variation*, *interaction*, and *selection* in complex social systems, and suggest interventions that *harness complexity* rather than try to control it: "a perspective that is not explanatory but active – seeking to improve but without being able to fully control" (p. xvi). Peltoniemi (2006) proposes a framework for formal modelling of business ecosystems as complex adaptive systems with a strong biological analogy.

Other possible areas of useful analogy include relationships between populations, schemes for usefully classifying ecosystems and habitat, ecosystem succession and evolution, deeper insights into ecological niches, and other levels of analysis. The three examples explored here – network structure and roles, ecosystem properties, and approaches to building theory – are intended to be illustrative rather than comprehensive.

2.6 Other ecosystems in management and engineering

"Ecosystem" also appears as a metaphor, label, or analogy in various peripheral domains and contexts other than "business ecosystems". Ecosystem variants that share a family resemblance but differ in their particular definition and formulation include *software ecosystem* (Messerschmitt & Szyperski, 2003), *technology ecosystem* (Adomavicius et al. 2008), *innovation ecosystem* (Adner, 2006; Kao, 2009; Adner & Kapoor, 2010), *marketing ecosystem* (Prendergast & Berthon, 2000; Singer, 2006), *digital business ecosystem* (Corollo et al. 2007; Nachira et al. 2007), and *mashup ecosystem* (Weiss & Gangadharan, 2010). Table 7 is a compilation of definitions.

Table 3: Ecosystem definitions from the literature

Construct	Definition
Business ecosystem	Moore (1993): "I suggest that a company be viewed not as a member of a single industry but as part of a *business ecosystem* that crosses a variety of industries. In a business ecosystem, companies co-evolve capabilities around a new innovation: they work cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations" (p. 76). Moore (1996): A business ecosystem is "an economic community supported by a foundation of interacting organizations and individuals – the organisms of the business world" (p. 26). The ecosystem includes customers, suppliers, lead producers, competitors, and other stakeholders, who "coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies" (p. 26). Moore (2006): Business ecosystems are a distinct organizational form: "intentional communities of economic actors whose individual business activities share in some large measure the fate of the whole community" (p. 33). "Managers establish business ecosystems to coordinate innovation across multiple markets and hierarchies. The activities of business ecosystems set the agenda for 'co-evolution' of markets and hierarchies and their outputs" (p. 32). Moore (2006): Alternatively, a business ecosystem can be viewed as "a network of interdependent niches that in turn are occupied by organizations. These niches can be said to be more or less open, to the degree to which they embrace alternative contributions One of the most exciting ideas in business today is that business ecosystems can be 'opened up' to the entire world of potential contributions and creative participants" (p. 34). Iansiti & Levien: "A biological ecosystem provides a powerful analogy for understanding a business network" (2004a, p. 8). Business ecosystems are "loose networks – of suppliers, distributors, outsourcing firms, makers of related products or services, technology providers and a host of other organizations – [that] affect, and are affected by, the creation and delivery of a company's own offerings" (2004b, p. 69). Teece (2009): "The community of organizations, institutions, and individuals that impact the enterprise and the enterprise's customers and supplies. The relevant community therefore includes complementors, suppliers, regulatory authorities, standard-setting bodies, the judiciary, and educational and research institutions" (p. 16).
Software ecosystem	Messerschmitt & Szyperski (2003): A software ecosystem is a set of businesses functioning as a unit and interacting with a shared market for software and services, together with relationships among them. These relationships are frequently underpinned by a common technological platform and operate through the exchange of information, resources, and artifacts.
Digital ecosystem	Corallo et al. (2007): The digital ecosystem is the enabling technology for a business ecosystem – "the digital software environment that supports the development of distributed and adaptive technologies and evolutionary business models for organizations" (p. 4). "The ICT-enabling infrastructure that supports the cooperation, the knowledge sharing and the building of a digital business ecosystem. The digital ecosystem is the pervasive soft support infrastructure populated by digital species able to evolve, adapt and mediate services and knowledge. In this metaphor the digital ecosystem is populated by digital species, exhibiting the structure and behaviour of natural species as much as a business ecosystem is populated by business species" (pp. 2-3).
Digital business ecosystem (DBE)	Corallo et al. (2007): A DBE is a business ecosystem overlaid with a digital ecosystem for Internet-based coordination between organizations; "Each organization adds one or more distinct aspects of product/service value to the value generated by the ecosystem by exchanging digital knowledge with other members: the business ecosystem evolves into the digital business ecosystem (DBE)" (p. 2).
Innovation ecosystem	Adner (2010): An innovation ecosystem is an extension of the supply-chain that considers not only suppliers and customers of a focal firm, but also complementary offers combined by the customer.
Technology ecosystem	Adomavicius et al. (2008): A technology ecosystem is a forecasting tool that considers the components, products and applications, and infrastructure technologies surround a focal technology.
Mashup ecosystem	Weiss & Gangadharan (2010): The mashup ecosystem the network of mashups (software that combines data and services through open interfaces), mashup platforms, data providers, and users.

Some constructs are variants of the business ecosystem framework (e.g., Messerschmitt & Szyperski, 2003; Adner, 2006; Kao, 2009). Others address systems comprised of technology elements rather than of business organizations or people (e.g., Corallo et al. 2007, Adomavicius et al. 2008, Weiss & Gangadharan, 2010). Insights from one ecosystem perspective can potentially inform another if the differences are recognized and carefully considered, and if scholars are alert to the potential for unclear labels, multiple labels that refer to the same phenomenon, and labels that admit multiple phenomena. There is a strong theoretical rationale to expect technological and organizational perspectives to be deeply entwined (Baldwin & Clark, 2000; Messerschmitt & Szyperski, 2003; see also section 3.4). Nonetheless, different constructs should be employed together only after considering ontological differences.

2.7 Summary and synthesis

Within the researcher and practitioner literature and the management vernacular, a variety of meanings are attached to the term "business ecosystem." As well, scholars, practitioners and business press authors have proposed an assortment of other similarly-named ecosystem concepts. Authors typically cite articles by Moore (1993, 1996, 1998, 2006), Iansiti & Levien (2004a, 2004b), and sources from the practitioner literature, without always addressing the ontological and semantic differences between their own constructs and the foundation articles being cited, or acknowledging that the constructs of these foundation articles differ from one another in technical but important ways.

The earliest writing on business ecosystems (Moore, 1993, 1996; Iansiti & Levien, 2004a, 2004b) argued that modern business has similarities to ecology, thus an ecological *metaphor* of management may be able to help managers make better decisions. The business ecosystem metaphor focused manager attention on interconnectedness and co-evolution, the networked aspects of modern business, shared fate and collective behavior, and a wider range of possible relationship types between companies. As these arguments became established in management discourse, practitioners and researchers increasingly adopted "ecosystem" as a *label* to describe specific management phenomena. In particular, Moore (2006) argued that the business ecosystem is a distinct mode of organizing economic production, equal in economic importance to hierarchical organizations and competitive markets, and demanding of equal recognition in theory and policy making. As well, some authors attached the "ecosystem" label to existing concepts, and others adopted the ecosystem label to distinguish the increasingly dynamic context of modern business from the static "business environment" of the past. In parallel, a body of management scholarship – sometimes overlapping with the early writing on ecological metaphors – began applying ecological theory, methods, tools and concepts more formally and with varying degrees of rigour to management theory and practice. The ecosystem *analogy* has enriched managerial discourse with a vocabulary around network structure and roles (hubs, niches, keystones, and dominators) and ecosystem properties (health, productivity, robustness, and niche creation). It is a young literature with much promise for continuing and future contribution.

Collectively, the literature surveyed in this chapter suggests that the various formulations of the business ecosystem construct can be situated within a conceptual space spanned by four dimensions. The first dimension is the *conceptual location* of the construct: is the business ecosystem construct a level of analysis *above* the firm or is it an economic mode of production that is an *alternative* to the firm? The second dimension is the *anchor* point: is the business ecosystem construct anchored around a *shared platform*, a *focal firm*, or is there *no anchor* (for example, a system defined only by its boundaries)? The third dimension is *boundaries*: is the ecosystem bounded by *proximity* of members to the ecosystem anchor, a *firm-level property* such as membership or industry sector code, or something else? The fourth dimension is the nature of the *elements* that comprise the ecosystem. Analogous to the biotic species of biological ecosystems, the elemental building blocks could be *organizations and individuals*, *technological elements*, or something else entirely; other phenomena may be analogous to the abiotic habitat. Table 4 is a summary of these four dimensions with examples from the literature. Specifying each dimension of an ecosystem construct appears sufficient to uniquely situate all of the different "ecosystems" reported in this chapter within the broader conceptual space.

Table 4: Dimensions distinguishing ecosystem constructs in the literature

Dimension	Description	Examples
Conceptual location	What is the underlying conceptual nature of the ecosystem construct?	• a level of analysis *above* the firm (Moore, 1993) • an economic mode of production that is an *alternative* to the firm (Moore, 2006)
Anchor	What is the anchor point of the ecosystem construct?	• a *shared platform* (Iyer & Davenport, 2008) • a *focal* organization (Li, 2009) • *no anchor* point, only boundaries (Iyer et al. 2006)
Boundaries	What is the boundary of the ecosystem construct that determines membership?	• relational *proximity* to the ecosystem anchor (Adner, 2010) • a property of the ecosystem member o industry code (Iyer et al. 2006) o membership status (West & Wood, 2008) • *geographic* boundaries – i.e., a regional cluster (Porter, 1990)
Elements	What are the elemental building blocks of the ecosystem construct?	• organizations and individuals (Moore, 1993, 1996, 2006; Iansiti & Levien, 2004a, 2004b; Teece, 2007, 2009) • technologies (Corallo et al. 2007; Adomavicius et al. 2008) • software services (Weiss & Gangadharan, 2010) • etc.

The key insight of this conceptual development and analysis is that applications requiring precision, such as rigorous scholarly management research, need to be explicit and clear about *which* ecosystem construct is being discussed. In later chapters of this research, the business ecosystem construct of Moore (2006) features prominently. In the framework of Table 4, the Moore (2006) business ecosystem is an economic mode of production, comprised of individuals and organizations, anchored around a shared platform, with membership determined by the relationship to the platform. It is primarily a label, but also draws strongly on Moore's earlier arguments on the efficacy of ecological metaphors (Moore, 1993, 1996). Authors may find Table 4 useful to explicitly situate their own ecosystem constructs, especially when citing arguments from previous work and when comparing results.

Within the present research, the meaning of "Eclipse ecosystem" is treated as an empirical question to be answered by attending closely to participant behaviour and discourse. The literature surveyed in this chapter, and the framework of Table 4, are conceptual building blocks useful in later chapters to interpret case evidence and theorize about relationships between constructs. The first guiding question of this research – "What is the Eclipse ecosystem?" – is answered, in part, in chapter 7 and chapter 11 by situating Eclipse within the conceptual space spanned by the four dimensions of Table 4.

This chapter has examined the business ecosystem as a management construct. The next chapter examines what the broader management literature has to say about business ecosystems.

3 Business ecosystems as a nexus of management literatures

This chapter examines the salient contributions from six empirically-grounded management literatures: (1) innovation, (2) open innovation, (3) platforms, (4) modularity and design, (5) interorganizational networks, and (6) communities. Each literature usefully informs some aspect of empirical inquiry into business ecosystems.

Table 5 summarizes constructs and insights drawn from each literature stream that are salient to business ecosystems. The following six sections then briefly and selectively outline the theoretical arguments and empirical evidence employed in later chapters to guide data collection, analysis, and theory development for this research.

3.1 Innovation

Innovation touches many domains – including management, engineering, economics, psychology, education, and the humanities – and many variant definitions exist. One criterion on which researchers generally agree is that an innovation has economic or social impact beyond an invention in the laboratory or an idea in an inventor's mind. Schumpeter (1934, p. 88) wrote: "As long as they are not not carried into practice, inventions are economically irrelevant. And to carry any improvement into effect is a task entirely different from the inventing of it, and a task moreover, requiring entirely different kinds of aptitudes." Innovation was once associated strongly (even exclusively) with firms selling products to consumers or to other firms – "An *innovation* in the economic sense is accomplished only with the first *commercial* transaction" (Freeman, 1982, p. 7) – but more recently, scholars also acknowledge the economic and social impacts of innovation through means other than direct commercial transactions.

Table 5: Contributions from a nexus of extant management research

Research stream	Key references	Central constructs	Salient insights for business ecosystems
Innovation	Freeman (1982); Allen (1983); Teece (1986, 2009); Roberts (1988); Utterback (1994); von Hippel (1988, 2005); Pisano (2006); Scotchmer (2006).	• Two aspects: invention, and commercialization • Complementary assets: generic, specialized, and cospecialized • Categories of innovation	• Innovation is increasingly a basis for competitive advantage for firms. • Innovation involves both creative invention and the commercial realization of value. • Innovators do not necessarily profit from the value created by their innovations; many factors interact to determine how value is appropriated. • Innovation depends on intangible and tacit assets not easily transferred. • Innovation arises from many sources: producers, users, organizations, communities.
Open innovation	Chesbrough (2003ab, 2006); Chesbrough et al. (2006); Chesbrough & Appleyard (2007); West & Bogers (2010).	• Inflows and outflows • Business models • Roles for innovators • Value creation and capture	• Firms can often do better by opening their R&D process to inflows of technology and ideas and to outflows of revenue opportunities rather than by keeping R&D closed. • Many possible roles for innovators are narrowly specialized and potentially profitable. • Business models transform potential into economic value; no value in technology per se.
Platforms	Cusumano & Gawer (2002); Gawer & Cusumano (2002, 2008); West (2003); Gawer (2009a).	• Leaders, complementors • Network effects; tipping • Degree of openness • Types and properties • Multiple sides and niches	• A platform is a foundation of building blocks for products, technologies and services. • Network effects reinforce early advantage; difficult to dislodge, past a tipping point. • Benefits accrue to the platform leader who can appropriate a larger share of value. • Complementors develop offers and technologies that increase the value of the platform. • Emphasis on platform leadership; less is known about community-developed platforms.
Modularity and design	Simon (1962, 1996); Baldwin & Clark (2000, 2006); Christensen et al. (2001); Baldwin & Woodard (2009).	• Architecture; interfaces • Visible design rules and hidden module parameters • Co-evolution of systems	• Platform architecture constrains and enables organization and industry structure. • Modularity enables design evolution; users select on value; architecture, organization, and industry structure are isomorphic in modularity and co-evolve together. • Modular architecture is at the heart of the open source software development process.
Inter-organizational networks	Williamson (1971, 1985, 1991); Chandler (1962, 1977, 1990); Powell (1990); Castells (2000); Iyer et al. (2006); Weiss & Gangadharan (2010).	• Markets, hierarchies, hybrids, networks • Inter-firm relationships • Relationship types • Relationship properties	• Many rival theoretical explanations partly account for the existence of markets, organizational hierarchies, interfirm relationships, and inter-organizational networks; none adequately account for business ecosystems. • Various constructs and measures exist to map interorganizational networks, but the data required to create network maps and compute measures is often difficult to obtain.
Communities	Raymond (1999); Perens (2007); Bacon (2009); O'Mahony (2003, 2005, 2007); Dahlander & Magnusson (2008); West & O'Mahony (2008); Boudreau & Lakhani (2009); Baldwin & von Hippel (2009).	• Community innovation • Open source software • Sponsored or autonomous community types • Design dimensions of sponsored communities • Transparency; accessibility	• Communities are voluntary groups of individuals with a common interest, sense of belonging, and shared identify; communities are sources of innovation and support. • Companies can benefit by sponsoring and deliberately cultivating communities. • When designing a community, sponsors consider the transparency and accessibility of three design dimensions: production, governance, and intellectual property. • Sponsored communities can transition to autonomous communities, but this process is not well understood; governance and company involvement is not well understood.

Nomenclature has been proposed to clarify the components of innovation. Roberts (1988; 2007, p. 36) offers a useful set of definitions:

> *Innovation is composed of two parts: (1) the generation of an idea or invention, and (2) the conversion of that invention into a business or other useful application... Innovation = Invention + Exploitation... Whereas invention is marked by discovery or a state of new existence, usually at the lab or bench, innovation is marked by first use, in manufacturing or in a market.*

West & Bogers (2010) label the two parts of the innovation process as *creation* (conception of a new invention or technology) and *commercialization*: "In the latter category, we also include non-commercial diffusion of inventions as they often have an economic impact" (p. 15)[22].

Innovators do not necessarily profit from their innovations (Teece, 1986). Depending on circumstances, profits from innovation may accrue instead to suppliers, intermediaries, competitors who copy or improve on the innovation, or customers. Factors that impact how the value of an innovation is appropriated include control and access to *complementary assets*, the *excludability regime* of whether or not others can be prevented from copying the innovation, *strategic choices* of the innovator and of others, and whether or not design decisions have converged around a *dominant design* (Abernathy & Utterback, 1978; Mitchell, 1991; Utterback, 1994). Particularly in regimes of strong excludability, new entrant innovators may be able to profit by collaboration

[22] Both exploitation and commercialization are partially misnomers, because "exploitation" may imply negative connotations, and "commercialization" is used to encompass situations involving no "commerce" in the sense of market transactions for business or financial gain. A better label would imply not only market transactions but also social forces, voluntary free sharing, agents with multiple motivations, open science, free software, adoption, diffusion, etc. Possible alternatives include dissemination, realization, actualization, communication, distribution, diffusion, productization, put up for sale, or convert to a useful thing, but "commercialization" remains the most widely used term for realizing value from a creative act of invention. A succinct inclusive label for the second component of innovation has not yet emerged.

with incumbents through intermediary "markets for ideas" (Gans & Stern, 2003; Gans et al. 2002, 2008). Alternatively, firms may deliberately choose to forgo excludability innovate to accelerate widespread adoption (Teece, 2006) and profit from an innovation's complements rather than the innovation itself (Pisano, 2006; Pisano & Teece, 2007).

Innovation depends on knowledge and know-how that is often tacit and intangible, difficult to codify, and not easily transferred or traded (Teece, 1981, 1982, 2000). "Intangibles typically do not reside with just a few individuals, but are deeply embedded in processes and procedures and sometimes even in simple (strategic) rules" (Teece, 2009, pp. iix-ix). Furthermore, as Teece (1986, p. 288) writes:

> *In almost all cases, the successful commercialization of an innovation requires that the know-how in question be utilized in conjunction with other capabilities or assets. Services such as marketing, competitive manufacturing, and after-sales support are almost always needed. These services are often obtained from complementary assets which are specialized. For example, the commercialization of a new drug is likely to require the dissemination of information over a specialized information channel.*

Complementary assets can be *generic* (not specialized to the innovation), *specialized* (unilateral dependence) or *cospecialized* (bilateral dependence). Teece (2009, p. 41) writes: "Cospecialized assets are a particular class of complementary assets where the value of an asset is a function of its use in conjunction with other assets. With cospecialization, joint use is value enhancing. Cospecialization results in 'thin' markets; that is, the assets in question are idiosyncratic and cannot be readily bought and sold."

The traditional emphasis of the management literature has been *producer innovation* where the firm is the locus of innovative activity. A growing literature recognizes that innovation also occurs outside the focal business firm, by end-users, suppliers, customers, universities, government and private laboratories, and other nations in increasingly global markets (von Hippel, 1988). *User innovation* (von Hippel, 1986, 2005; Baldwin et al. 2006) is concentrated among *lead users* – those users who experience needs ahead of the majority of a target market and who expect attractive innovation-related profits from a solution to their needs, and so are likely to innovate. *Community innovation* (von Hippel, 2001; Franke & Shah, 2003), exemplified by community-developed open source software (section 3.6), is an increasingly important source of user innovation. *Cumulative innovation* (Scotchmer, 2006; Fiona & O'Mahony, 2007) and *collective invention* (Allen, 1983) is distributed across multiple firms. Allen (1983) argues that many new and important production techniques have been developed collectively by groups of firms rather by than a single innovator in isolation who conceals their work. When firms collectively invent, they make available to their competitors the results of new plant designs so that their competitors can incorporate extensions of those designs into new facilities they build. Likewise, Scotchmer (2006) argues that knowledge "spillovers" between rivals, traditionally an undesired cost of R&D to be deliberately minimized, play a crucial role in advancing technological progress and improving social welfare. In this view, much innovation is cumulative, with new innovations building successively on previous innovations.

Some scholars have proposed different categories of innovation. Tushman & Anderson (1986) distinguish between *competency-enhancing innovation* and *competency-destroying innovation*, where the competencies of interest are those of the incumbent firms. Henderson & Clark (1990) distinguish between *modular innovation* that overturns core concepts but leaves linkages between components unchanged, and *architectural innovation* that redefines linkages without component changes. Henderson & Clark also redefine *radical innovation* as simultaneous change to both linkages and components, and *incremental innovation* as change that reinforces core concepts and linkages. Moore (1991) distinguishes between *continuous innovation* that requires no changes in the behaviour of technology adopters, and *discontinuous innovation* that requires behavioural change to adopt the technology. Christensen (1997) distinguishes between *sustaining innovation* where incumbents are at an advantage over new entrants, and *disruptive innovation* that is often associated with changes in market leadership. Christensen & Raynor (2003a) further distinguish between *low-end disruptions* that deliver a lower cost structure into existing value networks and *new-market disruptions* that deliver new value propositions into value networks that did not previously exist. Christensen (2006) argues that progress in the field of innovation has been, in part, the discovery and articulation of sharper and more fundamentally robust categories. Categories of innovation may be salient here to the extent that business ecosystems, vendor neutral governance, and community-developed platforms may be better matched to some categories than to others; alternatively, these organizing forms may better nurture some categories of innovation over others.

In summary, the salient lessons from the innovation literature include the following. (1) The capability to innovate is central to achieving and sustaining competitive advantage over rivals. (2) Innovation involves *both* creative invention *and* the realization of economic or social value. (3) Innovators are not necessarily the ones to profit from their innovations; who profits depends on many factors. (4) Innovation depends on tacit knowledge and know-how that is not easily transferred. (5) Innovation has many sources: not only producers, but also users, other organizations, and communities. (6) In some circumstances, there may be strong incentives to reveal and share rather conceal or protect innovation.

3.2 Open innovation

Chesbrough et al. (2006, p. 1) describe open innovation as follows:

> *Open Innovation is the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively. Open Innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology. Open Innovation processes combine internal and external ideas into architectures and systems. They utilize business models to define the requirements for these architectures and systems. The business model utilizes both external and internal ideas to create value, while defining internal mechanisms to claim some portion of that value. Open Innovation assumes that internal ideas can also be taken to market through external channels, outside the current businesses of the firm, to generate additional value.*

In the open innovation literature, "open" and "closed" refer to the *boundaries of the firm*, specifically the *R&D process* within the firm[23]. The object of central interest is a *focal firm*, and the central concept is improving firm performance, especially profitability.

[23] This an important and sometimes overlooked difference between the open innovation literature and other literatures examined here. Chesbrough's (2003a) interest is the R&D process of a focal firm, depicted following Wheelright & Clark (1992) as a funnel where many ideas enter at the wide base on the left and new products and services emerge from the narrow neck on the right. Likewise, "open innovation" and "closed innovation" refer to the *boundaries of the firm's R&D process* – whether that process is porous to admit inflows and allow outflows through the boundary, or vertically integrated such that all value chain activities from basic research, to development, manufacturing, operations, and customer support are conducted by the focal firm so that inflows and outflows occur only at the base and neck. In other literatures examined here, "open" refers instead to the accessibility and transparency of information and artifacts, property rights, and governance. Potential confusion is exacerbated by the overlap in empirical domains: companies pursuing an open innovation strategy frequently seek to benefit from open source software and user communities, and may choose to reveal and distribute assets under open source licenses. For example, bibliographic analysis of the open innovation literature by Dahlander & Gann (2010) reported that five of the ten articles most cited by open innovation papers directly concerned open source software.

According to Chesbrough (2003a), a *closed innovation* mindset of vertical integration with economies of scale and scope (Chandler, 1990) dominated industry for much of the twentieth century. In closed innovation thinking, the firm is the locus of innovation, spillovers from industrial R&D are an undesired cost of doing business, useful knowledge is scarce, intellectual property is employed defensively, and all important activities are internal to the firm (Chesbrough et al. 2006, pp. 4, pp. 6-11). Projects can only enter the R&D process of the firm in one way, from the science and technology base of the firm, and can only exit one way, by going to market as products or services provided by the firm.

In *open innovation* thinking, there are many ways for projects to enter the R&D process of the firm and many ways for them to profitably exit. Projects can be launched from either internal or external technology sources, and new technology can enter into the process at various stages. Projects can go to market through a company's own marketing and sales channels, or through licensing or creation of spin-off companies, or the sale of intellectual property rights in *intermediate markets* (Chesbrough, 2006). Firms that adopt open innovation do so because it make them more money: "*Purposeful* inflows and outflows are an opportunity for revenue" (Chesbrough et al. 2006, p. 4). "Firms are often better off commercializing external sources of innovations, and finding external paths for commercializing internally sourced innovation" (West & Bogers, 2010, p. 5). Open innovation is both a set of management practices and a cognitive model for creating, interpreting and researching those practices (Chesbrough et al. 2006, p. 286).

In environments with open innovation, there are multiple roles for innovators, and a firm may choose to focus its activities quite narrowly. For example, Chesbrough (2003b) identifies eight *open innovation roles* for organizations that fund, create, and commercialize innovation. (1) Innovation *investors* provide capital and other services to new ventures in exchange for expectations of future returns[24]. (2) Innovation *benefactors* provide research or commercialization funds as a strategic objective. Investors and benefactors collectively *fund innovation.* (3) Innovation *explorers* perform the discovery function that previously took place in corporate R&D laboratories. (4) Innovation *merchants* focus on a narrow set of technologies that are codified into intellectual property to be sold or licensed. (5) Innovation *architects* partition complex systems in ways that others can contribute to. (6) Innovation *missionaries* create and advance technologies to serve a cause other than financial profits. Collectively, explorers, merchants, architects and missionaries *generate innovation.* (7) Innovation *marketers* develop a deep understanding of current and potential needs in the market to identify which outside ideas to bring in-house. (8) *One-stop centres,* such as IBM's Global Services division, provide comprehensive products and services from all sources. Innovation marketers and one-stop centres collectively *commercialize innovation.*

A *business model* links technological potential to economic outcomes (Chesbrough & Rosenbloom, 2002). Chesbrough (2003a, p. xxx) writes:

> *There is no inherent value in a technology per se. The value is determined instead by the business model used to bring it to market. The same technology taken to market through two different business models will yield different amounts of value. An inferior technology with a better business model will often trump a better technology commercialized through an inferior business model.*

[24] Large & Muegge (2008) survey the literature on non-financial value-added (NFVA) by VC investors.

Teece (2009, p. 19) echoes this importance: "There is considerable evidence that business success depends as much on organizational innovation, for example, design of business models, as it does on the selection of physical technologies." Chesbrough et al. (2006, p. 31) identify two critical business model functions: "It must *create value* within the value chain; and it must *capture* a piece of *value* for the focal firm in that chain."

Researchers have proposed various frameworks for operationalizing the business model construct. The Johnson et al. (2008) framework consists of four interlocking elements that, taken together, create value for the company and deliver value to the customer: (1) a *customer value proposition* (CVP) that explains the target customer, the job to be done to solve an problem for the target customer, and the offer that satisfies the problem, (2) a *profit formula* of how the company makes money, comprised of a *revenue model*, a *cost structure*, a *margin model*, and *resource velocity*, (3) *key resources* needed to deliver the CVP, and (4) *key processes* that make the profitable delivery of the CVP repeatable and scalable. "As simple as the framework may seem, its power lies in the complex interdependencies of its parts" (p. 53). Changes to any element affect the others and the whole[25]. Regardless of the specific formulation, Chesbrough & Rosenbloom (2002) argue that the set of all feasible business models is not foreseeable in advance. A business model is discovered through search and heuristic logic in a reshaping process creating learning opportunities that themselves may contribute importantly to success.

[25] The Johnson et al. (2008) framework in its original formulation was developed to help established companies with a multidivisional structure delivering physical products to global markets to develop and implement new successful business models. Chapter 6 and chapter 10 of this study employ an adapted subset of the Johnson et al. (2008) framework to partially specify and compare the business models of case companies participating in the Eclipse ecosystem. The adapted model articulates a group of *target customers*, a *value proposition* (for those target customers), a trigger for *revenue generation* (i.e., a subset of the *profit formula*), and *required capabilities* (either resources or processes).

Chesbrough & Appleyard (2007) argue that established theories of business strategy are inadequate guides for managers confronting open innovation and related phenomena[26]. "If we are to make strategic sense of innovation communities, ecosystems, networks, and their implications for competitive advantage, we need a new approach to strategy – what we call 'open strategy'" (p. 58). They write (p. 73):

> *Open strategy balances the powerful value creation forces that can be found in creative individuals, innovation communities, and collaborative initiatives with the need to capture value in order to sustain continued participation and support of those initiatives. Traditional concepts of business strategy either underestimate the value of open invention and open coordination, or they ignore them outright. As the concept of openness spreads from software to science and other industries, we will need to update our concepts of strategy. Open strategy is an attempt to supply this update.*

Items of central importance in traditional business strategy, such as ownership, control, entry barriers, switching costs, and intra-industry rivalry, are of secondary importance in open strategy. Items beyond the scope of traditional strategy, such as the role of community participation, attracting individual volunteers, construction of innovation networks, and the notion of innovation ecosystems, are of central importance.

West & Bogers (2010) attempt a synthetic view of open innovation (OI), user innovation (UI), and cumulative innovation (CI) under a unified "metacategory" of *distributed innovation*[27]. All three literatures address anomalies in the *vertical integration innovation* paradigm of a firm commercializing its own inventions that

[26] Interestingly, Chesbrough & Appleyard (2007) employ a conventional definition of openness: "pooling of knowledge for innovative purposes where the contributors have access to the inputs of others and cannot exert exclusive rights over the resultant innovation" (p. 60); cf. footnote 81 and Chesbrough (2003a).

[27] West & Bogers (2010) is situated here in the open innovation stream rather that the innovation stream for two reasons. First, situating it here connects the open innovation literature with literatures that were previously reviewed, so that each subsection of this document builds logically on what came before. Second, although West & Bogers (2010) is a unifying and synthetic work, the open innovation perspective is most prominent, such that the user innovation and cumulative innovation literatures are largely situated within the open innovation cognitive model that emphasizes a focal firm.

dominated 20th century business thought (e.g., Chandler, 1962, 1977, 1990), but they differ in their assumptions and core research questions. Nine distinct *innovation modes* differ in three areas: (1) the locus of innovation *creation* (the supply-side of innovation; suppliers, science entities, users, rivals), (2) paths to *commercialization or diffusion* (the demand-side of innovation; a five-category typology based on locus of creation and locus of commercialization), and (3) *institutional conditions* (enabling assumptions and domain boundaries; industry maturity, access to knowledge, markets for innovation, appropriation and IP, community size). The two open innovation modes are *OI-inbound* (the firm commercializes others' innovations) and *OI-outbound* (others commercialize the firm's innovations). The four user innovation modes are *UI-input* (producers commercialize the innovation of lead users), *UI-self* (users enhance their own utility without sharing), *UI-share* (non-commercial diffusion of user innovation), and *UI-startup* (user-entrepreneurs self-commercialize their innovations). The two cumulative innovation modes are *CI-share* (innovators cooperatively share knowledge) and *CI-rival* (knowledge leaks between rivalrous competitors). West & Bogers note that published research has tended to emphasize the outside creation/inside commercialization quadrant, including the modes of inbound open innovation, UI-input, and UI-startup. A much smaller body of research considers the inside creation/outside commercialization quadrant in which a focal firm internally develops an innovation that is commercialized externally: "With the notable exception of OI-outbound research inspired by Chesbrough (2003[a]), little research has considered … creating innovations within the firm but commercializing them outside the firm" (p. 22).

In summary, the open innovation literature examines the rationale for a profit-seeking focal firm to "open up" its R&D process to purposeful inflows and outflows of knowledge – necessary conditions for effective participation in business ecosystems. The salient lessons from the open innovation literature include the following. (1) Profit-seeking firms can often perform better by "opening up" the vertically-integrated R&D process to purposeful inflows of ideas and technology and to alternative outflows of revenue opportunities. (2) In environments of open innovation, there are many potentially profitable roles for innovative organizations; each role specialize on a narrow aspect of the R&D process. (3) Business models convert technological potential into economic value; an effective business model both creates value and captures a portion of that value for the focal firm. (4) Items of central importance in open strategy are outside the scope of traditional business strategy; items of importance in traditional business strategy are of secondary importance in open strategy. (5) Distributed innovation, a metacategory including user innovation, cumulative innovation, and open innovation, is a potentially useful framework for categorizing innovation in business ecosystems.

3.3 Platforms

Gawer (2009a) defines an industry platform as "a building block, providing an essential function to a technological system – which acts as a foundation upon which other firms can develop complementary products, technologies, or services" (p. 2). Platforms are typically subject to positive feedback loops through *network effects in use* (Katz & Shapiro, 1985) and *increasing returns in supply* (Arthur, 1994, 1996) that tend to amplify early advantage: the more people who use platform products, the more incentives there are for complementors to introduce more complementary products (Cusumano & Gawer, 2002). Gawer (2009a, pp. 2-3) writes: "Platforms that make it past a certain tipping point tend to become really hard to dislodge. In a sense, as platforms' market share grows, so also grow their own barrier to entry." Gawer (2009b) distinguishes between four platform types: (1) *internal* platforms within a single firm, (2) *supply chain platforms* that link several firms within a supply chain, (3) *industry platforms* that link "industry ecosystems" of firms, and (4) *multi-sided platforms* that are an intermediary for transactions between several firms or groups of firms. Over time, platforms that are successful tend to promote to higher types – from internal, to supply chain, to industry[28].

Baldwin & Woodard (2009) note that platforms are defined differently in a least three different literatures: product development, technology strategy, and industrial economics. In the product development literature, the emphasis was on effective product development within a firm – internal platforms in the Gawer (2009b) typology. According to Wheelwright & Clark (1992, p. 96), "platform projects create products and processes that subsequent development efforts can expand and enhance through the addition or removal of incremental features, thus creating a product and process family." They are "adaptable and expandable." Other definitions from the product development literature include Meyer & Lehnerd (1997, p. 7): "a set of common components, modules, or parts from which a stream of derivative products can be efficiently created

[28] This section follows the language conventions of the platforms literature stream. In previous sections, Teece (2009), Moore (2006), and others had rejected "industry" as a useful unit of analysis for the environments in which business ecosystems are expected to thrive. The platforms literature tends to use the term loosely to refer to *some* level of analysis above the firm, not necessarily a traditional industry sector bounded by an SIC (specific industry code). In fact, the example in this paragraph shows Gawer seemingly using the terms "industry" and "ecosystem" synonymously. These arguments about problem definition and managerial mental models are related to those of managerial language and metaphor examined in chapter 2.

and launched," and Robertson & Ulrich (1998, p. 20): "the collection of assets that are shared by a set of products." In the technology strategy literature, the emphasis was on control and rent extraction. Bresnahan & Greenstein (1999, p. 4) defined a platform as "a bundle of standard components around which buyers and sellers coordinate efforts," and West (2003) defined it as an architecture of related standards, allowing modular substitution of complementary assets such as software and peripheral hardware. In the industrial economics literature, a platform mediates transactions between two or more groups of agents (Rochet & Tirole, 2003); it is an alternative to the traditional "merchant" business model of selling a product or service to a customer (Hagiu, 2007). This literature has tended to emphasize the discovery of optimal pricing strategies to maximize some outcome variable – growth, the short-term profits of the platform owner, or some combination (Hagiu, 2009). Rochet & Tirole (2006, p. 645) write: "Two-sided (or, more generally, multi-sided) markets are roughly defined as markets in which one or several platforms enable interactions between end-users and try to get the two (or multiple) sides 'on board' by appropriately charging each side. That is, platforms court each side while attempting to make, or least not lose, money overall." Baldwin & Woodard (2009) argue that these three literatures offer different perspectives on the same underlying phenomenon: "a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components" (p. 19). The fundamental architecture is a system partitioned into a set of 'core' components with low variety and a complementary set of 'peripheral' components with high variety. The low-variety components that constitute the platform are the long-lived elements of the system that implicitly or explicitly establish the system's interfaces – the *rules* governing interactions among different parts[29].

An early body of work in platform strategy examined *platform leadership*, defined by Cusumano & Gawer (2002) as the ability of a company to drive innovation around a particular platform technology at the broad industry level. In this view, *platform leaders* are "companies that drive industry-wide innovation for an evolving system of separately developed pieces of technology" (p. 52). *Wannabes* are companies that want to be platform leaders. *Complementors* (following Brandenburger & Nalebuff, 1996) are companies that make ancillary products that expand the platform's market. Some companies occupy multiple roles; for example, Intel and Microsoft are both platform leaders and complementors. Platform leaders employ the *four levers of platform leadership* to maintain and extend a leadership position (Gawer & Cusumano, 2002): (1) *scope* (decisions of which complements to make in-house and which to deliberately leave to other companies), (2) *product technology* (decisions of modularity, interfaces, and how much information to disclose), (3) *relationships with external complementors* (decisions around consensus and control, cooperation and competition, and handling potential conflicts of interest), and (4) *internal organization*. Platform leaders face three types of problems (Cusumano & Gawer, 2002): (1) how to maintain the integrity of the platform in the face of future technological innovation and the actions of other companies, (2) how to let the platform evolve while maintaining compatibility with past complements, and (3) how to maintain platform leadership. Complementors may

[29] The notion of platforms as a bundle of rules is developed more fully in section 3.4.

rationally fear the platform leader's ability to "squeeze" them out *ex post* if they are successful; Gawer & Henderson (2007) examine the ways in which Intel employs the levers of platform leadership to partly assuage this concern[30]. Platform wannabes seeking to become platform leaders must overcome technological and business challenges (Gawer & Cusumano, 2008). The technological challenges include designing the right architecture, designing the right interfaces, and disclosing IP selectively to encourage provision of complements. The business challenges include either making key complements or introducing incentives for other companies to become complementors. Gawer & Cusumano (2008) identify two generic strategies for aspiring platform leaders – *coring* to create a new platform where none existed before, and *tipping* to win against competing platforms by building market momentum. Gawer (2009b) argues that the appropriate entry strategy and strategic mix of coring and tipping depends importantly on the characteristics and ambitions of the firm and the circumstances within the target industry, and offers a preliminary conceptual framework.

A smaller literature examines the strategies for platform complementors. Cusumano & Gawer (2002) offer five managerial prescriptions for platform complementors: (1) focus on products that the platform leader is unlikely to offer, (2) be aware that changes occur rapidly, thus work on continuous communication, seek early

[30] This "complementors' dilemma" has much in common with the Gans & Stern (2003) market for ideas. Entrants without strong excludability whose value propositions depend on complementary assets controlled by incumbents operate in a quadrant that Gans & Stern (2003) call "reputation-based ideas trading." Success for the entrant depends on strategic choices by complementary asset holders to not directly compete in ways that appropriate for themselves all of the value created; entry decisions for the entrant thus depend on entrant expectations of incumbent future behaviour. The existence of well-specified platforms extends the Gans & Stern (2003) framework to situations with no literal "ideas trading" between entrants and incumbents – just strategic entry decisions and competitive responses. See also the Iansiti & Levien (2004a, 2004b) distinction between ecosystem keystones and dominators (chapter 2).

information, and pay attention to actions of the platform leader, (3) react quickly to demands of the platform leader (give no excuses to compete), (4) create products that enhance the value of the core product ("have peanuts to offer the elephant", p. 54), and (5) work with groups inside the platform company that are likely to offer the most neutral stance to promote the platform.

Because all platforms require complementary innovations to be useful, no platform is fully under the control of its originator (Gawer & Cusumano, 2008). Nonetheless, some platforms are more tightly controlled than others. West (2003, p. 1259) reflects on the tension between appropriability and adoption evident in the "hybrid" platform strategies of Apple, IBM, and Sun Microsystems:

> *To recoup the costs of developing a platform, its sponsor must be able to appropriate for itself some portion of the economic benefits of that platform. But to obtain any returns at all, the sponsor must get the platform adopted, which requires sharing the economic returns with buyers and other members of the value chain. The proprietary and open source strategies correspond to the two extremes of this trade-off. In making a platform strategy for the 21st century, leading computer vendors face a dilemma of how much is open enough to attract enough buyers while retaining adequate returns.*

Selecting the level of platform *openness* is a crucial decision for firms that create and maintain platforms (Eisenmann et al. 2009). Opening a platform can spur adoption by harnessing network effects, reducing users' concerns about lock-in, and stimulating production of differentiated goods that meet the needs of user segments. At the same time, opening a platform typically reduces users' switching costs and increases competition among platform providers, making it more difficult form them to appropriate rents from the platform. Schilling (2009) identifies three dilemmas facing firms that liberally diffuse technology to would-be competitors: (1) they relinquish the opportunity to capture monopoly rents when and if their technology emerges as a dominant design, (2) once relinquished, control can be very hard to regain, and (3) potential for fragmentation of the technology platform. A point of consensus among platform researchers is that open and closed is not a dichotomy (West, 2003; Eisenmann et al. 2009; Schilling, 2009). Schilling (2009) argues that there are subtly nuanced benefits of different positions that a firm could adoption along a "control continuum" between monopoly control and liberal diffusion. Eisenmann et al. (2009) argue that platform openness occurs at multiple levels depending on whether participation is unrestricted at the (1) demand-side-user (end-user), (2) supply-side user (application developer), (3) platform provider or (4) platform sponsor levels. Horizontal strategies for managing openness entail licensing, joint standard setting and technical interoperability with rival platforms. Vertical strategies for managing openness entail backward compatibility, platform and category exclusivity, and absorption of complements. Each strategy grants or restricts access for one of the four platform participants.

All of the literature examined thus far has considered platforms with a focal "platform leader" firm seeking to appropriate value. The platforms literature is nearly silent regarding vendor neutral platforms and not-for-profit ecosystem keystones – the central subject of this research.

Recent work on multi-sided platforms draws on a formal technical base of economic theory[31] (e.g., Rochet & Tirole, 2003, 2006; Armstrong, 2006; Hagui, 2009) to examine (1) how to get the different sides of an platform 'on board' in large numbers, and

[31] Eisenmann et al. (2006) and Eisenmann (2008) both provide accessible non-mathematical entry points to this literature; nonetheless, the underlying foundation work is highly technical.

(2) pricing arrangements to maximize the profits of the platform leader. Boudreau & Hagiu (2009) focus on the importance and variety of 'non-price instruments' that platform owners us to regulate and govern their platform. According to this view, multi-sided platforms play a regulatory role that goes beyond price setting of arms-length transactions to include imposing rules and contraints, creating inducements and otherwise shaping behaviours. Other roles include (1) ensuring coherent technical development and coordination among contributors to a multi-sided platform ecosystem, (2) designing the technical architecture that frames interaction, (3) encouraging complementors to make investments, (4) generally managing and maintaining the health of the ecosystem. These various non-price instruments solve what would otherside be multi-sided market failures. Evans (2009) examines the strategies of entrepreneurs who start multi-sided platforms and must secure a 'critical mass' on each side of the platform to ignite growth, otherwise their platform implodes. Strategies include the 'zig-zig' (successive accretions of customers on both sides to build up the value to both sides), and the 'two-step' (getting enough members on one side first, then members of the other side). The relevant strategies depend in large part on whether the nature of the platform requires securing participation by both platform sides at launch, whether it is possible to acquire one side before approaching the other side, and whether it is necessary to make pre-commitments to one side to induce them to make investments.

In summary, the platforms literature focuses on the shared technical base of the business ecosystem, and its relationships to organization, strategy, and outcomes. The salient lessons from the platforms literature include the following. (1) A platform is a set of stable components that supports variety and evolvability in a system by constraining the linkages between other components (Baldwin & Woodard, 2009, p. 19); it is a building block (or set of building blocks) on which others can develop products, services or technologies that complement the platform (Gawer, 2009a). (2) Platforms are often subject to network effects that amplify early advantage; past a tipping point, a platform becomes difficult to dislodge. (3) Multi-sided platforms mediate transactions between two or more sides; each side is comprised of platform stakeholders that create and appropriate value in similar ways. (4) Benefits accrue to the platform leader who controls a platform. (5) Platform leaders employ the four levers of platform leadership to maintain and extend a leadership position: scope, product technology, relationships with complementors, and internal organization (Gawer & Cusumano, 2002). (6) Platform complementors develop products, services, and technologies that complement the value of the platform. (7) Platforms may be open or closed to varying degrees; selecting the level of openness is a crucial decision with an inherent tension between adoption (driven by openness) and appropriability (enabled by high levels of control by the platform leader). (8) Much of the platforms literature focuses on platform leadership; it is nearly silent regarding community-developed and vendor neutral platforms.

Despite differing language conventions, the platforms literature has much in common conceptually with the ecosystems literature. Both literatures recognize the importance of a level of analysis above the firm, and the central importance of the shared technological base and the shared operating conventions of participants. The sides of a multi-sided platform resemble the niches of a business ecosystem. The platform roles of leader and complementor (Cusumano & Gawer, 2002) resemble the ecosystem roles of keystone and niche player (Iansiti & Levien, 2004a, 2004b).

3.4 Modularity and design

Modularity is a central construct in several research domains, including software engineering (MacCormack et al. 2006; Shaw & Garland, 1996), product design (Ulrich, 1995), organization theory (Sanchez & Mahoney, 1996), and industry structure (Christensen et al. 2001; Langlois & Robertson, 1992). According to Baldwin & Clark (2000, p. 2), modularity links the microstructure of these different domains, deep "in the very nature of things," in ways that enable theorizing about the fundamental nature and extent of their interconnectedness (see also Garud et al. 2003; Simon, 1996; Henderson & Clark, 1990). Moore (2006) was aware of this literature, and its influence is evident in his thinking about the co-evolution of interconnected technological, organizational, and business systems within business ecosystems.

Simon (1969, pp. 5-6) introduces the notion of the *artificial* – things that are "synthesized (though not always or usually with full forethought) by man." Artificial things "may imitate appearances in natural things while lacking, in one or many respects, the reality of the latter," they "can be characterized in terms of functions, goals, adaptation," and they "are often discussed, particularly when they are being designed, in terms of imperatives as well as descriptives." Products and platforms, organizations, and higher-level organizational structures such as industries and business ecosystems are all artificial things. All are *hierarchical systems* with the property of *near-decomposability*. A *hierarchy* is "a system that is composed of interrelated subsystems, each of the latter being in turn hierarchic in structure until we reach some lowest level of elementary subsystem" (Simon, 1996, pp. 184-185). It is a "parts-within-parts structure" (p. 186) – "partitioning in conjunction with the relations that hold among its parts" (p. 185)[32]. *Near decomposability* means that interactions *among* subsystems are much weaker than interactions *within* subsystems. It simplifies both the behaviour of a system, and the useful description of a system; although system properties are *emergent* – not *fully* comprehensible from an understanding of lower level components – close study of nearly-decomposable system components can provide *some* insights into the operation of the system[33]. An important subset of nearly decomposable systems are *modular* – "composed of units (or modules) that are designed independently but still function as an integrated whole" (Baldwin & Clark, 1997, p. 86)[34].

Baldwin & Clark's theory of *design rules* (Baldwin & Clark, 2000) was developed to explain changes in the computer industry between 1944 to 1980, which transformed from a quasi-monopoly of a few incumbents to a diverse *modular cluster* of many firms. Baldwin & Clark (1997, p. 84) write:

[32] This is the same conceptual argument underlying ecological hierarchy in chapter 2.

[33] In contrast, a fully-decomposable system would have no emergent properties, and a non-decomposable system would have only emergent properties – knowledge of subsystems would provide no information about the system. Simon's (1969) nearly-decomposable systems are closer to the former than the latter. Simon (1996, pp. 188-190) employs the *Parable of Two Watchmakers* to explain the evolution of complex systems and the advantage conferred by near decomposability. Tempus and Hora each assembled watches comprised of many parts. Hora used stable sub-assemblies in a hierarchical fashion and became rich. Tempus needed to assemble all parts at once to achieve a stable assembly and became poor. Hora incurred extra costs and effort in the ideal case (with no noise), but was more robust than Tempus to disturbances and interruptions in the actual non-ideal conditions in which their work occurred.

[34] Callebaut (2005, p. 22) warns that modularity and near decomposability of hierarchical systems are conceptually close but not identical concepts. Many complex systems do display both properties, but they are not the same. Examples of modular systems without hierarchy include a system of rooms connected by corridors, and the bones in a limb. The hierarchy in these systems is of a different sense (e.g., molecules, cells, tissues, etc.). Garud et al. (2003) treat modularity as a subset of nearly decomposable systems – a subset of central importance to management and strategy.

> *Through the widespread adoption of modular designs, the computer industry has dramatically increased its rate of innovation. Indeed, it is modularity, more than speedy processing and communication or any other technology, that is responsible for the heightened pace of change that managers in the computer industry now face. And strategies based on modularity are the best way to deal with that change.*

Design rule theory builds on well-established ideas in architectural design (Alexander, 1964), engineering design (Marple, 1961; Simon, 1962; Steward, 1981; Eppinger, 1991; Eppinger et al. 1994) and software engineering (Parnas, 1972, 1974; Parnas et al. 1985). Its central argument is that modularity in design alters the mechanisms by which designs can change. This enables *design evolution* – a value-seeking process with strong parallels to biological processes. The fundamental agents of design evolution are not blind replicators (like genes) but rather human beings capable of seeking value in new designs. Just as modularity enables design evolution, design evolution enables industry evolution. Table 6 (adapted from Baldwin & Clark, 2000, Table 9.1) contrasts these three forms.

Table 6: Design and industry evolution (adapted from Baldwin & Clark, 2000)

Element	Genetic evolution	Design evolution	Industry evolution
Units selected	• Genes • Organisms	• Artifacts • Designs • Task structures	• Enterprises • Managements
Selection criterion	Fitness (replication or reproduction)	Value to users[35]	Capital market value
Source of variation	Genetic operators: • Crossing over • Inversion • Mutation • Dominance • Segregation • Translocation • Duplication	Modular operators: • Splitting • Substitution • Augmentation • Exclusion • Inversion • Porting	Industry operators: • Entry • Exit • Acquisition • Divestiture • Hiring • Firing
Selection mechanisms	Survival and reproduction	Realization, purchase, and use	Survival; selection for inclusion in a portfolio of investments

[35] The original formulation of this table in Baldwin & Clark (2000, p. 225) has capital market value as the selection criterion for design evolutions. This was useful for producer innovation (section 3.1), but unhelpful in the broader circumstances of other innovation modes (section 3.2). In a later study of open source software architecture, Baldwin & Clark (2006) employed "value as perceived by users" as the selection criteria for design evolution. Table 6 incorporates these later refinements.

Formally, design rule theory posits that designs, markets, and industries are interconnected in a *complex adaptive system* (Holland, 1992; Miller & Page, 2007). Decomposition of a system into modules partitions information into *visible design rules* and *hidden design parameters*. *Visible design rules* fall into three categories: (1) an *architecture* specifies what modules will be part of the system and what their functions will be, (2) *interfaces* describe in detail how the modules will interact, including how they will fit together, connect, and communicate, and (3) *standards* specify how to test conformity to design rules and measure relative performance. *Hidden design parameters* are encapsulated within modules – they are not communicated beyond the boundaries of a module. *Modular operators* form a basic repertory of actions that can be performed in modular systems. Complex changes in a modular system can be represented as combinations of operators, and the value of specific operator-moves can be modeled using real options methods from finance. Modularity simultaneously multiplies and decentralizes design options (Baldwin & Clark, 2000, p. 14):

> *The multiplication occurs because changes in one module become independent of changes in other modules. Decentralization follows because, as long as designers adhere to the design rules, they are free to innovate (apply the modular operators) without reference to the original architects or any central planners of the design.*

The context in which designs and design processes are lodged operates on designs "like a force" (Baldwin & Clark, 2000, p. 12):

> *In particular, economies with capital markets offer large, direct rewards to value-creating enterprises, and commensurately large incentives for human beings to cooperate for the purpose of creating economic value.... Metaphorically, they 'pull' designs in the 'direction' of higher market value.*

In advanced economies, contract structures overlay design and task structures to define firms and markets, and guidance mechanisms direct and coordinate effort within collective enterprises. The *specifiable*, *verifiable*, and *predictable* interfaces at the architectural level determine the efficient placement of firm boundaries (Christensen et al. 2001; Christensen & Raynor, 2003a) and enable modular organizations (Langlois & Robertson, 1992; Sanchez & Mahony, 1996). A *modular cluster* is a group of firms and markets that "play host" to the evolution of a set of modular designs (Baldwin & Clark, 2000, p. 16). The modular cluster organizational form becomes a potentially viable industry structure following a large-scale modularization of a set of artifact designs.

Baldwin & Clark (2006) extend these arguments to open source software[36], arguing that the architecture of a software code base is a critical factor that lies at the heart of the open source development process. Drawing on their previous work on design rules, they argue that designs have option-value because a new design creates the right but not the obligation to adopt it. A modular design allows for experimentation and changes within modules without disturbing the functionality of the whole system. The authors then use a series of increasingly sophisticated game theory models of developer

[36] It is significant that the Baldwin & Clark (2000) study interval of 1944-1980 precedes the founding of the *Free Software Foundation* (1985), the release of the Linux kernel under the GNU Public License (1993), and the *Open Source Definition* (1998). The scope of their study of design rules in the computer industry instead includes the first computers, the dominance of the IBM System/360 mainframe computer in the 1960s, the DEC PDP-11 mini computer in the 1970s, and the development of workstation computers that became popular in the early 1980s. Lenfle & Baldwin (2007) write: "The open source development process … took Clark and many of his colleagues by surprise. Clark had been an early proponent of integrating users into a firm's product development process … but nothing in his prior experience led him to think that users might initiate and manage a product development process for themselves. Yet that was exactly what happened in open source development projects." Reflecting on the application of design rule theory to open source software, Baldwin & Clark write (Garud et al. 2003, pp. 168-169): "A cluster of firms and markets is by no means the only way to organize a modular system, nor is it necessarily the most efficient way to encourage modular design evolution… Open Source developers have managed to drive the principles of modularity deeper into their design hierarchies than any cluster of firms and markets – given their implicit coordination problems – would be able to do."

behavior to show that increased modularity (and thus increased option value) has two effects on the software development process. First, it increases the incentives of developers to get involved and remain involved in the development process. Second, it decreases the amount of free riding in the equilibrium[37]. Both effects promote growth of the developer community, suggesting that modular design is important to the success of open source development projects. Evidence from empirical studies supports a deep and positive connection between modularity and design evolution (MacCormack et al. 2006; LaMantia et al. 2008; Milev et al. 2009).

Baldwin & Woodard (2009) make important progress towards unifying this literature stream on modularity and design with the previous literature stream on platforms: "Although it is a recurring theme across diverse literatures, the reuse and sharing of core components of a platform is only half the story [of platforms]" (p. 22). It is *platform architecture* that unites the platform perspectives from the product development, technology strategy, and industrial economics literatures (p. 22-23):

> *In essence, a 'platform architecture' is a modularization that partitions the system into (1) a set of components whose design is stable and (2) a complementary set of components which are allowed – indeed encouraged – to vary. The combination of stability and variety is accomplished via 'stable, yet versatile' interfaces, which govern the interactions of components. The interface specifications are part of the platform; indeed they may be the only components that remain truly stable over long periods of time. The combination of stability and variety in the architecture makes it possible to create novelty without developing a whole new system from scratch. Thus platform systems are evolvable.*

[37] Economists define free riding as enjoying economic benefits without paying economic costs. Baldwin & Clark's game theory models consider using software without contributing to its development to be free riding. Software is a non-subtractable good (sections 5.2 and 5.3), so if there are network effects in use (as argued for platforms in section 3.3), then simply using the software without contributing to its development would contribute to its value; this view is widely held by practitioners in the open source software community (section 3.6). The key point, however – that the proportion of non-contributing users is lower with greater modularity and option value – is sill salient to community growth.

In summary, the modularity and design literature connects product design to organizational structure and to interorganizational structure; according to this view, the three levels of analysis co-evolve together in their own selection environments, with each constraining and enabling the others. The salient lessons from the design literature include the following: (1) Modularity in design alters the mechanisms by which designs can change, enabling designs to evolve through variation and selection processes. (2) Modular designs can be formally specified as visible design rules (of architecture, interfaces, and standards) and hidden design parameters that are encapsulated within modules and are invisible from outside. (3) Design variation arises from designers employing modular operators on designs; design selection arises from user adoption, based on value provided to users. (4) Modularity simultaneously multiplies design options (because changes to the hidden design parameters of one module are independent of changes to another module) and decentralizes design options (because changes can proceed independently and in parallel without central control). (5) Designs, organizations, and industries co-evolve; design rules constrain and enable the organizational rules by which tasks are performed within organizations, and the organizational rules by which firms internalize activities or contract them out to others. (6) Modularity is at the heart of the open source software development process. (7) The design rules view of platforms as bundles of rules offers a potential link to institutional rules governing individual and organization behaviour in business ecosystems.

3.5 Interorganizational networks

Organizational scholars have studied interorganizational networks and interfirm relationships for more than four decades, producing a thick literature (Granovetter, 1973; Burt, 1980, 1992; Porter, 1980; Powell, 1990)[38]. Moore (1993, p. 75) writes:

> *Much has been written about such networks, under the rubric of strategic alliances, virtual organizations, and the like. But these frameworks provide little systematic assistance for managers who seek to understand the underlying strategic logic of change. Even fewer of these theories help executives anticipate the managerial challenges of nurturing complex business communities that bring innovations to market.*

This section is a brief and selective survey of the vast interorganizational networks literature, starting from the traditional distinction between markets and hierarchies, and the traditional explanations for the existence of relationships between firms, which together comprise the point of departure for the business ecosystems literature. It then surveys contributions from the management literature on network organizational forms, and recent applications of network analysis to interfirm networks.

A thick literature on transaction cost economics (TCE) recognizes two prototypical organizational forms: markets and organizational hierarchies (i.e., firms). According to TCE, activities are internalized within an organizational hierarchy when the transaction costs of performing those activities within the firm are lower than the transaction costs of contracting those activities from the market (Williamson, 1975, 1985, 1991)[39]. In this view, firms are "islands of planned co-ordination in a sea of market

[38] Baker & Faulkner (2002, p. 520) call interorganizational networks "a venerable subject in sociology and organization theory." Nonetheless, the rise of the network organization form (as distinct from, for example, the multidivisional and conglomerate forms) is one of the contemporary debates and areas of active work.
[39] Oliver E. Williamson received the 2009 Nobel Memorial Prize in Economic Sciences (shared with Elinor Ostrom), for "his analysis of economic governance, especially the boundaries of the firm" (Williamson, 2009). See also footnote 23 and the survey of Ostrom's work on governance in chapter 5.

relations" (Richardson, 1972), and the reduction of transaction costs explains both the existence of firms and the location of firm boundaries.

Nonetheless, it has long been recognized that firms *do* form lasting relationships with other firms. Granovetter (1994, p. 453) writes:

> *In every known capitalist economy, firms do not conduct business as isolated units, but rather form cooperative relations with other firms, with legal and social boundaries of variable clarity around such relations. In no case do we observe an economy made up of atomized firms doing business at arms length with other firms across a market boundary.*

Williamson (1985, 1991) extended the TCE framework to consider the transaction costs of a variety of "hybrid" organizational forms, such as franchising or alliance arrangements. The hybrid approach has spawned a thick theoretical literature that treats markets and hierarchies as poles, and other forms of organization as intermediaries between these two poles, with Makadok & Coff (2009) as a recent example.

Powell (1990, p. 298) challenged both the market-hierarchy dichotomy (of early TCE) and the market-hierarchy continuum (of later TCE) and other hybrid theories:

> *I do not share the belief that the bulk of economic exchange fits comfortably at either of the poles of the market – hierarchy continuum... [A]lthough I was earlier of the view that nonmarket, nonhierarchical forms represented hybrid modes (Powell, 1987), I now find that this mixed mode or intermediate notion is not particularly helpful. It is historically inaccurate, overly static, and it detracts from our ability to explain many forms of collaboration that are viable means of exchange....*
>
> *By sticking to the twin pillars of markets and hierarchies, our attention is deflected from a diversity of organizational designs that are neither fish nor fowl, nor some mongrel hybrid, but a distinctly different form.*

For Powell, the salient features of network forms of organization are reciprocal patterns of communication and exchange: "transactions occur neither through discrete exchanges nor by administrative fiat, but through networks of individuals engaged in reciprocal, preferential, mutually supportive actions.... In essence, the parties to a network agree to forgo the right to pursue their own interests at the expense of others" (p. 303). Powell (1990, p. 300) offers a "first cut at summarizing some of the key differences among markets, hierarchies, and networks," summarized in Table 7. For comparison, Table 8 summarizes Moore's (2006) arguments on the differences between markets, hierarchies, and business ecosystems (reviewed in chapter 2).

Table 7: Markets, hierarchies, and networks (adapted from Powell, 1990, Table 1)

		Forms	
Key Features	**Market**	**Hierarchy**	**Network**
Normative basis	Contract – property rights	Employment relationship	Complementary strengths
Means of communication	Prices	Routines	Relational
Methods of conflict resolution	Haggling – resort to courts for enforcement	Administrative fiat – supervision	Norm of reciprocity – reputational concerns
Degree of flexibility	High	Low	Medium
Amount of commitment among the parties	Low	Medium to high	Medium to high
Tone or climate	Precision and/or suspicion	Formal, bureaucratic	Open ended, mutual benefits
Actor preferences/choices	Independent	Dependent	Interdependent
Mixing of forms	Repeat transactions; Contracts as hierarchical documents.	Informal organization; market-like features: profit centers, transfer pricing.	Status hierarchies Multiple partners Formal rules

Table 8: Markets, hierarchies, and business ecosystems (adapted from Moore, 2006)

		Forms	
Key Features	**Market**	**Hierarchy**	**Business ecosystem**
Function	Facilitate transactions for goods	Facilitate control over activities that produce goods	• Facilitate coordination of innovation in goods and the activities that produce them. • Facilitate managed co-evolution of the complex web of markets and hierarchies.
Ideal relationships sought	Perfectly transparent transactions	Perfect control	Perfect co-evolution of innovation
Levels of analysis to judge performance	Goods and services	Tasks and activities	A multitude of contributors
Guidance	The invisible hand of the market (Smith, 1776)	The visible hand of the professional manager or entrepreneur (Chandler, 1977)	The guiding hand of community leaders

According to sociologist Manuel Castells (2000), the *network enterprise* is the prevailing organizational form in the global networked *informational* economy (p. 199), where "successful organizations are those able to generate knowledge and process information efficiently; to adapt to the variable geometry of the global economy; to be flexible enough to change their means as rapidly as goals change, under the impact of fast cultural, technological, and institutional change; and to innovate, as innovation becomes the key competitive weapon" (p. 188). Networked enterprises differ from the vertically integrated corporations and the independent small business firms of the production economy; the system of means is not within the firm, but rather "the intersection of segments of autonomous systems of goals" (p. 187). Rather than one driving the other, the relationship between organization change and technological change has been reciprocal and reinforcing. It is "the convergence between organizational requirements and technological change [that] has established networking as the fundamental form of competition in the new, global economy" (p. 186). Moore (2006) later echoes arguments similar to Powell (1990) and Castells (2000) in the business ecosystem literature – that business ecosystems are not a hybrid existing between markets and firms, but rather an alternative means of production distinct from markets and hierarchies.

Table 9 summarizes a diverse variety of constructs devised by management scholars to label, categorize, and describe specific forms of interfirm relationship.

Table 9: Inter-firm constructs from the management literature

Construct	Definition
Value chain	"The *value chain* disaggregates a firm into its strategically relevant activities in order to understand the behavior of costs and the existing and potential sources of differentiation. A firm gains competitive advantage by performing these strategically important activities more cheaply or better than its competitors" (Porter, 1985, pp. 33-34, emphasis added). It is "a system of interdependent activities" (p. 48). A *value system* is the "larger stream of activities" (p. 34) that includes the value chains of suppliers, the firm, channels, and buyers.
Value net (game theory)	The *value net* is "a visual representation of the game of business" (Brandenburger & Nalebuff, 1996, p. 9), comprised of the focal firm, customers, suppliers, competitors, and complementors. A player is a *complementor* if customers value the focal company's product more when they have the other player's product than when they have the company's product alone.
Value network (disruptive innovation)	The *value network* is "the context within which a firm identifies and responds to customers' needs, solves problems, procures input, reacts to competitors, and strives for profit" (Christensen, 1997, p. 32). "Companies are embedded in value networks because their products generally are embedded, or nested hierarchically, as components within other products and eventually within end systems of use" (p. 32). Past choices shape perceptions.
Value constellation (open innovation)	*Value constellations* are "interorganizational networks linking firms with different assets and competencies together in response to or in anticipation of new market opportunities" (Vanhaverbeke & Cloodt, 2006, p. 259; see also Normann & Ramirez, 1993)
Business web (b-web)	A *business web* is "a distinct system of suppliers, distributors, commerce service providers, infrastructure providers, and customers that use the Internet for their primary business communications and transactions" (Tapscott et al. 2000, p. 2).
Value net (supply chain)	A *value net* is "a business design that uses digital supply chain concepts to achieve both superior customer satisfaction and company profitability" (Bovet & Martha, 2000, p. 2).
Community of firms	A *community of firms* is "a new organizational form ... [that] reflects a new breed of inter-firm collaboration" (Snow et al. 2009, p. 69) modeled after innovative communities of individuals.
Community of practice	*Communities of practice* are "groups of people informally bound together by shared expertise and passion for a joint enterprise" (Wenger & Snyder, 2000, p. 129).
Business group	A *business group* is a "collection of firms bound together in some formal and/or informal ways" (Granovetter, 1995, p. 95), that is "legally independent, with economic (such as ownership, financial, and commercial) and social (such as family, kinship, and friendship) ties" (Yiu et al. 2005, p. 183). Examples: Keiretsu (Japan), Chaebol (South Korea).
Modular cluster	A *modular cluster* is a group of firms and markets that "play host" to the evolution of a set of modular designs (Baldwin & Clark, 2000, p. 16). It becomes a potentially viable industry structure following a large-scale modularization of a set of artifact designs.
Network enterprise	The *network enterprise* is "the prevailing organizational form in the information economy" (Castells, 2000, p. 199). It is "that specific form of enterprise whose system of means is constituted by the intersection of segments of autonomous systems of goals" (p. 187). "The network enterprise makes material the culture of the informational, global economy: it transforms signals into commodities by processing knowledge" (p. 188).
Network organization	"Network forms of organization – typified by reciprocal patterns of communication and exchange – represent a viable pattern of economic organization" (Powell, 1990, p. 295).
Regional cluster	Clusters are "geographic concentrations of interconnected companies and institutions in a particular field" (Porter, 1998, p. 78). They are "a robust organizational form that offers advantages in efficiency, effectiveness, and flexibility" (pp. 80). "Proximity ... amplifies all of the pressures to innovate and upgrade" (p. 90).
Strategic Alliance	*Strategic alliance* is an umbrella term for a variety of interfirm partnerships, including joint ventures, equity block ownership, licensing, supplier agreements, outsourcing, R&D partnerships, marketing partnerships, networks, and communities of firms (Culpan, 2009).

Organizational scholars have produced a large and growing literature on networks. Scott & Davis (2007, p. 278)[40] write:

> *Network has become perhaps the dominant metaphor of our time, used to describe structures from the brain (neural networks) to malevolent social groupings (terrorist networks) to global computer communication systems (the World Wide Web). Defined simply, a network consists of "nodes" and "ties", or relationships among the nodes. Nodes can be actors – such as persons, groups, or organizations – or other entities, such as neurons or abstract ideas. Ties can take on endless forms, from physical linkages to personal relationships.*

Research on interorganizational networks is "broad and eclectic" (Baker & Faulkner, 2002, p. 520) with much confusion around the appropriate unit of analysis and domain of activity. A salient theme for business ecosystems research is techniques to map, visualize, measure, and compare interorganizational networks. Scott & Davis (2007, p. 280) introduce some basic concepts of network analysis:

> *A network is a system of relationships among parts. The parts are generally referred to as* nodes, *and the relationships or connections among the parts are* ties. *The underlying premise is that ties are often more influential in affecting behavior than the specific attributes of nodes.*

Nodes are typically social actors, such as individuals or organizations, but could alternatively be technologies or other interconnected entities. Useful analytical levels include the *ego network* of direct contacts to a focal node, the *overall network* of all nodes and ties within a particular domain, and the *network position* of a particular node within the larger topography. The meaning of different network measures depends entirely on the nature of the relationship being mapped (p. 281). Potentially useful measures include *distance* (the shortest path between two nodes), *centrality* (the position

[40] The recent growth of the interorganizational networks literature is evident in Richard Scott's popular organization theory text: the 5th edition (Scott, 2003) had two pages on joint ventures, alliances, and mergers (pp. 206-207), and three pages on network forms of organization (pp. 281-283); the revised edition (Scott & Davis, 2007) has a 30-page chapter on the networks in and around organizations.

or importance of a node), *clustering* and *structural holes* (how connected or cohesive neighborhoods are), *equivalence* (nodes sharing the same patterns of relationships), *density* (the extent to which nodes that could be connected really are connected), and *centralization* (the extent to which some actors are well connected and others are not)[41].

Iansiti & Levien (2004b, p. 71) argued against precisely mapping the business ecosystem: "Drawing the precise boundaries of an ecosystem is an impossible and, in any case, academic exercise. Rather you should try to systematically identify the organizations with which your future is most closely intertwined and determine the dependance that are most critical to your business." Nonetheless, several academics have applied network analysis techniques to map networks of ecosystem relationships with varying levels of success. Iyer et al. (2006, p. 42) write:

> *The business ecosystem has emerged as the new referent for strategy formation. Hence, mapping the ecosystem in terms of inter-firm interconnections is important in order to understand how firms access complementary resources through relationships. A diagrammatic representation of the ecosystem with key relationships provides the context for strategy formation and implementation.*

They recommend that managers diagram and quantitatively "dashboard" the historical patterns of their ecosystem in order to better "comprehend, analyze, and act on moves and counter-moves by different players" (p. 47). As a demonstration, Iyer et al. (2006) construct and examine a database of major alliances and relationships between 509 companies within SIC 7372 (design and deliver packaged software) between 1990 and 2002, and conclude that the software sector operates as a *small world* network[42] (Watts,

[41] Goyal (2007) provides an alternative framework, starting from economics rather than network theory.

[42] A small world network is characterized by a high degree of clustering and short path lengths between any two nodes. In popular culture, this sort of network is better known through the expression "six degrees of separation," from the results of a study by Milgram (1967).

1999, 2003). Weiss & Gangadharan (2010) employ similar techniques to examine the ecosystem of mashups – software that combines data and services provided by others through open interfaces with data provided by users. Li (2009) employs network analysis within a broader study of the Cisco business ecosystem.

In summary, the organization literature on interorganizational networks has two distinct salient streams. The first stream is anchored in the traditional distinction between markets and hierarchies, along with traditional studies of strategic alliances, joint ventures, and partnerships; it provides a useful foil for the emerging literature on business ecosystems, which rejects many of its assumptions and conclusions. The second stream acknowledges that interorganizational networks exist as an empirical phenomena, and seeks to understand their origins and properties; it is a vast and fragmented literature, with many rival constructs and perspectives. The salient lessons from the organization literature include the following. (1) Traditional explanations for the existence of markets, organizational hierarchies, relationships between firms, and inter-organizational networks cannot adequately account for business ecosystems. (2) Research on interorganizational networks is broad and eclectic; consensus has not yet emerged with regards to the appropriate level of analysis, constructs, and measures. (3) Network analysis could be useful for mapping and visualizing the structure of a business ecosystem, however the data to do this is often difficult to obtain.

3.6 Communities

"Community" is a term with many different meanings; West & Lakhani (2008, p. 224) observe:

> *a welter of overlapping literatures and terms: innovation communities, knowledge producing communities, online communities, scientific communities, technical communities, user communities, virtual communities, or communities of practice. That doesn't even include the disparate uses of 'community' in sociology, where Brint (2001) recounts that some 100 different definitions have been used.*

This research takes particular care to specify its community constructs, to employ those definitions consistently, and to employ alternative labels for different concepts.

A useful umbrella definition, adapted from *The American Heritage Dictionary* (2009), is a group of people having common interests and a similar sense of identity; the etymology is Old French and Latin roots meaning citizenry, fellowship, and common. Bacon (2009, p. 7) associates community with a shared core belief, a sense of belonging, a collection of shared processes, and a social economy whose currency of social capital rather than financial capital. "At the heart of how this movement works is *communication*" (Bacon, 2009, p. 8). Membership is voluntary: new members can join, and existing members can exit.

Scholars are divided on whether community membership is comprised exclusively of individual people, or whether community membership also admits higher level social structures such as groups or organizations (West & Lakhani, 2008). Community constructs comprised exclusively of individuals are common in sociology (Brint, 2001), organizational behaviour (Brown & Duguid, 1991; Wenger & Snyder, 2000; Amin & Roberts, 2008), and the meritocratic communities of open source software developers (examined later in this section). Chapters 7 and 8 will later argue that the "Eclipse developer community" – a central feature of the Eclipse field setting and a central construct within the explanatory theory developed in later chapters – is best understood as a community of individuals. Community members may also be members of other organizations, groups, and collectives, and these memberships may motivate individual behaviour within the community, but higher-level social actors are outside of the community (of individuals) construct.

Community constructs comprised of organizations, or comprised of both individuals and organizations, are also found in the management literature. Examples include Lynn et al. (1996), von Burg & Kenney (2003), Miles et al. (2005), von Hippel (2005), Culpan (2009), and Snow et al. (2009). Moore's (2006) definition of business ecosystem as a "community of economic actors" (Moore, 2006, p. 33; see also sections 2.3 and 2.4) is adapted within this document to a "collective of economic actors" except when directly quoting Moore in order to more clearly distinguish between the "developer community" and "business ecosystem" constructs central to the explanatory theory developed in later chapters. Furthermore, chapter 7 will later argue that the "Eclipse community" (with no other adjectives) is understood within Eclipse practitioner parlance to broadly encompass the "Eclipse developer community" of individuals that maintain and extend Eclipse software as well as the "Eclipse user community" of individuals and organizations that use Eclipse software and the "Eclipse adopter community" of individuals and organizations that extend Eclipse software. Thus, although the "Eclipse developer community" is comprised exclusively of individuals, both the "Eclipse user community" and "Eclipse adopter community", as well as the broadly inclusive "Eclipse community," are commonly understood by Eclipse practitioners to include organizations (section 7.5 elaborates on this argument with support from case evidence).

Other scholarly literatures and domains of practice can and do define community in different ways – a potential source of semantic confusion in interdisciplinary research. Two salient examples for this research are the community construct in ecology (section 2.2) and the community construct in the IAD literature (introduced in chapter 5). In the field of ecology, "community" is a narrow technical term referring to all biological populations of species occupying a geographical area (Table 10 in chapter 2) – a level of ecological organization immediately above the organism, and immediately below the ecosystem[43]. In the IAD framework, "community" is a technical term for the collective of social actors whose behaviour is structured by an institution (see, for example, Figure 2, Table 12 and Table 13 in chapter 5). "Community" faces many of the same definitional challenges as "ecosystem" discussed in chapter 2.

Within the vast community literature, two particularly salient streams are community innovation and open source software communities. The first stream examines how communities outside the boundaries of firms often play a role in creating, shaping and disseminating technological and social innovations (von Hippel, 1988, 2001; 2005; Chesbrough & Appleyard, 2007; West & Lakhani, 2008; West & O'Mahony, 2008; Bacon, 2009; Boudreau & Lakhani, 2009; Fichter, 2009; West & Bolgers, 2010) and providing valuable support to others (Lakhani & von Hippel, 2003; Franke & Shah,

[43] See also section 2.3 and footnotes 17 and 21 regarding language tensions between the management and ecology literatures.

2003). Nonetheless, communities are ignored in traditional business strategy (Chesbrough & Appleyard, 2007) and only "rarely to be found in the recent writing on open innovation" (West & Lakhani, 2008, p. 223). A small but growing literature explores how companies can benefit by cultivating and enabling communities of users and innovators (von Hippel, 2005; Dahlander & Wallin, 2006; Bacon, 2009). Boudreau & Lakhani (2009) examine the circumstances under which companies should organize outside innovation as collaborative communities rather than competitive markets. Baldwin & von Hippel (2009) examine the circumstances favouring single-user innovation and community-user innovation over producer innovation.

The second salient community stream is open source software and the communities of developers and users that form around successful open source software projects. Open source software refers to software projects distributed under *open source licenses* that allow users to use, modify, and redistribute the software[44]. It also refers to the processes by which the software is created and distributed[45]. West & O'Mahony (2008, p. 146) write:

> *[Open source software] communities are composed of individuals who collaborate toward a common goal but do not share a common employer and are not governed by an employment hierarchy. By using both online and offline means, open source software communities collectively produce software that is freely and publicly available – creating in effect a shared public good that can be used for either public or private purposes.*

[44] Philosophical and motivational differences distinguish *free software,* championed by the Free Software Foundation (FSF), and *open source software,* championed by the Open Source Initiative (ISI). Both organizations maintain separate, largely (but not entirely) compatible definitions and licensing guidelines. The history of this schism and the nature of these differences are discussed elsewhere (Raymond, 2001; Gay, 2002; Fitzgerald, 2006; Scacchi, 2007a, 2007b), but are not salient to this study.

[45] According to Weber (2004), the essence of open source is not the software; it is the process by which software is created. This is a common theme in the open source literature.

According to Fitzgerald (2006, p. 587), open source has altered the basic nature of the software industry:

> *On the supply side, fundamental changes have occurred to the development process, reward mechanisms, distribution of development work, and business models that govern how profit can be achieved. On the demand side, the alternatives traditionally available to organizations for software acquisition – buy or build – have been supplemented with another credible alternative.*

There is a modest but rapidly growing scholarly literature examining open source software and the communities that develop it[46]. Three salient themes within the open source software literature are examined below: (1) the motivation for individual participation, (2) the motivation for company participation, and (3) governance.

The first open source theme, individual motivation, is what Feller et al. (2005) call a "relatively mature research area." Early essays, grounded in personal experience of the authors, examined the motivation of "hackers" (Levy, 1984; Raymond, 2001; Gay, 2002) – a subculture of passionate computer programmers at the genesis of open source. Lerner & Tirole (2002) argued that developer participation was rational economic calculus – a cost-benefit analysis that recognized signaling incentives, career concerns, and ego gratification. Later studies employed survey methods (Ghosh et al. 2002; Hertel et al. 2003; Ghosh, 2005; Lakhani & Wolf, 2005; Roberts et al. 2006; Subramanyam & Xia, 2008) to discover a wide range of developer motivations – both intrinsic and extrinsic, pecuniary and non-pecuniary, and self-interested and socially motivated. Heterogeneous motivations include advancing career and personal development, self-

[46]Recent special issues in highly ranked journals include *Industry and Innovation* (April 2008), *Management Science* (July 2006), *Software Process: Improvement and Practice* (March/April 2006), *interactions* (November/December 2005), *First Monday* (October 2005), *IBM Systems Journal* (2005), and *Research Policy* (July 2003). Recent surveys include Scacchi (2007a), von Krogh & von Hippel (2006), Scacchi et al. (2006), and Feller et al. (2005).

determination, peer recognition, identification, self-promotion from periphery to core, and belief in the inherent value of free software (Scacchi, 2007a, 2007b). Feller et al. (2005, p. xix) write:

> *[Diversity of motivation is] an inherent source of strength within the [open source] community. By allowing individuals with multiple motivation types to coexist and collaborate, the [open source] community can and does attract a wide range of participants. Individuals can join for their own idiosyncratic reasons, and the [open source] community does not have to be overly concerned about matching motivations to incentives.*

An early study by Lakhani & Wolf (2005) found that more than half of open source software developers contributed code while at work. Fitzgerald (2006) and Perens (2007) argue that many open source software developers are employed by companies to develop open source software as part of their formal job assignment; on projects with active company involvement, those companies may employ most or even all active developers.

The second open source theme is motivation for company participation. Fogel (2006, p. xiii) describes the economic rationale as follows:

> *There are organizations in whose interest it is to have certain software exist, but that they don't need to sell copies, they just want to make sure the software is available and maintained, as a tool instead of a commodity.*

Nonetheless, large commercial organizations are not always well perceived within the open source community (Fitzgerald, 2006, p. 596). Practices that may be at odds with the open source community spirit and shared values include support for patents, subscription agreements with lock-in through confidential support bulletins, and relationships with groups perceived negatively by the community. West & O'Mahony (2008) argue that companies participate in technical communities for at least three reasons: (1) to interpret, support, extend and diffuse path-breaking innovations, (2) to further develop and refine innovations, and (3) to coordinate the work of firms and individuals. This is a young but active area of research (Grand et al. 2004; Dahlander & Magnusson, 2005, 2008; Henkel, 2006; Dahlander & Wallin, 2006; Dahlander, 2007; Feller et al. 2008). Researchers have identified a wide assortment of business models anchored around open source assets (Chesbrough & Appleyard, 2007; Dahlander, 2007; Dahlander & Magnusson, 2008). In some of these models, companies complete the "whole-product" (Moore, 1991) by adding elements of the offer, such as sales and marketing, in which developers may not be interested (Fitzgerald, 2006). West & O'Mahony (2005) distinguish between *organic communities* founded by individuals that grow through grass roots communications, and *synthetic communities* founded by companies and grow with strategic direction. West & O'Mahony (2008) revise these categories to emphasize current governance practices rather than founding circumstances. An *autonomous open source community* is presently independent of any one firm and community managed. A *sponsored open source community* is one where one or more privileged corporate entities control short- or long-term activities. Sponsors face a tension between control and growth.

The third open source theme is community governance, which has received comparatively less attention than other aspects of open source communities (West & O'Mahony, 2008, p. 148). The small literature on open source community governance includes mainly single case and multiple case studies: O'Mahony (2003, 2005, 2007), von Krogh et al. (2003), Shah (2006), Markus (2007), O'Mahony & Ferraro (2007), Sadowski et al (2008), and Mateos-Garcia & Steinmueller (2008). West & O'Mahony (2008) examine the communities surrounding twelve open source projects initiated by corporate sponsors (Sendmail, Berkeley DB, MySQL, PHP, Mozilla, Jikes, Darwin, OpenOffice, Eclipse[47], Helix, Chandler, Sugar) and a comparison group of five projects originating from autonomous communities (Apache, Gnome, Debian, Linux, and the Linux Standard Base). The authors distinguish between two dimensions of openness: *transparency* (allowing outsiders to follow and understand a community's production efforts) and *accessibility* (allowing external participants to influence the community's production efforts). Sponsors were more likely to offer transparency than accessibility; all sponsors in the sample worked to achieve transparency in their communities, but sponsors varied considerably in the importance they placed on providing accessibility to external parties. Sponsors considered *three design dimensions* that together created a specific *participation architecture* – a topic examined more closely in chapter 4. West & O'Mahony (2008, p. 165) argue that transitions in governance are particularly interesting and under-studied:

> *Communities may transition from autonomous to sponsored, as has happened when a community founder (usually a hobbyist programmer) forms a company to monetize the value of the code. Or they may transition from sponsored to autonomous, often as part of a larger transition from a proprietary software project to a sponsored open source community to an autonomous community. Both Mozilla (now Firefox) and Eclipse went through such transitions, but they have received little empirical examination. What affects the evolution of such projects? What consequences do these changes have for the code, the community, and for innovation in general?*

Section 1.2 previously identified the transition of Eclipse from a sponsored community controlled by IBM to an autonomous community with vendor-neutral governance as one of several factors that distinguished Eclipse as unusual and interesting, and motivated its selection as the field setting for this research.

[47] Section 7.3 examines salient findings from West & O'Mahony (2008) regarding the Eclipse Foundation.

In summary, the salient lessons from the community literature include the following: (1) Communities can be usefully defined as *voluntary groups of individuals with a common interest, sense of belonging, and shared identity.* (2) Communities can create, shape, and disseminate innovation, and provide valuable support. (3) Companies can benefit by cultivating and enabling communities of users and innovators.
(4) Community governance, particularly the participation and sponsorship of companies in community governance, is not well understood. (5) Company sponsors of open source software communities consider three community design dimensions – intellectual property rights, production, and governance – and the transparency and accessibility of parameters within each dimension. (6) Sponsors of communities confront a tension between openness (to promote participation) and control. (7) Sponsored communities can transition to autonomous communities, but this process is not well understood.

3.7 Summary and synthesis

Six empirically-grounded management literatures – innovation, open innovation, platforms, modularity, interorganizational networks, and community – each inform some aspect of empirical inquiry into business ecosystems. With respect to the business ecosystems literature of the previous chapter, each of these six management literatures is at once both broader and narrower in scope; broader because each examines phenomena also present outside of business ecosystems, and narrower because each neglects phenomena central to the workings of a business ecosystem. Business ecosystems exist at the *nexus* of these six literatures, but each literature is incomplete on its own.

The innovation literature provides a theoretical underpinning for empirical inquiry into the processes and forms of value creation and appropriation within business ecosystems. The open innovation literature elucidates the motivation for firms to open up their R&D processes, the roles adopted and the mechanisms used to appropriate value, and the outcomes achieved. The platforms literature examines a possible anchor point of a business ecosystem and the interactions between companies and platforms. The modularity literature provides a theoretical underpinning to the "co-evolution of innovation across a multitude of contributors" (Moore, 2006, p. 74) by connecting the design rules of the platform to the organization of tasks and the structure of business organizations, and to the structure of interactions between firms. The interorganizational networks literature provides constructs and tools to examine business ecosystem structure – the ecosystem members, roles, and relationships between members. The community literature examines the operation and governance by which innovation around a community-developed platform can occur. Not all business ecosystems benefit from community innovation, but an important subset do, including the Eclipse ecosystem.

Insights from this chapter provide theoretical building blocks to theorize and explain the results of detailed field work in later chapters. The next chapter draws similarly on arguments from the practitioner literature.

4 Business ecosystems as architectures of participation

Publicist Tim O'Reilly coined the expression "architecture of participation" to describe "the nature of systems that are designed for user contribution" and introduced it to the open source community in a series of talks, essays and blog posts (e.g., O'Reilly, 2001, 2004, 2005a, 2005b, 2005c). Within such systems, "users pursuing their own 'selfish' interests build collective value as an automatic byproduct" (O'Reilly, 2005a, p. 253). Positive network effects are achieved by design so that systems get better the more people use them.

O'Reilly appropriates the notion of system architecture from the engineering field where *architecture* refers to a high-level design of the over-all way in which the major components of a system fit together (Whitney et al. 2004; see also section 3.4). System architecture is important in engineering design for several reasons: it may be technically difficult to accomplish and require a broad skill set, it often determines (or places upper bounds on) systems performance and (lower bounds on) cost, and it may be difficult (or impossible) to change later thus imposing path dependence. These are common notions in software engineering, computer engineering, systems engineering, and elsewhere.

Building user participation into the architecture of a system is an evocative concept that relates the design of the technical system, the organization of the community, and the outcomes to be achieved. For researchers, it brings together established theories of design and product development that relate product architecture with organizational structure, theories of human motivation and decision making that emphasize incentives, and social theories of community formation and growth. For software practitioners, it succinctly makes explicit an accepted element of the open source community culture (Schweik, 2007; Shirky, 2008).

Architecture of participation arguments are central to this research for three reasons. First, the vocabulary and logic is prominent in Eclipse practitioner discourse and practitioner explanations for the operation of the Eclipse field setting (examined in chapter 7). Second, these arguments motivated salient scholarly research on participation by Baldwin & Clark (2006) and West & O'Mahony (2008), examined in section 4.2. Third, a main theoretical contribution of this research advances a theory of participation that extends Baldwin & Clark (2006) and West & O'Mahony (2008) to explicitly include the business ecosystem as a source for community resources (section 11.3).

This chapter examines the architecture of participation in three section. The first section parses the practitioner writing to unpack and make explicit the various implied heuristics. The second section summarizes progress made by researchers towards formulating an operationalized construct. The third section identifies the unique contribution of architecture of participation heuristics and theorizing to this research.

4.1 Practitioner heuristics

Management researchers have described O'Reilly's writing on the architecture of participation as “a collection of heuristics” (Baldwin & Clark, 2006; West & O'Mahony, 2008). This subsection examines articles by Tim O'Reilly (published in print and on-line between 2001 and 2005) and blogs at the O'Reilly Publishing website (posted between 2006 and 2008) to identify these heuristics and make them explicit.

O'Reilly's earliest writings on the architecture of participation examined the design characteristics of successful open source applications. In an editorial published in *Linux Magazine*, O'Reilly (2001) writes: "Well-designed open source projects have what you might call an architecture of participation, one in which the protocols between participating programs are well defined, so that the individual programs can work together despite being developed independently." Open source and interoperability "are inextricably linked, because open architectures, like those of Unix/Linux and the Internet, are what make bottom-up open source technology development possible." An architecture of participation is therefore "an Internet-style architecture that that allows … independent solutions to interoperate, both with each other and with proprietary offerings… What we don't need is a single, monolithic Linux and open source strategy."

Later writing emphasizes the characteristics of the user and developer community. O'Reilly (2003) writes:

> *What really distinguishes open source is not just source, but an 'architecture of participation' that includes low barriers to entry by newcomers, and some mechanism for isolating the cathedral from the bazaar. This architecture of participation allows for a real free market of ideas, in which anyone can put forward a proposed solution to a problem; it becomes adopted, if at all, by acclamation and the organic spread of its usefulness.*

The article goes on to argue that the most successful open source communities have a central "cathedral" that is tightly controlled, surrounded by an extensive "bazaar" that is significantly open[48]. Within the bazaar, individuals are empowered to innovate, and users "self-select the offerings that provide real value." Contributors from outside the core

[48] O'Reilly appropriates the cathedral and bazaar analogies from the writing of Raymond (2001), examined in section 3.6, which would be very familiar to his target audience.

community are valued and welcomed. The potential for users to *fork*[49] the project provides developers with strong incentives to be responsive to users. However, community participation is not an automatic byproduct of access to source code: there are unsuccessful projects where the source is open but the dynamics of the community do not welcome participation, and there is evidence of community participation occurring within commercially-motivated companies and within corporate-sponsored communities[50].

O'Reilly (2004, 2005a) is the clearest and most complete articulation of O'Reilly's mature ideas on the architecture of participation[51]. The essay credits several influences on the development its ideas, including Lawrence Lessig's (2000) study of the architecture of cyberspace (making the case that understanding the architecture of systems is necessary to understand their effects), Dan Bricklin's (2000) description of Napster as a "cornucopia of the commons" (in contrast to Hardin's "tragedy of the commons", section 5.2), and an unpublished portion of an interview with Linus Torvalds (DiBona et al. 1999) that suggests product architecture may be more important than source code. In the interview, Torvalds states: "I couldn't do what I did with Linux for Windows, even if I had the source code. The architecture just wouldn't support it."

[49] A *fork* is a split in the open source developer community that spawns competing parallel projects that cannot easily exchange code (Raymond, 2001, p. 72). Open source licenses permit forking, but forking is relatively uncommon in practice. Raymond (2001, p. 72) writes: "Splits in major projects have been rare, and are always accompanied by re-labeling and a large volume of public self-justification. It is clear... that the splitters felt they were going against a fairly powerful community norm." According to Fogel (2006), "forkability" is the indispensable ingredient that binds developers together and makes them willing to compromise. It is the *potential* to fork – not the act of forking – that motivates responsiveness.

[50] The examples provided in O'Reilly (2004) are the birth of Microsoft's ASP.net (which originated as a personal project of two developers during some downtime between project assignments and was picked up by other developers before coming to the attention of Bill Gates) and the Amazon web services community.

[51] The portion of the O'Reilly (2005a) essay that directly discusses the architecture of participation was first published online as O'Reilly (2004). Page numbers here refer to the 2005 publication, but the online 2004 version is more often cited in the open source community.

Some systems, such as Linux, the Internet, and the World Wide Web, have a *natural* architecture of participation. Other systems create an architecture of participation by *deliberate design*: "But as Amazon demonstrates, by consistent effort (as well as economic incentives such as the Associates program), it is possible to overlay such an architecture on a system that would not normally seem to possess it" (p. 253). O'Reilly describes the results of a keyword search on the term "JavaScript" as a vivid illustration of the Amazon architecture of participation (p. 253):

> *[It] is usually topped by my own* JavaScript: The Definitive Guide. *The book has 192 reviews, averaging 4 1/2 stars. Those reviews are among the more than ten million user reviews contributed by* amazon.com *customers.*
>
> *Now contrast the #2 player in online books,* barnesandnoble.com. *The top result is a book published by Barnes & Noble itself, and there is no evidence of user-supplied content.* JavaScript: The Definitive Guide *has only 18 comments, the order-of-magnitude difference in user participation closely mirroring the order-of-magnitude difference in sales.*
>
> *Amazon doesn't have a natural network-effect advantage like eBay, but they've built one by architecting their site for user participation. Everything from user reviews, alternative product recommendations, ListMania, and the Associates program, which allows users to earn commissions for recommending books, encourages users to collaborate in enhancing the site. Amazon Web Services, introduced in 2001, take the story even further, allowing users to build alternate interfaces and specialized shopping experiences (as well as other unexpected applications) using Amazon's data and commerce engine as a back end.*
>
> *Amazon's distance from competitors, and the security it enjoys as a market leader, is driven by the value added by its users. If, as Eric Raymond said in* The Cathedral & the Bazaar, *one of the secrets of open source is "treating your users as co-developers", Amazon has learned this secret. But note that it's completely independent of open source licensing practices! We start to see that what has been presented as a rigidly constrained model for open source may consist of a bundle of competencies, not all of which will always be found together.*

The underlying logic is of positive network effects so that systems get better the more people use them. O'Reilly (2005c) extends these arguments to "Web 2.0" – the business revolution in the computer industry caused by the move to the Internet as a platform:

> *What we're now calling Web 2.0 is a direct outgrowth of the core principles that made open source software successful, but in my opinion, many of the projects and companies that make up the Web 2.0 movement have gone far beyond open source in their understanding of how to build systems that leverage ... the architecture of participation.*

A close reading of O'Reilly's essays, articles and presentations (O'Reilly, 2001, 2004, 2005a, 2005b, 2005c) and blog posts at the O'Reilly Media websites published between 2006 and 2008 identified the following architecture of participation heuristics:

- Small modular applications.
- Well-defined application interfaces, minimally specified, that place few constraints on interoperability with other applications.
- Transparency of design: the internal design of the system is open to be examined.
- A small core and well-defined extension mechanism, also described as a tightly-controlled cathedral surrounded by an open bazaar.
- Rival ideas and solutions compete with one another in a free market for ideas.
- Low barriers to entry for new users.
- Contributions from outside the community are welcomed, and compete on a level playing field with contributions from within the community.
- User value, as assessed by users, is the criterion for selecting one solution rather than a different one.
- Users have the credible capability to fork the project, providing strong incentives for developers to be responsive to users.
- Participation is automatic; contribution is the default behavior of using the system, and no extra effort is required to contribute.
- Users trust the system.
- Dial-tone: users can do something themselves that previously required a professional operator, analogous to direct-dialing a telephone call rather than placing a call with the assistance of an operator[52].
- Value is extracted from what users already do without requiring behaviour change

O'Reilly's heuristics are eclectic. Some may be generally applicable to many systems (e.g., low barriers to entry, or user trust), while others may be specialized to particular contexts (e.g., open source code for open source software systems)[53].

[52] O'Reilly's notion of "dial-tone" is similar to Shirky's (2008) "mass amateurization" (capability available to amateurs that was previously exclusive to professionals) or Christensen & Raynor's (2003a, p. 49) "low-end disruption" (people able to do something themselves that previously required paying a specialist).

[53] Chapter 11 revisits and reinterprets this list and augments it with new findings from this study.

The architecture of participation is not a dichotomous trait that a system does or does not possess, but rather a scale where some systems are architected to promote participation more effectively than others. Additionally, effectiveness may depend importantly on the *fit* between system properties and context. O'Reilly (2006) proposes that both the platform architecture and the characteristics of the development community determine the *strength* of an open source architecture of participation (Table 10).

Table 10: Strength of architecture of participation (adapted from O'Reilly, 2006)

Strength	License	Architecture	Community	Example open source projects
Level 1 (low)	Open source	Monolithic	Open or closed	Sendmail (a closed community), OpenOffice.org (an open community).
Level 2 (medium)	Open source	Modular	Closed	MySQL, JBoss.
Level 3 (high)	Open source	Modular	Open; pure meritocracy	Linux, Apache, Mozilla, Perl, Python, Ruby, PHP.

The architecture of participation has been widely influential in the technology community. It is cited frequently in presentations by software developers, community project leaders, entrepreneurs and investors (e.g., Udell, 2004; Taylor, 2006; Shirky, 2008), and the logic is featured explicitly in presentations by Eclipse Foundation staff (e.g., Milinkovich, 2007, 2008a, 2008b, 2008c; Skerrett, 2008a, 2008b, 2009) and implicitly in the technical writing of Eclipse platform architects (e.g., Gamma & Wiegand, 2005; Gamma, 2008; McAffer et al. 2010a).

4.2 Towards an operationalized construct

Management researchers have made some progress towards operationalizing the architecture of participation as a useful management construct and the basis for a theory of participation. Baldwin & Clark (2006, p. 1116) argue that the architecture of a codebase is a critical factor that lies at the heart of the open source development process, and develop the theoretical argument that codebase architectures that are *more modular* or have *more option value* increase the incentives for developers to join and remain involved in an open source development effort. These arguments revise and extend earlier work on Design Rule Theory (Baldwin & Clark, 1997, 2000; Garud et al. 2003; Lenfle & Baldwin, 2007) surveyed in chapter 3.

West & O'Mahony (2008) consider technical architecture to be one subset of what they call *participation architecture*: "the socio-technical framework that extends participation opportunities to external parties and integrates their contributions" (p. 146). According to West & O'Mahony (2008), sponsors consider *three design dimensions* that together create a specific participation architecture: (1) *production* (they way that the community conducts production processes), (2) *governance* (the processes by which decisions are made within the community), and (3*) intellectual property rights* (the allocation of rights to use the community's output). Within the three design dimensions are eleven design parameters: three production parameters (live code access, public commit process, and sub-project creation), four governance parameters (non-profit foundation, membership, member fee, and community release authority), and four intellectual property parameters (code ownership by foundation, subproject ownership, software license, and license type). The three design dimensions map to the openness constructs of transparency and accessibility in different ways (Table 11).

Table 11: Community design dimensions (adapted from West & O'Mahony, 2008)

Community design dimension	Transparency	Accessibility	Proprietary model (neither transparent nor accessible)
Production	Ability to read code and observe or follow production processes.	Ability to change code directly.	Production remains within a single corporation.
Governance	Publicly visible governance, observers can understand how decisions are made.	Ability to participate in governance.	The corporation makes all decisions at its own discretion.
Intellectual property	Rights to use code and access source code.	Ability to reuse and recombine code in the creation of derivative code.	Limited use rights are granted by the corporation for a licensing fee.

The degree to which sponsors offer transparency or accessibility affects the sponsor's ability to attract external participants and grow communities. Sponsors who provisioned source code under an open license sought greater external participation and technological adoption, whether from prospective users, producers of complementary products, or even direct competitors. For many community sponsors in the West & O'Mahony (2008) sample, the authors conclude that divesting some degree of control was as much a *legitimating* strategy as it was a *recruitment* strategy.

4.3 Summary and synthesis

The architecture of participation is a widely-cited concept from the practitioner literatures in software, Internet services, and technology entrepreneurship. At its essence, the theory of action underpinning the architecture of participation is a familiar economic argument – that individuals act in ways that provide them with benefit, preferring actions (or inaction) that require little or no effort, time, or change in behaviour. The intellectual origins are classic economic arguments on the tragedies of common resources and collective action – and the rejection of those arguments for systems successfully designed for user participation. These are the same intellectual roots as the IAD framework (section 5.1), Ostrom's work on governance of common pool resources (section 5.2), and the emerging body of scholarship on governance of knowledge commons (section 5.3).

The salient insights from this chapter for this research include the following. (1) The architectures of some systems better encourage user participation than do the architectures of other systems. (2) An architecture of participation can be natural or accidental, or it can be deliberately designed. (3) Management researchers have made progress towards developing a theory of participation that considers technical architecture (Baldwin & Clark, 2006) and community design (West & O'Mahony, 2008) but extant theory is silent regarding the link between participation in the commercial ecosystem and participation in the development community – a gap addressed by this research.

This chapter, and the previous two chapters, have surveyed extant research and its potential contributions to business ecosystems. The next chapter introduces an organizing framework to structure data collection and analysis.

5 Institutional Analysis and Design (IAD)

The Institutional Analysis and Design (IAD) framework (Kiser & Ostrom, 1982; Ostrom, 1990, 2005; Ostrom & Hess, 2007) is the central framework guiding data collection and analysis in this thesis. This chapter introduces IAD and situates it within management scholarship, specifically within the "regulative pillar" of the institutional perspective of organizations (Scott, 2008); it briefly surveys how IAD has been applied previously to study effective governance of common-pool natural resources (Ostrom, 1990), open source software (Schweik, 2007) and Internet goods (Hofmokl, 2010).

5.1 Institutions, rules, and resources

The *Institutional Analysis and Design (IAD) framework* is an institutional framework for investigating the governance of collective action. It has been useful for previous case study research on polycentric governance and collective action in other domains (Ostrom, 2009; Poteete et al. 2010) but has not previously been employed to examine the institutions of business ecosystems. Ostrom (2005, p. 3) writes:

> *Institutions are the prescriptions that humans use to organize all forms of repetitive and structured interactions including those within families, neighborhoods, markets, firms, sports leagues, churches, private associations, and governments at all scales. Individuals interacting within rule-structured situations face choices regarding the actions and strategies they take, leading to consequences for themselves and for others.*

Rules or the absence of rules structure situations: "The rules affecting one situation are themselves crafted by individuals interacting in deeper-level situations" (p. 3). Rules can be tacit and taken-for-granted, and go unnoticed and unquestioned by participants (p. 5):

> *Frequently, we are not even conscious of all the rules, norms, and strategies we follow. Nor have the social sciences developed adequate theoretical tools to help us translate our implicit knowledge into a consistent explicit theory of complex human behavior.*

IAD is not a theory or a model, but rather a *framework* to organize diagnostic and prescriptive inquiry. According to Ostrom & Hess (2007, p. 43), "IAD is particularly appropriate for analyses of various types of commons and common-pool resources" and "well suited for analysis of resources where new technologies are developing at an extremely rapid pace." Ostrom (2005) and Poteete et al. (2010) distinguish between frameworks, theories and models, situating each as different elements in a nested set of theoretical concepts. "Frameworks provide a metatheoretic language that is needed to talk about theories and that can be used to compare theories. They attempt to identify *universal* elements that any relevant theory would need to include" (Ostrom, 2005, p. 28, emphasis in original; see also p. 6). *Theories* specify which components of a framework are relevant for certain kinds of questions, to diagnose a phenomenon, explain its processes, and predict outcomes. Several theories are usually compatible with any framework: "Microeconomic theory, game theory, transaction cost theory, social choice theory, public choice, constitutional and covenantal theory, and theories of public goods and common-pool resources are all compatible with the IAD framework" (Ostrom, 2005, p. 28) – as are the complex adaptive systems of the modularity and design literature (section 3.4) and the ecological analogy of business ecosystem (section 2.5). *Models* make precise assumptions about a limited set of parameters and variables for a specific situation. Multiple models are compatible with most theories of any particular situation. Ostrom's language and examples are anchored in the traditions of economics and political science rather than management studies, but the distinction remains a useful one for this research design, and parallels similar arguments in management writing. Van de Ven

(2007, p. 143), for example, argues that social scientists cannot really observe or test theories, only the models that operationalize and instrument theories (see also Morgan & Morrison, 1999; McKelvey, 2002).

The objective and scope of IAD are ambitious: "a universal framework composed of nested sets of components within components for explaining human behavior" (Ostrom, 2005, p. 7). Ostrom (2005) describes the IAD framework as a multilevel conceptual map that can be presented at many scales, starting from the biophysical structure, to individual cognitive structures, to structures composed of multiple individuals – families, firms, industries, nations, and many other units. Less formally, IAD has been described as a collection of nested building blocks that enable social scientists to understand human interactions and outcomes across diverse settings (Ostrom, 2009), and as "an analytical scaffolding that contains a universal set of intellectual building blocks" (Ostrom & Hess, 2007, p. 42) with a checklist of variables, organized into a causal schema, while allowing great flexibility (Gibson, 2005, p. 229). For any real social situation, the general IAD framework in its full richness is impractical; practical inquiry focuses pragmatically on aspects of the framework most relevant to the situation of interest. IAD can be used to analyze *static situations* crafted by existing rules and related to an unchanging environment, or *dynamic situations* where individuals develop new norms, new rules, and new physical technologies. It can be employed at multiple levels of analysis, where the whole system at one level is a part of a system at another level (Ostrom 2005), and to link together the different levels of a multilevel system.

According to Scott (2008, p. 48), "Institutions are comprised of regulative, normative and cultural-cognitive elements that, together with associated activities and resources, provide stability and meaning to social life." Scott associates these three groups of elements as the the *three pillars of institutions* – (1) *regulative*, (2) *normative*, and (3) *cultural-cognitive* – each with differing underlying assumptions, mechanisms, and indicators. The major differences between various schools of institutional scholarship center on which elements are accorded priority. The IAD framework most strongly emphasizes rules and incentives, consistent with Scott's regulative pillar, but it is sufficiently flexible to also accommodate normative and cultural-cognitive elements, and a behavioral rather than a strictly game-theoretic rational-choice model of decision-making (Ostrom, 2005).

Figure 2 is a simplified schematic of the IAD framework, adapted from Ostrom (2005, Figure 1.1, p. 13). The IAD framework posits three very broad clusters of variables, and provides three ways to enter when studying a question: the *underlying factors* on the left side, the *action arena* in the middle, and the *outcomes* on the right. In this research, the entry point will be from the right, as is typical for studies seeking to explain observed outcomes (Hess & Ostrom, 2007). In the static mode of analysis, the IAD underlying factors are assumed to be exogenous and fixed. In the dynamic mode, the underlying factors change over time, as indicated by the dotted feedback loop.

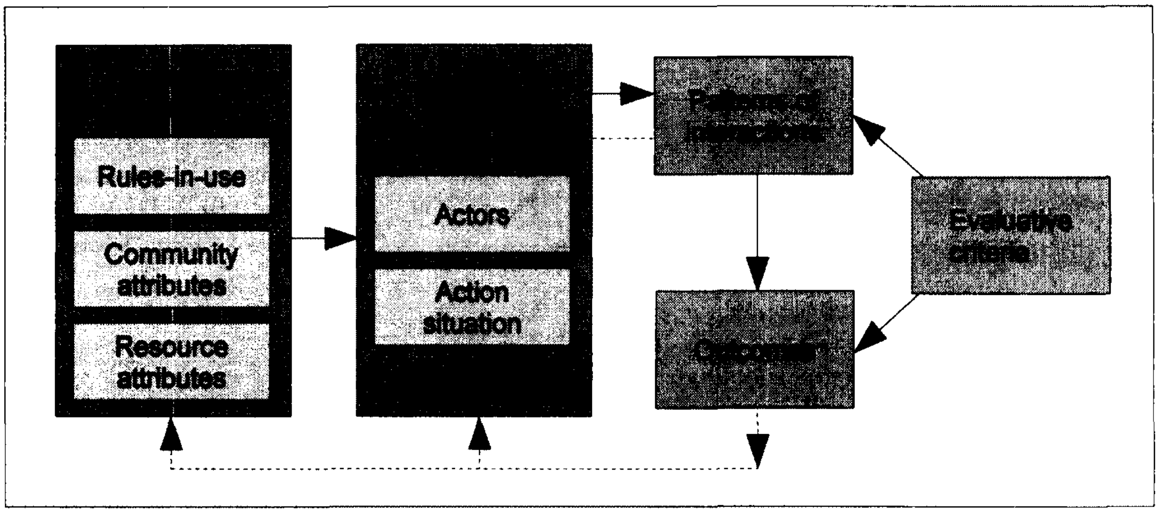


Figure 2: Schematic of the IAD framework (adapted from Ostrom, 2005, Figure 1.1)

The elements of the IAD framework each possess deeper structure which may or may not be relevant for any particular study. Ostrom (2005, p. 7) argues that although the number of variables that *could* be considered in any particular situation is very large, good scientific explanation employs "just enough variables to enable one to explain, understand, and predict outcomes in relevant settings." Management theorists also recognize parsimony as a quality of good theory (Pfeffer, 1982; Whetton, 1989; Eisenhardt, 1989). Accordingly, each IAD element is described below focusing on the aspects salient to the guiding research questions and field setting of this research.

The *outcomes* of interest for the Eclipse field setting include the health of the Eclipse ecosystem, the active participation by organizations in the ecosystem, and the active participation and productive output of developers in the community; the appropriate outcomes differ for each of the three levels to be examined – ecosystem, project, and company. *Patterns of interaction* are the link between the action situation and outcomes, as impacted by incentives, actions, and the actors. In commons research, how the actors interact strongly affects the success or failure of the resource (Ostrom, 1990). Interactions can be trusting, conflictual (especially with rapid change in community membership, values, and goals), unfocused and unthinking, or influenced by the tribalism of different groups (hierarchies, lack of respect, and distrust). Adequate information about the structure of the situation, the opportunities that participants face, and the costs of diverse action may develop increasing trust that can lead to positive outcomes (Ostrom & Hess, 2007). *Evaluative criteria* are used by researchers to assess the outcomes that are being achieved as well as the likely set of outcomes that could be achieved under alternative actions or institutional arrangements. They are applied to evaluate both the outcomes and the interactions among participants that lead to outcomes.

The *action arena* is the core unit of analysis. It is comprised of an *action situation* and the *actors* in that situation. For some research problems, it may be useful to specify a formal mathematical model using game theory or decision theory; for other problems, it may be more appropriate to describe the salient properties of the action arena using words. This research adopts the latter approach. In any case, of particular concern are the *incentives* facing different participants, and the awareness of participants of those incentives. Poteete et al. (2010) identifies seven clusters of variables that may be useful for characterizing an action situation, and four clusters of variables that may be useful for characterizing a participant. Possible action situation variables include (1) the set of participant actors (who may be single individuals or organizational actors), (2) the positions to be filled by participants, (3) potential outcomes, (4) action-outcome linkages, (5) the control that participants exercise, (6) types of information available to participants about actions and outcomes and their linkages, and (7) the costs and benefits assigned to actions and outcomes. Possible actor variables include (1) the way the participant acquires, processes, retains, and uses information and knowledge about contingencies, (2) the preferences of a participant related to actions and outcomes, (3) the conscious or unconscious processes that a participant uses for selection particular courses of action, and (4) the resources that the participant brings to the situation.

The *underlying factors* include three clusters of variables: (1) the *rules-in use* by participants to order their relationships, (2) the *resource attributes* that describe the structure of the resource system, and (3) the *community or constituent attributes* that describe the social context in which the action arena is placed. At any point in time, the action arena is viewed as a set of dependent variables determined by the (independent) exogenous variables. When long time periods are considered, the underlying factors are reshaped by long feedback loops indicated by the dashed line in Figure 2.

The concept of *rules* is central to the analysis of institutions. Rules can be thought of as the set of instructions for creating an action situation in a particular environment. Rules are the "shared understanding by participants about *enforced* prescriptions concerning what actions (or outcomes) are *required*, *prohibited*, or *permitted*" (Ostrom, 2005, p. 18, emphasis in original). "Well-understood and enforced rules operate so as to *rule out* some actions and to *rule in* others" (p. 18, emphasis in original). The label "rules-in-use" distinguishes the salient rules that are generally known and enforced from "rules-in-form" – the normative instructions that are "merely written in administrative procedures, legislation, or a contract and not known by the participants or enforced" (Ostrom & Hess, 2007, p. 50). (This research uses a third label, "rules-as-written," for all codified rules, which may or may not be implemented in practice.) *Property rights* are the important category of rules that define actions that individuals may take in relation to other individuals regarding some thing: "understanding that property rights – whether intellectual or real – are *bundles* of rights is extremely important (Ostrom & Hess, 2007, p. 53, emphasis added). Understanding the process of governance is partly about asking where the rules originate. Rules can be analyzed at three levels: *operational, policy*, and *constitutional* (Kiser & Ostrom, 1982; Ostrom, 1990). At the *operational* level, individuals are interacting with each other and the relevant physical/material world, making day-to-day decisions. The *policy* (or collective choice) level is where individuals interact to make the rules at the operational level about how a resource should be managed. The *constitutional* level includes the rules that define who must, may, or must not participate in making collective choices that in turn affect the set of operational rules. "Rules, however, rarely so constrain behavior that they are the sole structure factors affecting who participates, what their incentives are, what interactions ensue, and what outcomes are obtained" (Ostrom & Hess, 2007, p. 50). Carefully observing and documenting rules is an important area of attention at all levels of this research, especially the rules regarding platform contribution and appropriation.

The *resource attributes* affect what actions are physically possible, what outcomes can be produced, how actions are linked to outcomes, and what is contained in the actors' information sets. Under different biophysical conditions, the same set of rules may yield entirely different types of action situations. Typically, the attributes of the resource as an economic good – particularly exclusion and subtractability – are important considerations; other considerations may include the size, complexity, and predictability of the resource systems. The resource system of concern here is the Eclipse platform, and its attributes are examined in chapter 7.

The *community attributes* or *constituent attributes* that affect action arenas include "the values of behavior generally accepted in the community; the level of common understanding that potential participants share (or do not share) about the structure of particular types of action arenas; the extent of homogeneity in the preferences of those living in a community; the size and composition of the relevant community; and the extent of inequality of basic assets among those affected" (Ostrom, 2005, pp. 26-27). "The term 'culture' is frequently applied to the values shared within a community. Culture affects the mental models that participants in a situation may share. Cultures evolve over time faster than our underlying genetic endowment can evolve" (p. 27). The communities of interest here are the actors at each level of analysis – ecosystem, company, or project. To avoid possible confusion with the Eclipse community construct, this research employs the term "community attributes" only when referring specifically to the IAD attributes of the Eclipse community, and the term "constituent attributes" when referring to the IAD attributes of the Eclipse ecosystem (see subsection 6.2.2).

Previous paragraphs have described the IAD elements in their general form, suitable for a wide range of phenomena. Ostrom & Hess (2007; Hess & Ostrom, 2003, 2007) recommend several specific refinements to IAD for analysis of "knowledge commons" situations – particularly regarding the most relevant physical and community elements. These recent refinements are described in section 5.3 after the next section examines the traditional application of IAD to common-pool natural resources.

5.2 Institutions of common-pool natural resources

Ostrom (1990) identifies three classic models of collective action that together comprise the conventional theory of the commons: Mancour Olson's *Logic of Collective Action* (Olson, 1965), Gareth Hardin's *Tragedy of the Commons* (Hardin, 1968), and the *Prisoners' Dilemma* game of analytic game theory (Poundstone, 1992). All predict that collective action cannot be sustained without strong property rights or a coercive state. Hardin (1968, p. 1244) famously writes: "Ruin is the destination toward which all men rush, each pursuing his own best interest in a society that believes in the freedom of the commons. Freedom in a commons brings ruin to all." Hess & Ostrom (2007, p. 11) write: "All of these models ... can be useful in helping to conceptualize some of the incentives in simple situations... The problem with them is that they have been overused as realistic models of much more complex and dynamic situations."

Over three decades of subsequent empirical work, much of it by Elinor Ostrom and her colleagues[54], found that collective action problems can sometimes be solved by voluntary action. "Some resource users did self-organize and succeeded in preventing severe over-harvesting of resources they depended on" (Poteete et al. 2010, p. 39), and "although these institutions have not always succeeded, neither have Hardin's preferred alternatives of private or state ownership" (Dietz et al. 2003). The IAD framework was an outcomes of this research program, arising from the need of commons researchers to

[54] Elinor Ostrom received the 2009 Nobel Memorial Prize in Economic Sciences (shared with Oliver E. Williamson), for "her analysis of economic governance, especially the commons" (Ostrom, 2009).

compare diverse collective action situations – groundwater basins, irrigation systems, grazing systems, and forests, as well as urban goods such as policing and education – to investigate the determinates of successful collective action (Poteete et al. 2010). The earliest applications were to guide case study research and to enable cross-case comparisons (Ostrom, 1990). Later applications employed IAD for meta-analysis, experimental designs in the laboratory and in the field, mixed method studies, agent-based simulation models, and large sample studies (Ostrom, 2005; Poteete et al. 2010).

An early result of the natural resources studies was the Ostrom & Ostrom (1977) typology recognizing four basic types of goods, classified according to two dimensions: (1) difficulty of *exclusion* of potential beneficiaries (low/high) and (2) *subtractability* of use (low/high)[55]. The introduction of *subtractability* (sometimes referred to as *rivalry*) was an important extension and refinement of the Samuelson (1954) public/private distinction that considered only exclusion[56]. *Public goods* (low subtractability, difficult to exclude) include useful knowledge and sunsets. *Private goods* (high subtractability, easy to exclude) include personal computers and doughnuts. *Toll goods* (sometimes called *club goods*; low subtractability, easy to exclude) include magazine subscriptions and golf memberships. *Common-pool resources* (high subtractability, difficult to exclude) include irrigation systems and libraries. Figure 3 illustrates the four basic types of economic goods, adapted from Ostrom (2005, Figure 1.3, p. 24) and Ostrom & Ostrom

[55] The first published description of IAD was Kiser & Ostrom (1982); thus the Ostrom & Ostrom (1977) typology actually precedes IAD. Ostrom (2005) recounts the intellectual history of these ideas.

[56] Samuelson (1954) had previously argued that all goods usable by humans are either *pure private* or *pure public* goods according to exclusion. Early treatments following Samuelson had considered only the possibility or impossibility of exclusion; later treatments considered *cost of exclusion*, so that goods with prohibitively high costs of inclusion were *de facto* public goods. The additional consideration of subtractability by Ostrom & Ostrom (1977) was a useful extension.

(1977, p. 12). The type of good being examined is a typical starting point for identifying the biophysical conditions impacting an action area. The natural resources commons examined by Ostrom were typically common-pool resources, but the platform resource of a business ecosystem could be complex, comprising multiple elements of the four types or any combination of types. Although information is normally a public good, property rights and access technology could transform it into a toll good or even a private good[57], and services with finite capacity could behave as common pool resources[58].

		Subtractability of use	
		Low	High
Difficulty of excluding potential beneficiaries	Low	[illegible]	[illegible]
	High	[illegible]	[illegible]

Figure 3: Types of goods (adapted from Ostrom, 2005; Ostrom & Ostrom, 1977)

Ostrom (1990) performed cross-case analysis of hundreds of IAD empirical studies seeking general design principles of robust and long-enduring institutions – chapter 11 returns to those results for comparison with the findings of this thesis.

[57] See, for example, Katz & Shapiro (1985) and Shapiro & Varian (1998, 1999) on the economics of information. Commons researchers refer to the threat of transforming public goods into private goods via property rights and technology as the *dilemma of enclosure* (Kranich, 2007), a historical allusion to the gradual physical enclosure of shared public land in Europe by landowners and the state with fences and walls and private property rights over a period of approximately 500 years (Hess & Ostrom, 2007, pp. 12-14). A second commons dilemma is the *dilemma of preservation* (Waters, 2007). More generally, "protecting the commons" and "building new knowledge commons" are two prevalent themes in the knowledge commons literature (Hess & Ostrom, 2007).

[58] One example of a resource with finite capacity would be a download server for software distribution. Finite bandwidth and processing capability place an upper limit on the download capacity shared by users.

5.3 Institutions of the new knowledge commons

IAD scholars recommend several refinements to IAD for "knowledge commons" resources (Hess & Ostrom, 2007; Ostrom & Hess, 2007). This section draws on that work to sharpen the general IAD constructs to increase their relevance for this research and to remove from consideration elements that are not helpful for the present research.

Hess & Ostrom (2007) take stock of the body of work, "still in its early infancy" (p. 4), looking at "knowledge as a shared resource, a complex ecosystem that is a commons – a resource shared by a group of people that is subject to social dilemmas" (p. 3). A key question in commons research is whether or not the lessons from natural-resources can transfer to the knowledge commons. Their cautious and tentative answer is affirmative, but with "a continual challenge to identify the similarities between knowledge commons and traditional commons, such as forests and fisheries, all the while exploring the ways knowledge as a resource is fundamentally different from natural-resource systems" (p. 5). A key difference is subtractability: common-pool resources are highly subtractable, while most knowledge is non-subtractable. "In fact, the more people who share useful knowledge, the greater the common good. Consideration of knowledge as a commons, therefore, suggests that the unifying thread in all commons resources is that they are jointly used, managed by groups of varying sizes and interests" (p. 5). *Open access* to information differs from open access to land or water. Open access in the traditional commons can lead to over-consumption and depletion. Open access to a non-subtractable knowledge commons means free and unfettered access but authors may still retain ownership. Bollier (2007, p. 28-29) summarizes the similarities and differences of the traditional commons and knowledge commons:

> *The commons has too many variations to be captured in a fixed, universal set of principles. Each commons has distinctive dynamics based on its participants, history, cultural values, the nature of the resource, and so forth. Still there are some recurring themes evident in different commons... The commons fills a theoretical void by explaining how significant value can be created and sustained outside of the market system. The commons paradigm does not look primarily to a system of property, contracts, and markets, but to social norms and rules, and to legal mechanisms that enable people to share ownership and control of resources.*

Commons researchers see their work as a departure from the "standard economic narrative about how things work" (Bollier, 2007, p. 27).

Hess & Ostrom (2003) introduce the distinction between *facilities* and *artifacts* to describe the resource characteristics of a knowledge commons[59]. *Facilities* store artifacts and make them available. *Artifacts* are discreet, observable, namable representations of ideas, such as articles, research notes, books, databases, maps, computer files, and web pages. Traditional knowledge artifacts are typically rivalrous; digital artifacts are typically non-rivalrous, used concurrently by multiple users. Artifacts are the physical resource or flow units of a facility, where *flow unit* is an economic term referring to a resource that must be passed from one individual to another to have any public value.

Ostrom & Hess (2007) propose useful refinements to several of the IAD elements for studies of knowledge commons. Levels of action and decisions are likely to be *polycentric* with "decentralized, alternative areas of authority and rule and decision making" (p. 55). For all knowledge commons, the exogenous physical characteristics are those of *distributed digital information* (p. 45), which differ from the characteristics of

[59] This is an alternative to the two-fold distinction between the *resource system* (e.g., a fishery) and the *resource unit* (the fish) that is useful for investigating natural resource commons (Ostrom, 1990).

physical libraries and archives that preceded the digital era; the Hess & Ostrom (2003) distinction of facilities and artifacts may be useful for specifying these properties. Within the community element, it may be useful to distinguish between multiple roles, e.g., *resource providers*, *resource users*, and *policy-makers*, which are usually nested; a useful question is whether the values of the community are shared or divided. Rules-in-use for the knowledge commons include intellectual property (IP) rights, best understood as *bundles* of various rights to access, contribution, extraction, removal, extension, exclusion of others' rights, and alienation (i.e., the right to sell other rights) (Ostrom & Hess, 2007, pp. 52-53; Schlager & Ostrom, 1992). In any particular bundle of property rights, none, some, or all of these rights may be present. One set of property rights salient to the Eclipse field setting is the *Eclipse Public License (EPL)* governing the distribution of Eclipse software assets (examined in chapter 7 and subsection 7.8.4.

Regarding outcomes in the knowledge commons, Ostrom & Hess (2007, p. 60) write: "Most of the outcomes that have been written about in the newly emerging knowledge-commons literature are either types of enclosures of information that used to be open or the creation of new digital commons that provide better access to information." The present research is a variant on the second theme. Of sixteen cited examples of knowledge commons publications, outcomes examined include cooperation, institution building, collective action, self-organization, congestion of the Internet, free riding, conflict, deception, withdrawal, enclosure, and inequity. "There are a myriad number of competing outcomes – some of which are considered negative, while others are seen as positive" (p. 60). The authors identify six evaluation criteria as most frequently used: (1) increasing scientific knowledge, (2) sustainability and preservation, (3) participation standards, (4) economic efficiency, (5) equity through fiscal equivalence, and (6) redistributional equity. The evaluation criteria of interest for the field setting of this research are the criteria in use by Eclipse practitioners: an empirical question answered in chapter 8 (subsection 8.7) for the Eclipse developer community and chapter 10 (subsection 10.3) for the Eclipse ecosystem, emerging from close observation of practitioner discourse within the Eclipse field setting.

Schweik (2007) previously employed the IAD framework to guide description of "the free and open source software commons." Schweik (2007, p. 279-280) concludes:

> *The primary innovation in [open source software] projects is a combination of a new approach to software licensing coupled with Internet-based collaborative tools. This resulted in a new form of 'commons' ... but one that differs slightly from the environment commons that most readers are familiar with... In [an open source] commons, groups of people act collectively to produce a public good (the software), rather than overappropriate the resource (e.g., Hardin, 1968). In other words, the challenge in [open source] commons is how to achieve collective action to create and maintain a commons or public good rather than the issue of protecting an existing commons from destruction (a public bad).*
>
> *Importantly, [open source] projects produce this public good through a common-property regime (Benkler, 2002; Boyle, 2003) – one form of commons. In environmental-commons literature, the phrase 'common property' is defined as a resource ... where members of a defined group possess a set of legal rights, including the ability to exclude nonmembers from using that resource.... Some readers unfamiliar with [open source] software may be surprised to hear that there are property rights (copyright) and ownership issues involved... But as a result of [open source] licensing ... some individuals involved in the project do indeed have legal rights to the code (the resource), have control over what goes into future versions of the software, and can exclude others from submitting new code to a new release.*

Schweik employs IAD systematically, discussing the action arena, rules-in-use, community attributes, and physical attributes of open source software projects (Table 12).

Table 12: Open source software as a commons (adapted from Schweik, 2007)

IAD Element	Description
Underlying factors	
Rules-in-use	Copyleft, licensing, project governance
Community attributes	Either a group of passionate (volunteer) developers and users, or programmers who are paid by some organization to participate, or some combination of both.
Physical attributes	1) Utility of the software. 2) Design or structure of the software. 3) Collaborative infrastructure that helps to coordinate and manage production.
Action arena	
Actors	Computer programers or users who contribute, either voluntarily or because they are paid to do so, toward the further production of open source software.
Action situation	Whether, at some point in time, it is worth their while to continue to contribute to the development of this software.
Outcomes	Key to success of open source common-property regimes is the willingness of a programmer to contribute to the collaborative effort (action arena), and the cumulative effort of a least small teams of actors to collectively produce and maintain software (outcome).

Schweik argues that the collaboration principles around open source software projects could be applied to any collaboration built around intellectual property to increase the speed at which innovations and new discoveries are made: “In other words, we can conceive of a future where such 'knowledge commons' are built not around software, but more generally any kind of work or 'content'” (p. 277).

Hofmokl (2010) categorizes more than twenty “Internet goods” according to the classic biophysical attributes of exclusion and subtractability and concludes that few Internet goods are common-pool resources; there is, however, great diversity with respect to private goods, public goods, and toll goods. Hofmokl (2010) is a concrete example of scholarly knowledge commons research employing a small subset of the IAD framework.

5.4 Summary and synthesis

Table 13 summarizes the salient elements of the IAD framework and provides a template for characterizing the institutions examined in this field study.

Table 13: Framework for data collection (adapted from IAD)

IAD Element	Description
Underlying factors	
Rules-in-use	"Shared understanding about enforced prescriptions concerning what actions or outcomes are required, prohibited, or permitted" (Ostrom, 2005, p. 18). Rules operate at multiple levels: e.g., operational, policy, constitutional. May be tacit or explicit; may go unnoticed and unquestioned by participants. Property rights (as "bundles" of rights).
Community attributes	Properties (culture) of the community in which the action arena is placed: e.g., size and composition, extent of homogeneity with regards to values, expectations, preferences, shared understanding, and access to resources.
Resource attributes	Properties of the economic good (subtractability, excludability). Facilities and artifacts. Utility and design. Size, complexity, and predictability.
Action arena	
Actors	Description of the actors, incentives, and motivations.
Action situation	Participation decisions: whether to participate, in what form, and to what extent.
Interaction patterns	The link between the action situation and outcomes: e.g., trusting, conflictual, unfocused and unthinking, or influenced by tribalism of different groups, etc.
Outcomes	Outcomes at each level of analysis.
Evaluative criteria	Criteria to assess the outcomes achieved, and to assess the potential outcomes under alternative actions and institutional arrangements.

Ostrom & Hess (2007, pp. 43-44) summarize what is known about institutions that effectively govern collective action:

> *Effective design [of institutions] requires successful collective action and self-governing behaviors; trust and reciprocity; and the continual design and/or evolution of appropriate rules. We have learned that successful commons governance requires an active community and evolving rules that are well understood and enforced. When a resource is large and complex, users may lack a common understanding of resource dynamics, and they frequently have substantially diverse interests; thus, the costs of sustaining large and diverse resources are much higher than when governing small and relatively homogenous resources.*

Chapter 11 returns to the commons literature to re-interpret the institutional arrangements of Eclipse with regards to the extant body of knowledge about robust and long-enduring institutions to identify similarities and points of difference, and to explore any anomalies.

This chapter completed the four-chapter literature review that began in chapter 2. The next chapter details the research design and method.

6 Research design

This chapter presents the research design and detailed method to answer the guiding research questions posed in chapter 1. It is structured in four sections, beginning with an overview of the major research design decisions and the rationale for each, followed by a detailed explanation of the method employed at each nested level of the research design, an explanation of how research results are presented in subsequent chapters, and a discussion of validity and reliability.

6.1 Overview of research design

The research design is a *nested case study* (Yin, 2003a) of company participation in the Eclipse Foundation, community, and business ecosystem. For reasons explained in chapter 1, Eclipse is an *exceptional* (Siggelkow, 2007) and *exemplary* case (Yin, 2003b, p. 13): the Eclipse ecosystem is *unusual*, it is *interesting* to researchers and practitioners, and it is *understudied* by management scholars (section 1.2). The Eclipse field setting includes a membership-based not-for-profit foundation, a platform providing open source software frameworks and tools, an open source community that maintains and extends the platform software, and a commercial ecosystem creating complementary products, services and technologies built on the Eclipse platform.

Data sources for the research include direct observation of participants and participant communications, extensive archival data, and interviews with participants.

Theory-building is *inductive* (Poole et al. 2000; Locke, 2007; Eisenhardt & Graebner, 2007) using methods adapted from Eisenhardt (1989), Yin (2003a) and Van de

Ven (2007) to develop explanations accounting for observations and patterns in the data. There are some similarities to the *grounded theory* research designs of Glaser & Strauss (1967), Strauss & Corbin (1990), and Glaser (1998), but also important differences[60], such as the absence of formal hierarchical coding (Strauss & Corbin, 1990), a multilevel case structure and extensive development of *a priori* constructs from the literature (Eisenhardt, 1989), and a unit of analysis quite different from the "process by which actors construct meaning out of intersubjective experience" (Suddaby, 2006, p. 634).

Theory-building is *multilevel* (Hitt et al. 2007), examining governance and participation within the Eclipse field setting at three nested levels: *projects* (selected from the population of Eclipse open source software projects), *polycentric governance structures* (including the Eclipse Foundation board of directors, Eclipse Foundation staff, and cross-project governing councils), and *companies* (selected from the population of companies participating in the development of selected projects). Collectively, these three nested levels examine individual and organizational participation in the major components of the Eclipse field setting. Figure 4 illustrates the nesting of cases at each level of analysis within this research design – a top-level case of the total system (the Eclipse field setting including the Eclipse Foundation, platform, community, and ecosystem), a nested multiple case study of Eclipse projects, an embedded case study of polycentric governance structures, and a nested multiple case study of participating companies – adapted from the visual notations of Yin (2003a, figure 2.5, p. 5).

[60] Eisenhardt & Graebner (2007) recommend that researchers avoid the term "grounded theory" unless employing the specific methods of Glaser & Strauss, and instead explain precisely how theory is inducted from the data: "The key here is to convey the rigor, creativity, and open-mindedness of the research processes while sidestepping confusion and philosophical pitfalls" (p. 30). That approach is adopted here.

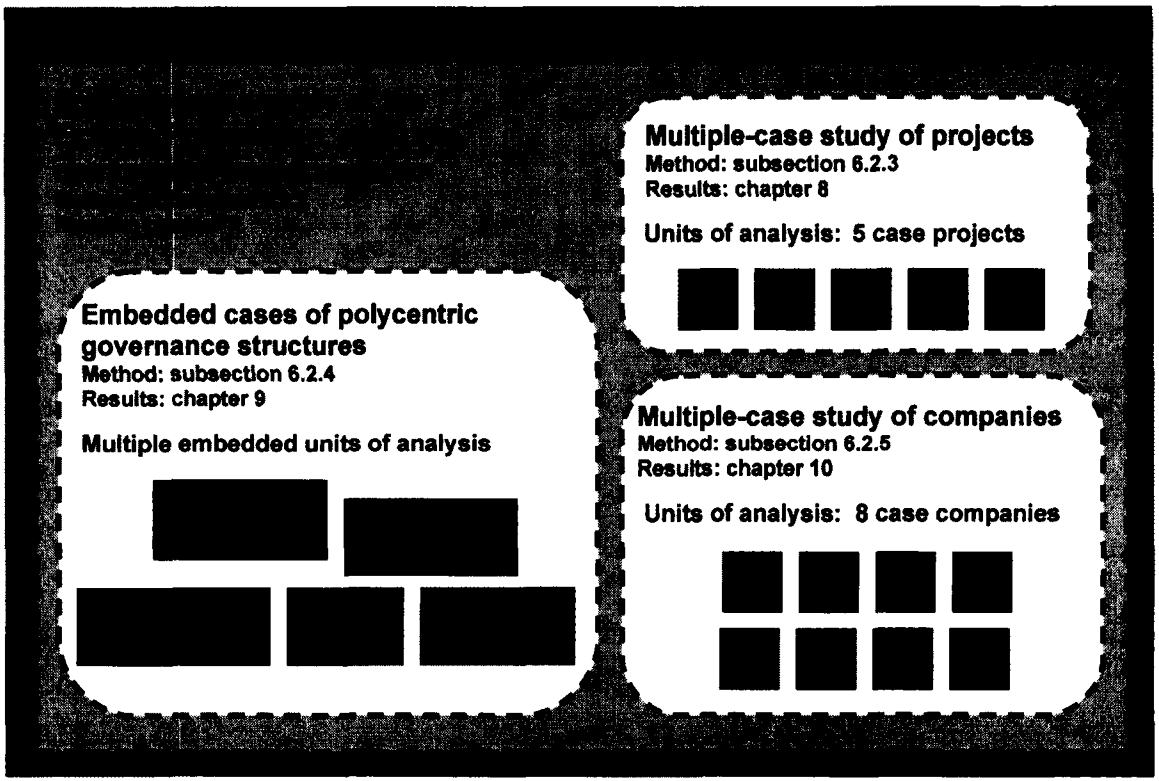

Multiple-case study of projects
Method: subsection 6.2.3
Results: chapter 8
Units of analysis: 5 case projects
Embedded cases of polycentric governance structures
Method: subsection 6.2.4
Results: chapter 9
Multiple embedded units of analysis
Multiple-case study of companies
Method: subsection 6.2.5
Results: chapter 10
Units of analysis: 8 case companies

Figure 4: Nested research design

The sampling logic is *theoretical sampling* (Glaser & Strauss, 1967), as recommended by Eisenhardt (1989; Eisenhardt & Graebner, 2007) for building management theory from case study data. With theoretical sampling, "the process of data collection is *controlled* by the emerging theory" (Glaser & Strauss, 1967, p. 45, emphasis in original). It is neither desirable nor even possible to fully specify the sample in advance; instead, the analyst jointly collects and analyzes data, and decides what data to collect next and where to find them. Cases at all levels are selected for theoretical, not statistical, reasons, based on potential to replicate or extend emergent theory.

Finally, theory-building is deliberately *multidisciplinary*. The Institutional Analysis and Design (IAD) framework (chapter 5; Ostrom, 1990, 2005; Ostrom & Hess,

2007) provides a "a metatheoretic language … [of] *universal* elements that any relevant theory would need to include" (Ostrom, 2005, p. 28) that links the project, company, and governance levels without imposing or assuming any particular theory of action. IAD contributes a flexible framework to guide data collection and accommodate constructs and theoretical insights from extant research on business ecosystems (chapter 2), relevant findings from six streams of empirically-grounded management research (chapter 3), the practitioner literature on systems with an architecture of participation (chapter 4), and past scholarship on the governance of institutions for collective action (chapter 5).

The following two subsections elaborate on the rationale for a nested case study research design, and the rationale for selecting the IAD framework, respectively.

6.1.1 Rationale for a nested case study design

The selection of a nested case study design is motivated by its potential to (1) provide better understanding of a phenomenon about which little is known, (2) produce managerially relevant knowledge, (3) discover connections between phenomena at multiple levels of analysis, (4) sharpen existing theory and discover anomalies that can extend theory in new directions, and (5) suggest new lines of inquiry for future research. Each motivation is discussed below.

First, Yin (2003a) and Eisenhardt (1989) recommend case study research as an appropriate research design to better understand a phenomenon about which little is known. As argued in chapter 1, business ecosystems are a nascent area of scholarly inquiry with large gaps and omissions in the body of knowledge, particularly concerning vendor neutral business ecosystems and community-developed platforms. Eisenhardt

(1989) argues that theory building from case study data is likely to generate novel theory that is empirically testable with measurable constructs and falsifiable hypotheses. In domains with little extant theory, where key variables and their relationships are under-explored, production of valid and testable theory can be a high-impact contribution.

Second, case studies that are conducted in close interaction with practitioners in real management situations are ideally suited to creating managerially relevant knowledge (Gibbert et al. 2008; Amabile et al. 2001; Leonard-Barton, 1990). The research questions and the field setting of this thesis are relevant to top management teams of technology-intensive companies, entrepreneurs, open source software developers, and public policy-makers seeking to promote innovation and social welfare. Managers can benefit from insights about better leveraging ecosystems and communities for competitive advantage, and developers can benefit from better understanding the consequences of increased company involvement in open source projects (Fitzgerald, 2006). As argued by Moore (2006), policy-makers who better understand the operation of business ecosystems can be more effective at promoting innovation and social welfare.

Third, multilevel case studies are well-suited to discover connections between phenomena at multiple levels of analysis (Hitt et al. 2007). Business ecosystems, as conceived by practitioners (chapter 2), are complex multilevel phenomena with emergent properties (Simon, 1962). Hitt et al. (2007, p. 1393) write: "Intensive case studies … may be especially useful for understanding multilevel phenomena."

Fourth, because theory advances partly through the discovery of anomalies for which prior categorization schemes and attendant theory cannot account, Christensen

(2006; see also Carlile & Christensen, 2005; Christensen & Carlile, 2009) argues that theory building scholars should actively seek out anomalies with detailed fieldwork conducted close to the phenomena. The nested case design of this research deliberately adopts all three mechanisms recommended by Christensen (2006) for high-impact research: (1) it employs *lenses of other disciplines*, particularly the IAD framework and scholarly research on commons governance and collective action, surveyed in chapter 5 but not previously employed to study business ecosystems; (2) it employs a *nested research design* to examine the phenomenon within the phenomenon, as explained in this subsection; and (3) it examines a *broader variety of phenomena* than previous research, which has thus far examined communities and ecosystems in separate studies (unable to attend closely to interactions), and emphasized ecosystems anchored around for-profit keystone companies with privileged control over the platform development (chapter 1).

Fifth, an anomaly-seeking, theory-building, nested case study design is likely to suggest new lines of inquiry for future research (Eisenhardt, 1989; Christensen, 2006; Hitt et al. 2007). As explained in chapter 1, this research is intended to provide the foundation for a multidisciplinary mixed-method research program in technology innovation management. Follow-on studies can further explore specific insights discovered here, extend the case data set both within the Eclipse ecosystem and to other field settings, introduce new methods of analysis enabled by larger data sets, test and sharpen emergent theory using deductive methods, and address a broader range of related research questions of high impact to theory and practice (chapter 12 elaborates further on opportunities for future follow-on research).

6.1.2 Rationale for the IAD framework

Four factors motivate the selection of IAD as the central organizing framework for data collection: (1) fit with the research problem, (2) fit with the nested case study design, (3) fit with the other conceptual frameworks of this research, and (4) potential for cross-case comparison with the commons literature. Each motivation is discussed below.

First, the research problem of understanding participation in vendor neutral business ecosystems and community-developed platforms has many similarities to the problem of understanding effective commons governance where IAD has been most useful (Hess & Ostrom, 2003, 2007; Hofmokl, 2010). Like natural resources commons and knowledge commons, vendor neutral business ecosystems are polycentric, subject to social dilemmas, and anomalous with respect to neoclassical economic theory.

Second, the IAD framework offers a potentially high-impact complement to the nested case study research design. Ostrom (1990) and Poteete et al. (2010) provide examples of high-impact case research based on IAD. IAD is inherently multilevel, providing a means to effectively theorize about phenomena at different levels of analysis.

Third, IAD fits well with the other conceptual frameworks of this research. It provides a unifying framework to link together design rules (Baldwin & Clark, 2000, 2006) as IAD rules-in-use and resource attributes, community design dimensions (West & O'Mahony, 2008) as rules-in-use and community attributes, ecosystem health and open source project success metrics as IAD outcomes, and insights from the management and practitioner literatures as underlying factors and relationships between IAD elements.

Fourth, Poteete et al. (2010, ch. 2) report how adoption of the IAD framework for commons research facilitated comparison across cases and synthetic learning across research programs and research groups. In the 1980s, the U.S. National Research Council (NRC) panel on Common Property Resource Management encouraged use of the IAD framework for analysis of collective action related to natural resources "to help authors of case studies from diverse disciplines and resource interests organize presentation of their own cases in a manner that would increase the likelihood that an initial synthesis could be derived from the empirical studies" (p. 41). The nascent business ecosystems literature of today (surveyed in chapters 2, 3 and 4) has similarities with the nascent commons literature of the 1980s, including diverse disciplinary perspectives and limited comparability of studies. Similarly, IAD may facilitate cross-case comparison and synthesis between the business ecosystem and commons literatures.

This section has provided an overview of the research design and the rationale for each of the major design decisions. The next section is a detailed treatment of the research method and the pragmatics of data collection and analysis.

6.2 Method

This explanation of the research method is structured as eight subsections. Following a brief overview of data collection and analysis activities, the first subsection is a summary of the case study questions at each level of analysis, and an explanation of nested dependencies. The next four subsections are detailed treatments of the four levels of case analysis, each attending exclusively to one level: the top-level case of the Eclipse field setting (subsection 6.2.2), the nested multiple case study of Eclipse projects (subsection 6.2.3), the embedded case study of polycentric governance structures (subsection 6.2.4), and the nested multiple case study of participating companies (subsection 6.2.5). The description of method at each level of analysis is similarly organized, beginning with explanation of the case study questions, followed by the unit of analysis, method of case selection, data sources, analysis, and structure of the case notes document. The final three subsections address topics that span levels of analysis: a consolidated treatment of data sources (subsection 6.2.6), interview field procedures (subsection 6.2.7), and the case study database and coding procedures (subsection 6.2.8).

Multilevel case methods are inherently complex, with interdependencies between nested levels of analysis that are challenging to specify and succinctly explain (Yin, 2003a, 2003b; Carlile & Christensen, 2005; Hitt et al. 2007). Furthermore, this research introduces additional methodological complexity by attending to multiple interdependent but non-hierarchical constructs within the top-level case[61]. Some readers may find it helpful to read this chapter twice – a first reading to identify and situate major concepts, and a second reading to appreciate methodological details.

[61] An adequate treatment of this argument requires case evidence of the Eclipse field setting and theory developed in chapter 7. Briefly, the various institutional constructs of importance to practitioners within the Eclipse field setting operate at different levels of analysis, but are not related by decomposition in the sense of Simon (1969; see also section 3.4). For example, an individual may be a committer on an Eclipse project and thus a participant in the Eclipse open source developer community. The same individual may also be an employee of an Eclipse member company that participates in the Eclipse ecosystem. Although the developer community and business ecosystem share some components, neither *contains* the other in a "parts-within-parts structure" (Simon, 1996, p. 186). This is an important difference between this research and the multilevel research exemplars cited by Yin (2003a) and Hitt et al. (2007). For example, the Lipset et al. (1956) study of *Union Democracy* examined individual union members, union shops, local unions, and a national union (see also Yin, 2003a, box 10) – all in a decomposable social hierarchy (Astley & Van de Ven, 1983; Sanchez & Mahoney, 1996) much like a modular product architecture (Ulrich, 1995), or the nearly-decomposable ecological hierarchy of Table 2 (Simon, 1973; Odum, 1997; Odum & Barrett, 2005). The need for non-decomposable constructs in this research design is driven by the complexity of the Eclipse field setting, with its multiple overlapping institutions, and differing treatment of individual and organizational participants (chapter 7).

Figure 5 situates the four streams of concurrent activity within the broader research design, adapting the visual notation of Yin (2003a, figure 2.4, p. 40). Implementation is iterative rather than strictly linear, with dotted lines indicating opportunities to cycle back to earlier steps to take advantage of opportunities in the data.

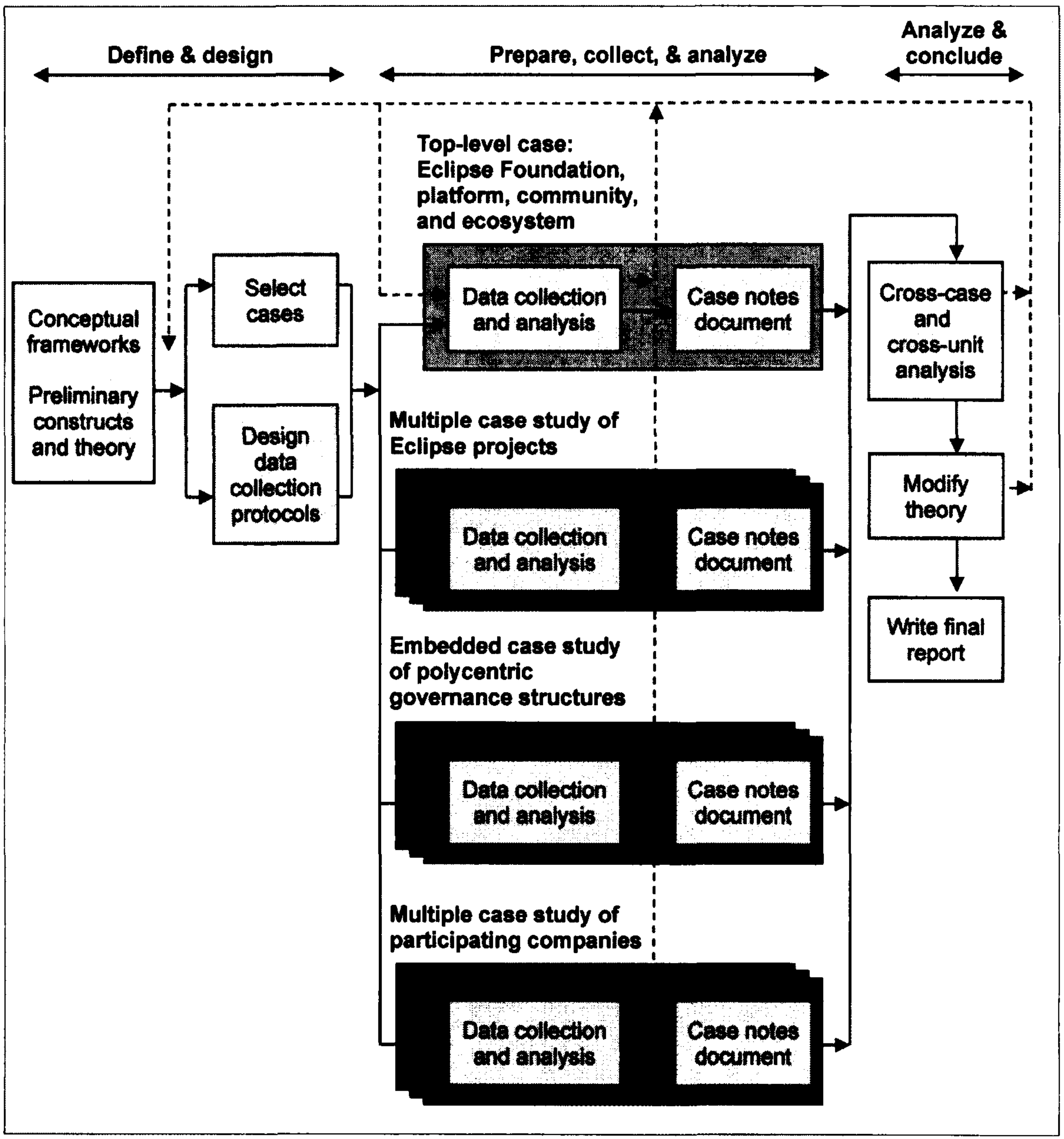
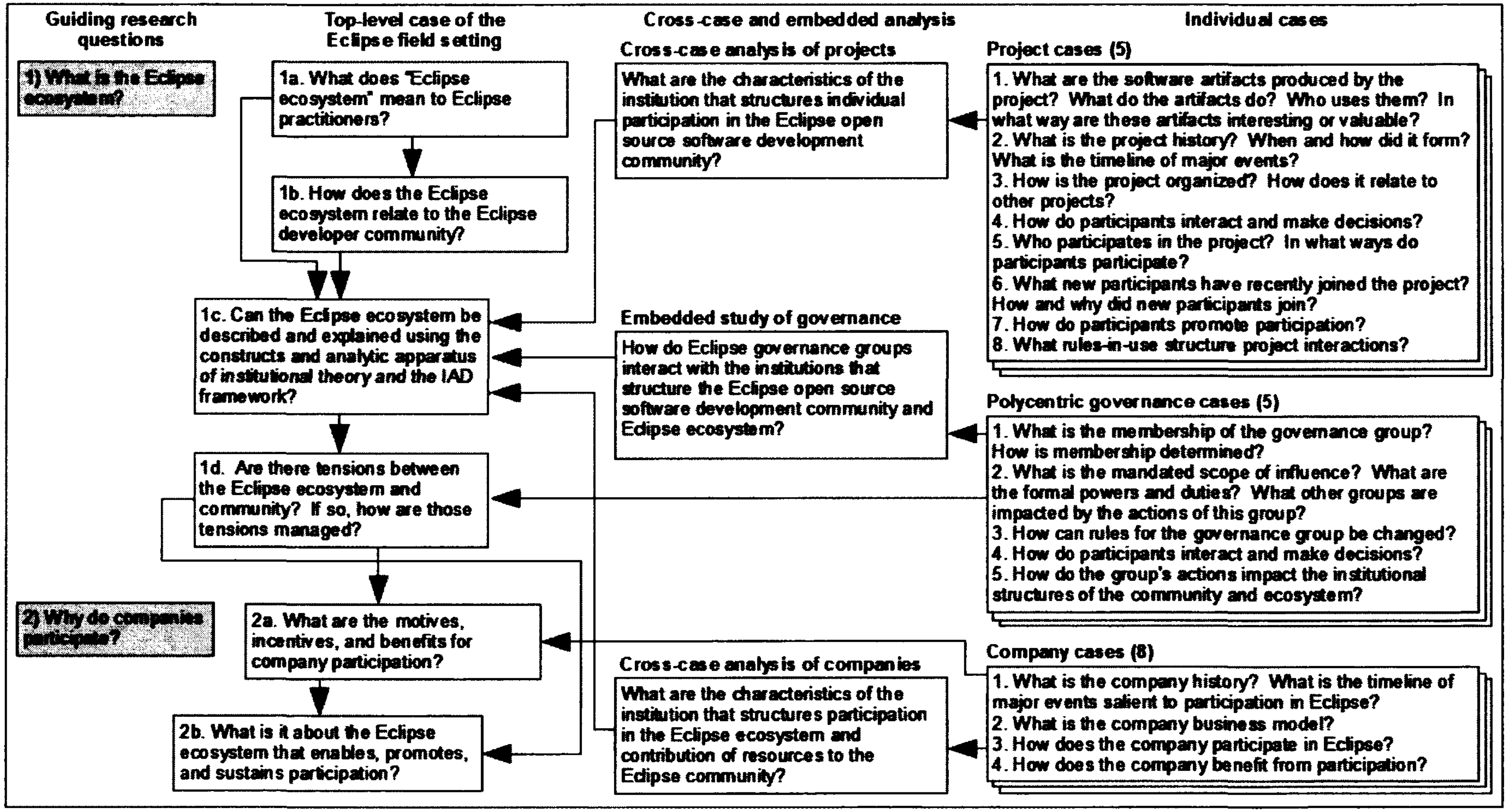

Define & design
Prepare, collect, & analyze
Analyze & conclude
Conceptual frameworks
Preliminary constructs and theory
Select cases
Design data collection protocols
Top-level case: Eclipse Foundation, platform, community, and ecosystem
Data collection and analysis
Case notes document
Multiple case study of Eclipse projects
Data collection and analysis
Case notes document
Embedded case study of polycentric governance structures
Data collection and analysis
Case notes document
Multiple case study of participating companies
Data collection and analysis
Case notes document
Cross-case and cross-unit analysis
Modify theory
Write final report

Figure 5: Method overview (adapted from Yin, 2003a, p. 40; Eisenhardt, 1989)

The discussion thus far has emphasized the differences at each level of case analysis, but the four levels are also tightly coupled in several ways. First, selection of cases at each level of analysis impacts the other levels; the project study determines the selection of companies, and both together impact selection of appropriate respondents for the governance study. Second, some data sources inform multiple levels of analysis; for example, press releases of the Eclipse Foundation can be coded for insights into the relationships between components of the top-level case, individual project cases, individual company cases, and embedded governance cases. Third, there is significant overlap of interview respondents; for example, a single well-informed respondent with a long history of contribution to Eclipse could fill multiple roles: a committer on multiple projects, an employee or owner of an ecosystem company, and a member of a council or the board of directors. Fourth, multilevel conceptual arguments including IAD (Poteete et al. 2010) and architectural modularity (Baldwin & Clark, 2000) connect the levels at a conceptual level. Tight coupling permitted data collection and analysis at all four levels to proceed concurrently.

6.2.1 Relationships between case study questions

The guiding research questions identified in chapter 1 are answered iteratively through a series of lower-level *case study questions* at each level of analysis (Yin, 2003a, p. 74). Figure 6 illustrates the relationships between the guiding research questions, the case study questions at the top-level case, and the case study questions of the nested cases. The following subsections provide detailed treatment of the case questions, grouped by level of analysis.



Guiding research questions
Top-level case of the Eclipse field setting
Cross-case and embedded analysis
Individual cases
1) What is the Eclipse ecosystem?
2) Why do companies participate?

1a. What does "Eclipse ecosystem" mean to Eclipse practitioners?

1b. How does the Eclipse ecosystem relate to the Eclipse developer community?

1c. Can the Eclipse ecosystem be described and explained using the constructs and analytic apparatus of institutional theory and the IAD framework?

1d. Are there tensions between the Eclipse ecosystem and community? If so, how are those tensions managed?

2a. What are the motives, incentives, and benefits for company participation?

2b. What is it about the Eclipse ecosystem that enables, promotes, and sustains participation?

Cross-case analysis of projects

What are the characteristics of the institution that structures individual participation in the Eclipse open source software development community?

Embedded study of governance

How do Eclipse governance groups interact with the institutions that structure the Eclipse open source software development community and Eclipse ecosystem?

Cross-case analysis of companies

What are the characteristics of the institution that structures participation in the Eclipse ecosystem and contribution of resources to the Eclipse community?

Project cases (5)

1. What are the software artifacts produced by the project? What do the artifacts do? Who uses them? In what way are these artifacts interesting or valuable?
2. What is the project history? When and how did it form? What is the timeline of major events?
3. How is the project organized? How does it relate to other projects?
4. How do participants interact and make decisions?
5. Who participates in the project? In what ways do participants participate?
6. What new participants have recently joined the project? How and why did new participants join?
7. How do participants promote participation?
8. What rules-in-use structure project interactions?

Polycentric governance cases (5)

1. What is the membership of the governance group? How is membership determined?
2. What is the mandated scope of influence? What are the formal powers and duties? What other groups are impacted by the actions of this group?
3. How can rules for the governance group be changed?
4. How do participants interact and make decisions?
5. How do the group's actions impact the institutional structures of the community and ecosystem?

Company cases (8)

1. What is the company history? What is the timeline of major events salient to participation in Eclipse?
2. What is the company business model?
3. How does the company participate in Eclipse?
4. How does the company benefit from participation?

Figure 6: Relationships between case study questions

6.2.2 Top-level case

At the top-level case, the case study questions are as follows:

1. **What is the Eclipse ecosystem?**
   1a. What does "Eclipse ecosystem" mean to Eclipse practitioners?
   1b. How does the Eclipse ecosystem relate to the Eclipse developer community?
   1c. Can the Eclipse ecosystem be described and explained using the constructs and analytic apparatus of institutional theory and the IAD framework?
   1d. Are there tensions between the Eclipse ecosystem and community? If so, how are those tensions managed?

2. **Why do companies participate?**
   2a. What are the motives, incentives, and benefits for company participation?
   2b. What is it about the Eclipse ecosystem that enables, promotes, and sustains participation?

Each top-level case study question operationalizes some specific and addressable aspect of the guiding research questions posed in chapter 1.

The steps taken to answer each question are detailed below. Three of the six top-level case study questions (1a, 1b, and 1d) are answered directly by coding specific data sources using qualitative analysis techniques recommended by Yin (2003a, 2003b) and Miles & Huberman (1994), triangulating codes from multiple sources, and organizing coded data into emergent categories. Answers to the other three top-level case study questions (1c, 2a, and 2b) depend on answers to questions at other levels of analysis.

The practitioner meaning of "Eclipse ecosystem" (question 1a) is treated here as an empirical question. In scholarly research, the definition of an empirical phenomenon may be explicit, implicit (through citations to prior research), or phenomenological (West & Lakhani, 2008, p. 224). Rather than impose a definition *a priori* on theoretical grounds, this research takes the third approach by closely examining a field setting that *participating practitioners self-describe as an ecosystem*, making its meaning and properties explicit, and inducing an operational definition. Chapter 2 surveyed a plethora of meanings associated with the ecosystem concept in scholarly writing and practitioner discourse, and identified a four dimensional framework in which to situate the Eclipse ecosystem. The method to do this has five steps: (1) code instances of "Eclipse ecosystem" and related terms occurring in various data sources including interviews, presentations by practitioners, archived press releases, Eclipse Foundation documents, and publications by Eclipse Foundation staff; (2) group similar instances into categories according to emergent themes discovered in the data; (3) construct a timeline to identify temporal patterns; (4) compare participant results with constructs and definitions in the literature; and (5) review results with practitioners to solicit and incorporate feedback.

Likewise, the relationship between the ecosystem and community (question 1b) is identified in five steps: (1) code data sources for mention of Eclipse components and relationships between components; (2) group codes according to source, destination, and direction; (3) construct a diagram of components and relationships between components (including the Eclipse ecosystem specified by the answer to question 1a);
(4) operationalize each component, first as a management construct with a corresponding literature, and second, situating that construct within the IAD framework; (5) continually compare the operationalized system with emergent results from the nested cases and the institutional structure of the Eclipse ecosystem (question 1c); and (6) review results with informed practitioners and solicit and incorporate feedback.

Identifying the institutional structure of the Eclipse ecosystem (question 1c) is shown in Figure 6 to be centrally situated within the research design: it consumes the results of question 1a and 1b and the results of cross-case and embedded case analysis at all three nested levels, and it informs subsequent case questions about company participation, and feeds back to question 1b about relationships within the field setting. It is the major effort of this research with regards to data collection and analysis.

The data collection framework developed in chapters 2, 3, 4 and 5 provides empty "table shells" (Yin, 2003a, p. 75; Miles & Huberman, 1994) to populate with data from the nested cases. Table 14 and Table 15 summarize the table shells for two *action arenas* of central interest: the project level at which individuals in the development community make decisions about day-to-day participation (Table 14), and the level of economic actors at which profit-seeking companies and individuals make decisions about organizational participation in the business ecosystem (Table 15). The row headings in the first column of each table are the IAD elements from Table 13 (chapter 5) that provide meta-categories for coding qualitative data at the nested cases. The second column summarizes data sources for the category (mainly results at nested levels of case analysis detailed in subsections 6.2.3, 6.2.4, and 6.2.5) or answers to question 1a or 1b (indicated by *italics*). The precise meaning of "participation" is not assumed but rather allowed to emerge from the case data at the project and company levels. The shell table for the third *action arena* of central interest – polycentric governance structures that shape and reshape the operating rules for projects and companies (Table 16) – includes rows for membership, scope of authority, and changes to operating rules. This is a subset of IAD elements of primary interest at the *policy level* that establishes the operating rules and the *constitutional* level that defines the rules for who participates at the policy level (section 5.1; Kiser & Ostrom, 1982; Ostrom, 1990).

Table 14: Table shell of the institution of individual participation

IAD Element	Data Sources
Underlying factors	
Rules-in-use	Governance documents. Field notes. Cross-case results of the nested project cases and governance cases. Eclipse platform design rules.
Community attributes	Attributes of the Eclipse development community: observations from field notes, nested project cases, and community surveys.
Resource attributes	From question 1b: Salient attributes of the Eclipse platform.
Action arena	
Actors	From question 1a: *Individuals with intrinsic and extrinsic motivations.*
Action situation	From question 1a: *Participation in the Eclipse open source developer community.*
Patterns of interaction	Patterns observed from the nested project case study.
Outcomes	Project outcomes of interest to Eclipse practitioners: field notes, documents, interviews.
Evaluative criteria	High positive measures of successful project outcomes.

Table 15: Table shell of the institution of participation by economic actors

IAD Element	Data Sources
Underlying factors	
Rules-in-use	Governance documents. Field notes. Cross-case results of the nested company cases and governance cases. Eclipse platform design rules.
Constituent attributes	Attributes of the Eclipse business ecosystem: observations from field notes, nested company cases, and secondary sources.
Resource attributes	From question 1b: Salient attributes of the Eclipse platform.
Action arena	
Actors	From question 1b: *Economic actors: companies and individuals seeking profit.*
Action situation	From question 1b: *Participation in the Eclipse business ecosystem.*
Patterns of interaction	Patterns observed from the nested company case study.
Outcomes	Ecosystem outcomes of interest to Eclipse practitioners: field notes, documents, interviews.
Evaluative criteria	High positive measures of ecosystem outcomes.

Table 16: Table shell for governance groups

Category	Data sources
Membership	Formal rules of membership specified in governance documents. Observations from field notes, governance cases, interviews, community discourse.
Scope of authority	Governance documents; observations from field notes; within-group discourse; meeting minutes; interviews; community discourse. - May include subcategories if salient: e.g., responsibilities, obligations, etc.
Changes to the operating rules	Governance documents.

The Eclipse platform (introduced in section 1.2 and examined closely in section 7.4 and subsection 7.8.4) anchors the context of all three action arenas; its *resource attributes* and *rules-in-use* emerge from triangulating insights from the top-level case with insights from the project-level cases. The characteristics of each action arena, the rules-in-use, and the outcomes, patterns of interaction, and evaluation criteria emerge from analysis at the nested project case and nested company case.

Tensions between the community and ecosystem and strategies for tension management (question 1d) are identified in three steps: (1) code data sources for possible tensions and resolution mechanisms; (2) group codes into similar themes, and (3) review results with informed practitioners to solicit and incorporate feedback. As argued in section 1.2, Eclipse is interesting, in part, because it appears to adequately manage any tensions between community and ecosystem. *A priori*, it was expected that insights from this question may provide possible answers to question 1b below.

The benefits of company participation (question 2a) emerge from cross-case analysis of the company-level study, informed by respondent interviews at all levels. Informed practitioners reviewed and provided feedback on preliminary results.

The characteristics of the Eclipse ecosystem that promote and enable participation (question 2b) emerge from mapping the benefits of company participation (the answer to question 2a) to the characteristics of the institutional structures of the Eclipse (the answer to question 1a). Explanation building (Yin, 2003a, p. 120-122) and tabular data displays (e.g., Miles & Huberman, 1994) are further informed by the responses of interview respondents to direct questions concerning their views on emerging findings.

6.2.3 Nested case study of Eclipse projects

The case study question at the project level is the following:

What are the characteristics of the institution that structures individual participation in the Eclipse open source software development community?

This question is motivated by question 1c of the top-level case which seeks to describe and explain the Eclipse ecosystem using the analytic apparatus of the IAD framework[62], and an *a priori* expectation (section 1.2), later supported by case evidence and emergent theory (chapters 7 and 8), that the Eclipse ecosystem and Eclipse community are deeply interconnected. This question is answered by completing the project-level cells of the IAD table shells from question 1c (i.e., Table 14), drawing on cross-case analysis of project cases. The analytic strategy for project cross-case analysis is tabular data displays (Eisenhardt, 1989; Miles & Huberman, 1994) and explanation-building (Yin, 2003a).

For each individual project case, the case study questions are the following:

1. **What are the software artifacts produced by the project? What do the artifacts do? Who uses them? In what way are these artifacts interesting or valuable?**
2. **What is the project history? When and how did it form? What is the timeline of major events?**
3. **How is the project organized? How does it relate to other projects?**
4. **How do participants interact and make decisions?**
5. **Who participates in the project? In what ways do participants participate?**
6. **What new participants have recently joined the project? How and why did new participants join?**
7. **How do participants promote participation?**
8. **What rules-in-use structure project interactions?**

[62] The argument that participation in the Eclipse development community is structured largely by Eclipse projects requires case evidence and theory developed in chapter 8. Briefly, the *Eclipse Development Process* (version 2.5, section 4, p. 5) requires that activity to create and maintain Eclipse platform software occurs within Eclipse projects. Each Eclipse project has a development team comprised of individual committers and contributors to that project (EDP 2.5, section 4.7, p. 8). The Eclipse development community is the population of all committers and contributors (EDP 2.5, section 2.3, p. 3) to all projects.

The unit of analysis is the *Eclipse project* – a technical term within the Eclipse field setting referring to the "main operational unit at Eclipse" *(Eclipse Development Process*, version 2.5, section 4, p. 5) and the context in which Eclipse software development occurs[63]. The population of Eclipse projects is approximately 250 total projects with approximately 120 projects currently active[64]. *A priori*, the case study protocol (Muegge, 2010) had anticipated approximately five project cases selected by theoretical sampling logic – no fewer than four (the smallest number of cases recommended by Eisenhardt, 1989), and sufficient to provide replication and diversity along dimensions of interest (described in section 8.5). Eisenhardt (1989, p. 545) writes: "While there is no ideal number of cases, a number between 4 and 10 usually works well. With fewer than 4 cases, it is often difficult to generate theory with much complexity, and its empirical grounding is likely to be unconvincing."

Following approximately 18 months of observation and exploratory data collection, five case projects were selected by August 31 2010. Section 8.5 identifies the five projects and the rationale motivating their selection[65]. Each case project was

[63] Chapter 8 closely examines the structure, rules and community norms of Eclipse projects.

[64] Estimates on the number of total (lifetime) and active Eclipse projects are from presentations at the Eclipse Annual General Meeting at EclipseCon in Santa Clara California in March 2010.

[65] A full appreciation of the rationale for project case selection requires case evidence presented in chapters 7 and 8, and theory developed in chapter 7. Two brief illustrative examples are provided here. First, the sample includes projects that are community-driven (where multiple companies employ committers) and projects that are primarily company-driven (where one company employs all or nearly all committers on the project), with replication. Including both in the sample was expected to elucidate similarities and differences in the rules-in-use and action arenas for individuals and companies making participation decisions (both incumbent and entrant). Second, the sample includes a range of project launch dates: three projects that were underway when few processes were in place prior to the transition of Eclipse from an industry consortium to a vendor-neutral foundation, and two projects launched more recently within the Eclipse Foundation after the *Bylaws of the Eclipse Foundation* (2003, 2008) and a version of the *Eclipse Development Process* (2003, 2007, 2008, 2010) were in effect. Section 8.5 elaborates in more detail, including variation along three other salient dimensions.

observed longitudinally for a six-month period beginning September 2010 and continuing to February 2011, providing abundant opportunities to directly observe project activity over a range of events and contexts[66] through the same communication channels employed by developers[67].

Each project case attends closely to the archival records for the eighteen-month time period beginning June 2009 and continuing through to November 2010. This time period covers the full twelve-month release cycle for the 2010 Eclipse Helios release, and the first half of the release cycle for the 2011 Eclipse Indigo release. Data collected on this time interval includes commit counts for each project committer (by month), activity on project mailing lists, and new committer nominations and elections.

Other project-level data sources include project documents and resources, Eclipse Foundation documents, published books and articles, community blog discussions, interviews with project participants, field notes of the principal researcher's observations and direct interaction with project participants, and the hands-on experience of the principal researcher using the project software artifacts. Project documents and resources include initial project proposals, project charters of top-level projects, documents from project reviews, intellectual property logs, and content of project websites and wikis.

[66] More specifically, this time interval included milestone builds for the 2011 Eclipse Indigo release (M2, M3, M4 and M5), the second service release (SR2) for the 2010 Helios release, Eclipse Summit Europe 2010 (November 2-11, Ludwigsburg, Germany), project roadmapping activities for Indigo, and planning for EclipseCon 2011 events (March 20-24, Santa Clara, California). Chapters 7 and 8 explain these terms.

[67] Project communication channels included project mailing lists, Bugzilla updates, cross-project mailing lists, user forums and other project resources . Salient cross-project communication channels included the cross-platform-issues-dev mailing list for the Indigo annual release train (subsection 8.4.1), the epp-dev mailing list for Eclipse packages (subsection 8.4.2), and Eclipse community blogs at *Planet Eclipse*. Chapter 8 and subsection 9.4.1 explain these terms and elaborate on Eclipse project and Eclipse community communication channels and IT infrastructure maintained by the Eclipse Foundation staff.

Salient documents from the Eclipse Foundation include governance documents, press releases about the project, and salient discussion in the minutes of the Eclipse Foundation board of directors and Eclipse members meeting presentations.

The case notes document for each case project follows a nine-section template developed from the individual project-level case study questions. The first seven sections – software, timeline of project milestones, background, organization, interaction and decision-making, participation, and observations of rules-in-use – each directly address one or more project case study questions. The eighth section is a bibliography of sources. The ninth section is an appendix for miscellaneous case notes and reflections about the case. The individual project case study questions and the project case notes template evolved throughout data collection and analysis, consistent with the iterative feedback loops of Figure 5. They are presented here in their final form at the completion of the research.

The completed case notes documents for the individual project cases are artifacts used for cross-case analysis addressing the project-level case study question to discover the characteristics of the developer community institution. The longest project case notes document is twenty-three single-spaced type-written pages, and the shortest is sixteen pages.

6.2.4 Embedded case study of polycentric governance structures

The case study question at the governance level is the following:

How do Eclipse governance groups interact with the institutions that structure the Eclipse open source software development community and Eclipse ecosystem?

This question is motivated by question 1c of the top-level case which seeks to describe and explain the Eclipse ecosystem using the analytic apparatus of the IAD framework, and an *a priori* expectation (section 1.2), later supported by case evidence and emergent theory (chapters 7 and 9), that Eclipse governance and company participation in the Eclipse ecosystem are deeply interconnected. This question is answered by completing the governance-level aspects of the IAD table shells from question 1c of the top-level case (i.e., an instance of Table 16 for each governance group), and constructing a visual timeline of governance events and changes to the Eclipse operating, policy, and constitutional rules. The objective is an understanding of what governance groups exist, who participates in each group, how each group influences the operating rules of the community and ecosystem.

For individual governance group cases, the case study questions are as follows:

1. **What is the membership of the governance group? How is membership determined?**
2. **What is the mandated scope of influence? What are the formal powers and duties? What other groups are impacted by the actions of this group?**
3. **How can the rules for the governance group be changed?**
4. **How do participants interact and make decisions?**
5. **How do the group's actions impact the institutional structures of the community and ecosystem?**

The embedded units of analysis are the various organizational subunits of polycentric governance not connected to a particular Eclipse project. *A priori*, the sample of governance groups had been expected to be a census of the five major organizational subunits known at the time of the case study protocol (Muegge, 2010) – the Eclipse board of directors, the Eclipse Executive Director and Eclipse Foundation staff, and the three Eclipse councils – and any additional groups discovered to be salient as the research was underway. No additional cases were added, but three changes from the case study protocol are noted here. First, although the Executive Director and Eclipse Foundation staff are examined within the same governance case, they are treated as separate governance entities in chapter 9 with their own table instances. Second, additional instances of governance table shells were created for some roles within the Eclipse community meritocracy: specifically, Eclipse committers (Table 30), Eclipse project leads (Table 31), and Eclipse project management committee (PMC) members and leads (Table 32). These tables are presented with the project results where they are most salient (subsection 8.1.3) and also discussed with the governance results (chapter 9). Third, a governance table shell instance is provided for the Eclipse membership-at-large detailing the powers and obligations of each Eclipse membership class (Table 22).

Data sources include Eclipse Foundation governance documents, meeting minutes, documents produced by the group, community discussion in participant blogs, and interviews with participants. The public communication channels for each governance group, including mailing lists, meeting minutes, and Bugzilla activity, were observed longitudinally for the same interval as the project cases (subsection 6.2.3): a six-month period beginning September 2010 and continuing through to February 2011.

The case notes document for each governance group case includes a five-section template developed from the governance-level case study questions. However, as Yin (2003a, p. 43) points out, different data collection and analysis techniques may be appropriate for different units of analysis in an embedded case study, thus the governance case reports show greater individual variation from the template than do either the project or company reports. The first section of the template, formal rules, addresses the first three governance case questions about membership, scope and the process for changing the operating rules. The next two sections address activities within the governance group: the second section examines the group's activities over time and the third section sorts the coded activities into emergent categories according to outcomes. The fourth section is a bibliography of sources. The fifth section is an appendix for miscellaneous case notes and reflections about the case. Some case reports include additional sections for observations or coded data useful for other nested case reports or for the top-level case. For example, the board of directors case notes document includes a lengthy section that codes the *Abridged minutes of the Eclipse Foundation board of directors* for insights to the project, governance, and company cases, and questions 1a, 1b, and 1d of the top-level case. The longest governance case notes document is thirty-two single-spaced type-written pages and the shortest is ten pages.

6.2.5 Nested case study of participating companies

The case study question at the company level is the following:

What are the characteristics of the institution that structures company participation in the Eclipse ecosystem and contribution of resources to the Eclipse community?

This question is motivated by question 1c of the top-level case which seeks to describe and explain the Eclipse ecosystem using the analytic apparatus of the IAD framework. This question is answered by completing the ecosystem-level aspects of the IAD table shells from question 1c of the top-level case (i.e., Table 15), drawing on insights from cross-case analysis of company cases. The analytic strategy for company cross-case analysis is a combination of tabular data displays (Eisenhardt, 1989; Miles & Huberman, 1994) and explanation-building (Yin, 2003a).

For each individual company case, the case study questions are as follows:

1. **What is the company history? What is the timeline of major events salient to participation in Eclipse?**
2. **What is the company business model? (Alternatively, how does the company create value for stakeholders and generate profits?)**
3. **How does the company participate in Eclipse?**
4. **How does the company benefit from participation?**

The unit of analysis is the *participating company*, where participation in Eclipse includes (at a minimum) membership in the Eclipse Foundation and contribution of resources to one or more case projects. The population of companies at this level of case analysis is therefore determined by the selection of cases at the project level of analysis (subsection 6.2.3) – although there are approximately 170 Eclipse member organizations, all organizations eligible for selection are participants in at least one case project. *A* *priori*, the case study protocol (Muegge, 2010) had anticipated approximately eight company cases – no fewer than six (for adequate variation) and no more than ten (the largest number of cases recommended by Eisenhardt (1989) for a single cross-case research study). Eisenhardt (1989, p. 545) writes: "With more than 10 cases, it quickly becomes difficult to cope with the complexity and volume of the data."

At the time of case selection for the project cases (August 31 2010), a list of sixteen candidate companies was prepared. During the longitudinal observation of project cases (September 2010 to February 2011; described in subsection 6.2.3), the principle researcher observed closely any project activities and other activities in the Eclipse community by employees of the candidate companies. By November 2010, eight case studies were underway, selected according to theoretical sampling logic for the potential of each case to replicate and extend emergent theory. No additional cases were added after November 2010. Section 10.1 identifies the eight companies and the rationale motivating their selection.

Data sources include direct observation of company activities within case study projects during the six-month period of September 2010 to February 2011 (see subsection 6.2.3), company documents (including financial reports, websites, press releases, and blogs by company staff), publications about the company (books, articles, and case studies), salient Eclipse documents from the Eclipse Foundation (including announcements, press releases, and minutes of membership meetings and meetings of the board of directors), and interviews with participants.

The case notes document for each company follows an eleven-section template developed from the company-level case study questions. The first three sections – a brief summary, a timeline of salient events, and an overview of the company's technology – introduce the company. The fourth section, background, is a narrative of the salient company history, emphasizing participation in Eclipse. The fifth section articulates the company business model using a business model framework adapted from Johnson et al. (2008; Johnson, 2010) surveyed in chapter 3. The sixth section examines competition and competitive strategy. The seventh section examines the company's participation in Eclipse and the benefits of participation. The eighth section examines company participation in other open source software activity. The ninth section is notes on key people within the company. The tenth section is a bibliography of sources accessed to create the case. The eleventh section is an appendix for miscellaneous case notes and reflections about the case. The longest company case notes document is thirty single-spaced type-written pages, and the shortest is twelve pages.

6.2.6 Consolidated summary of data sources

Data sources at all four levels of analysis include the following:

- *Direct observation* of participants (e.g., conferences, meetings and conference calls) and participant communication channels (e.g., the various public communication channels of the Eclipse developer community).
- *Documents and archival records*, including governance documents, development process documents, and quantitative project metrics from the Eclipse dashboards and dash project. The publicly available archival data on the Eclipse field setting

are vast. Detailed and comprehensive archives of project, governance, and community communications – including emails, forum posts, bug reports, meeting minutes, project reviews, and commits to the source code repositories – are often complete back to the launch of the Eclipse Foundation in 2004 and are sometimes complete back to the launch of the Eclipse consortium in 2001.

- *Interviews* with informed participants, including project committers, project leads, members project management committees (PMCs) of top-level projects, members of the three Eclipse councils (requirements, planning, and architecture), Eclipse Foundation staff, directors on the Eclipse Foundation board of directors, and managers at member companies. With plentiful opportunity for direct observation of participant communications and an abundance of documents and archival records, the role of interviews within this research design is mainly to (1) triangulate key pieces of evidence (Jick, 1979), (2) fill-in gaps in the archival record, (3) better understand practitioners' perspectives on events, (4) help the principle researcher to interpret and make sense of observations, and (5) solicit and obtain feedback on emergent insights.

Because the research design is nested and multilevel, some sources inform two or more cases at multiple levels of analysis.

Table 17 summarizes the main sources of evidence examined at each level of case study analysis. Table 18 rearranges the information from Table 17 according to data source: direct observation, documents and archival records, or informant interviews. Subsections 6.2.7 and 6.2.8 report on the quantity of data and analysis.

Table 17: Summary of data sources (by unit of analysis)

			Data sources		
Level of analysis	Unit of analysis	#	Direct observation	Documents and archival records	Interviews

Top-level case	Eclipse field setting	1	1 new member jump-start meeting;
			4 members meetings;
			presentations and panels by Eclipse Foundation staff.
				Governance documents;
				community blogs and conversations;
				press releases;
				archive of members meeting minutes.
					Interviews at all levels of analysis also inform the top-level case.

Projects	Eclipse project	5	Hands-on with software artifacts; EclipseCon presentations and meetings; project communications.	Project dashboard; project archives; community blogs.	Project leads, committers, and PMC members.
Companies	Company participating in Eclipse projects	8	Participation in case study projects and governance groups.	Company website, press coverage, public documents	Managers and software developers.
Governance groups (Eclipse Foundation)	Board of directors	1	none (board meetings are closed to observers)	Abridged meeting minutes; documents; board resolutions.	Directors, officers, and past directors.
	Eclipse Foundation (Executive Director and staff)	1	Activity in case study projects and governance groups; site visit to the Eclipse Foundation offices.	Blogs; presentations; published articles.	Eclipse Foundation Executive Director and staff members.
	Requirements council (RC)	1	None (RC is currently inactive)	Meeting minutes; wiki; website; documents.	Council members.
	Planning council (PC)	1	Planning council mailing list.	Meeting minutes; wiki; website; documents; multiple mailing lists.	Council members.
	Architecture council (AC)	1	Architecture council mailing list and bug reports.	Meeting minutes; wiki, website, documents, bug reports, mailing list.	Council members.
Totals	8 types of unit of analysis	18 nested cases			

Table 18: Summary of data sources (by type)

Source	Type	#	Details
Direct observation	Meetings	4	Eclipse members meetings (2009Q1, 2010Q1, Q4, 2011Q2)
	Conference Events	3	EclipseCon 2009, EclipseCon 2010, Eclipse Summit Europe 2010; Project BoFs (Birds of a Feather sessions) at Eclipse events.
	Project activity and community communications		Six months of *direct observation* of project and community communication channels (September 2010 - February 2011): developer mailing lists, cross-project mailing lists, Bugzilla activity, user forums, community blogs, project wikis, etc.
	Software artifacts	5	Hands-on use by the principal researcher of the software artifacts produced by each case study project.
Documents and archival records	Eclipse projects and community conversations.	5	Theoretical sampling of projects on diversity of company participation, project age, type of software asset, project organization, and membership segment. Project documents (charters, release reviews, IP logs), Eclipse Foundation documents (press releases, announcements), meeting minutes, community blogs, developer mailing lists, Bugzilla activity, user forums, Eclipse dashboards, published books and articles. Close attention to June 2009 – November 2010 (18 months).
	Companies participating in Eclipse projects	8	Theoretical sampling from the population of companies participating in case study projects. Company websites, press releases, public documents.
	Eclipse Foundation		Governance documents; press releases; announcements.
	Eclipse staff		Blogs @ Planet Eclipse. Posts to mailing lists.
	Board of directors		Abridged meeting minutes (quarterly from 2004Q2)
	Councils	3	Meeting minutes; documents; wikis; mailing list archives.
	Eclipse platform		Eclipse dashboards, published books and articles.
	Processes (as rules-in-use)		*Bylaws of the Eclipse Foundation* (2003, 2008) *Eclipse Membership Agreement* *Eclipse Development Process* (EDP) Eclipse *Intellectual Property Policy* *Eclipse Public License* (EPL) The Eclipse.org website, and published books and articles.
Interviews	Interview sessions	30	Includes multiple interview sessions with some informants.
	Individual informants	20	Most informants (18 of 20) held multiple roles within the Eclipse field setting (see below).
	Eclipse Foundation ED and staff	6	Executive Director, other directors, and staff.
	Directors of the board of directors	4	Theoretical sampling from the population of ~18 current directors and many former directors.
	Members of Eclipse councils	8	Theoretical sampling from council memberships of ~ 13, 22, and 55 members, respectively.
	Employees of case companies	14	Theoretical sampling from the population of individuals employed by the 8 case study companies.
	Individuals participating in case projects.	14	Theoretical sampling from populations of ~60 committers, ~10 project and subproject leads and co-leads, and ~25 PMC members of the 5 case projects.

6.2.7 Interview field procedures

All interactions with human subjects conform to Carleton University research ethics guidelines and the *Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans* (2005). Potential informants were provided a *Letter of Introduction* and completed a *Consent Form* approved by the Carleton ethics committee.

The format of each participant interview was *semi-structured*, with *open-ended questions* to initiate stories and *probing questions* to establish details, following interview best practices recommended by Foddy (1993) and Yin (2003a). Prior to each interview, the principal researcher reviewed the developing case notes documents, noting gaps, discrepancies, and interesting insights, and prepared a brief *interview guide* of questions intended for that particular informant. Some questions covered general lines of inquiry; other sought to elicit facts or opinions unique to the informant.

Interviews were conducted by the principal researcher, either face-to-face at Eclipse events, or by telephone or Internet technologies. At the start of each interview, the interviewer reminded the informant of the research objective and the general topics to be discussed. The dialogue throughout the interview was interactive and conversational, loosely following the topics and order of the interview guide, with probing questions to elicit concrete examples to back up abstractions. During each interview, the interviewer recorded key-point notes. Where circumstances were favourable and informants provided explicit permission, interviews were digitally recorded for later review to correct and expand key-point notes. Immediately following each interview, the interviewer recorded subjective impressions as an addendum to the key-point notes.

Table 19 provides examples of interview questions that were posed to multiple informants at each level of case analysis (project cases, governance cases, and company cases). The questions posed to any particular informant depended on that informant's background, their areas of participation in Eclipse, and the developing content of the related case notes documents. Interview questions were intended to fill in gaps in the case table shells and case notes documents, to triangulate other data sources, and to understand informant perspectives on the field setting and research questions.

Table 19: Sample interview questions.

	Introductory questions **Sample questions intended to elicit information on the participant's involvement in the field setting and to provide context for later questions about specific cases.**
1	How did you first get involved in the Eclipse Foundation?
2	In which areas of Eclipse have you been involved? Why those projects? Why those particular councils or governance groups?
3	How has Eclipse changed in the time that you have been involved?
	Project-level questions **Sample questions intended to elicit information about a case project [PROJECT-X].**
4	Within [PROJECT-X], how is the work divided among the committers? Who does what?
5	How do you decide what aspects of [PROJECT-X] to work on?
6	In what ways is your employer or manager involved in the work you do on [PROJECT-X]?
7	Tell me about a recent decision at [PROJECT-X]. How was the outcome reached? Who was involved? What were the impacts?
	Governance-level questions **Sample questions intended to elicit information about an Eclipse governance group [GROUP-Y].**
8	What does [GROUP-Y] do within Eclipse? What functions does it perform?
9	Tell me about a recent decision at [GROUP-Y]. How was the outcome reached? Who was involved? What were the impacts?
10	How has [GROUP-Y] changed over time?
	Company-level questions **Sample questions to elicit information about a case company [COMPANY-Z].**
11	In what ways does [COMPANY-Z] participate in Eclipse?
12	How does [COMPANY-Z] generate revenue?
13	How does [COMPANY-Z] benefit from its participation in Eclipse?
14	Anyone can use Eclipse software without membership or participation in the Eclipse Foundation. If membership and participation are optional, why does [COMPANY-Z] participate?

As summarized in Table 18, data collection included a total of thirty interviews conducted with twenty individual informants. Eleven informants were interviewed once, eight informants were interviewed twice, and one informant was interviewed three times. Interviews varied in duration from thirty minutes to more than two hours, with a typical interview duration of one hour. All informants held at least one salient role within the Eclipse field setting; roles included committer, lead or co-lead on a case project, member or lead of a case project PMC, employee or former employee of a case company, or member of a governance group (foundation staff, the board of directors, or a cross-project council). Eighteen informants held more than one role and could thus inform multiple cases: seven informants held two roles, three informants held three roles, five informants held four roles, two informants held five roles, and one informant held eight roles.

6.2.8 Case study database and coding procedures

Yin (2003a, p. 101) recommends that case study researchers maintain a *case study database* of raw data and project artifacts that is separate and distinct from the contents of the case study report. The case study database for this research includes three components: (1) a collection of physical documents that are annotated and coded manually, (2) a digital archive of electronic documents, software, spreadsheets, and other digital files stored on a portable flash drive, and (3) a collection of interim research artifacts – field notes, case notes, tabular materials and researcher memos – stored electronically with the digital archive and periodically printed for inclusion with the physical documents. The case study database is secured and retained by the principal researcher, consistent with the requirements of the Carleton research ethics committee.

The collection of physical documents includes a bankers box of files and two binders of documents, together comprising more than four thousand pages of paper.

The digital archive of electronic documents consumes approximately 2GB of digital storage. It includes complete digital copies of documents and archival records listed in Table 17 and Table 18, an archive of mailing list activity on lists related to case projects and governance groups between September 2010 and August 2011, and selected excerpts from archived and directly observed community discourse, including salient text from mailing lists and newsgroup posts, Bugzilla entries, blog entries and comments, meeting minutes, and wiki content.

Of the interim research artifacts, the most lengthy (approximately 225 pages) is the *field notes* of the principal researcher – the primary record of direct observations, interview notes, annotations of community blogs, summaries and annotations of Eclipse governance documents, notes about IT infrastructure, press releases, trade press articles, preliminary notes about projects, companies, people, and governance groups prior to case selection, and memos of emergent insights and ideas under development.

The second most lengthy research artifact (approximately 125 pages) is the *top-level case notes*, including the compiled coding and analysis for the top-level case study questions, theorizing about observations, and notes on cross-case analysis of the project, governance, and company cases. Twenty other case notes documents include five case notes on Eclipse projects, eight case notes on participating companies, and five case notes on embedded polycentric governance groups. Other project artifacts include a general *interview guide*, notes on the Eclipse platform, an annotated bibliography of published

Eclipse research, and an annotated bibliography of the architecture of participation.

Data sources expected to inform only a single case (e.g., a project release review document) are annotated and coded in the case notes document for that case. Data sources strongly associated with a particular case but with potential to inform other cases (e.g., meeting minutes of the board of directors) are annotated and coded with the most salient case, then salient excerpts are copied to other case notes documents. Data sources expected to inform multiple cases (e.g., press releases) are annotated and coded in the principal researcher's *field notes*, then salient excerpts are copied to other documents.

6.3 Presentation of results

Results for each of the four levels of this research design – the top-level case, the multiple case study of Eclipse projects, the embedded case study of governance groups, and the multiple case study of participating companies – are presented in chapters 7, 8, 9 and 10, respectively, in the same sequence in which the steps of the research method were presented in subsections 6.2.2, 6.2.3, 6.2.4 and 6.2.5.

The specific order and organization of results follows what Yin (2003a, p. 154) calls a "theory-building" structure: each chapter develops part of a theoretical argument, then chapter 11 brings together the results to answer the research questions and provide a synthetic explanation of company participation in the Eclipse ecosystem. As with the explanation of the research method in section 6.2, there is no single linear path through the research results without referencing results reported in later sections: Figure 8 showed that insights from the top-level case both structure and inform the nested cases, just as results of the nested cases inform the top-level case and other nested cases.

Yin (2003a, ch. 6) identifies four varieties of written case reports, distinguished by their differing emphasis on traditional case narratives, structured narratives, or cross-case analysis. Presentation of case studies at the top-level case (chapter 7) and the governance level (chapter 9) follows what Yin calls a *multiple narrative format*. In chapter 7, the the top-level case of the Eclipse field setting is presented as a single extended narrative with sections examining each of the embedded components. In chapter 9, the embedded governance cases are presented as abbreviated narratives, approximately following the order of the governance case study questions and the topics of the case notes document. Presentation of case studies at the project level (chapter 8) and company level (chapter 10) adopts Yin's fourth alternative: a report consisting largely of *cross-case analysis*, structured and ordered by cross-case topic, with information about individual project and company cases presented in "abbreviated vignettes" (Yin, 2003a, p. 148) where useful to illustrate specific points or to help clarify arguments developed from cross-case results. The cross-case presentation format is well-matched to the project and company portions of this research, with their relatively large number of individual cases, abundance of data for each case, and clarity about the variables and constructs of central interest. It is also the format recommended by Eisenhardt (1989) and Eisenhardt & Graebner (2007) for structuring theory-building cross-case studies.

After reporting the case evidence, each chapter concludes with a section on synthesis and theory development that integrates that chapter's results into the emerging theory successively developed in previous chapters. Theorizing about each of the first five case study questions is presented at the point in the document after most observations salient to the question have been reported (Table 20). Theorizing about the the sixth and final case study question (question 2b) is developed in the discussion (chapter 11).

Table 20: Presentation of evidence and theorizing

Case study question	Chapter where observations and theorizing are reported
1a. What does "Eclipse ecosystem" mean to Eclipse practitioners?	Chapter 7: Top-level case
1b. How does the Eclipse ecosystem relate to the Eclipse developer community?	Chapter 7: Top-level case
1c. Can the Eclipse ecosystem be described and explained using the constructs and analytic apparatus of institutional theory and the IAD framework?	Developed successively in four steps in chapters 7, 8, 9 and 10.
• Conceptual framework; relationships between parts; characteristics of the Eclipse platform.	Chapter 7: Top-level case
• Characteristics of the institution structuring individual participation in the Eclipse development community	Chapter 8: Eclipse projects
• Interaction between governance groups and Eclipse institutions	Chapter 9: Governance groups
• Characteristics of the institution structuring company participation in the Eclipse ecosystem and contribution of resources to the Eclipse community and platform.	Chapter 10: Company participation
1d. Are there tensions between ecosystem and community? If so, how are those tensions managed?	Chapter 10: Company participation
2a. What are the motives, incentives, and benefits for company participation?	Chapter 10: Company participation
2b. What is it about the Eclipse ecosystem that enables, promotes, and sustains participation	Chapter 11: Discussion

6.4 Validity and reliability

Yin (2003a, p. 34) identifies eleven "case study tactics" for case-based research designs producing results with strong *internal validity*, *construct validity*, *external validity*, and *reliability*. Table 21 explains how nine of those tactics, along with methodological advice from Poole et al. (2000) and Poteete et al. (2010), were built into the research design of this thesis.

Table 21: Tactics for validity and reliability (adapted from Yin, 2003a, p. 34)

Test	Tactic[68]	Actions taken in this research design
Construct validity correct operational measures for concepts being studied	Multiple sources of evidence	Answers to research questions *triangulate* multiple sources of evidence (Jick, 1979): archival data from multiple perspectives, direct observation of participants and participant communications, and practitioner interviews. Subsection 6.2.6, Table 17 and Table 18 document these data sources.
	Establish chain of evidence	*Case notes* documents for each project, company, and governance case link case study questions to the data collected and conclusions drawn.
	Key informants review draft results	Key informants reviewed preliminary results for factual accuracy and omissions, and provided comments on preliminary explanations. Case notes were revised and improved by respondent feedback.
Internal validity[69] correct causes and relationships	Explanation-building	*Explanation-building* is the general analytic strategy (Yin, 2003a, p. 109) for this study, especially for the second guiding research question. Narrative explanations were developed iteratively, starting during the first draft cases study reports, and successively revised as data were added. Explanations were discussed with key informants for further insights.
	Address rival explanations	*Rival explanations* were pursued throughout data collection and analysis, and are documented in the discussion (section 11.4).
External validity the domain to which findings can be generalized	Use theory in single case studies	Explanation-building and theorizing at the top-level case (where replication was not possible within the defined scope of this study) employed conceptual building blocks from the research and practitioner literatures on business ecosystems (chapter 2), management (chapter 3), architecture of participation (chapter 4), and commons governance (chapter 5).
	Use replication logic in multiple case studies	*Replication logic* is central to the design of the nested case studies. Project and company cases were selected by theoretical sampling for replication as well as variation and extension (sections 8.5 and 10.1).
Reliability can be repeated with same results	Case study protocol	A *case study protocol*, consistent with the recommendations of Yin (2003a, p. 69), was crafted prior to data collection, documented within the thesis proposal (Muegge, 2010), and revised throughout the study.
	Case study database	A *case study database* of raw data and project artifacts was maintained separately from the case study report (subsection 6.2.8), consistent with the recommendations of Yin (2003a, p. 101).

This chapter has articulated the research design and method to answer the guiding research questions of this thesis. The next four chapters present the research results, beginning with the top-level case of the Eclipse field setting.

[68] Gibbert et al. (2008) provides an expanded list of twenty-three case study research design procedures for validity and reliability; however, most are specialized niche procedures not relevant here. Guba & Lincoln (1994), wary of positivist assumptions of a single absolute account of social reality, recommend alternative *trustworthiness* criteria of *confirmability*, *credibility*, *transferability*, and *dependability*, each approximately paralleling one of Yin's criteria, respectively, and Bryman et al. (2011) recommends methodological techniques to produce high-quality research according to the Guba & Lincoln (1994) trustworthiness criteria. The actions recommended by Bryman et al. (2011) are similar to the tactics recommend by Yin (2003a) to produce results with strong validity and reliability in the positivist tradition.

[69] Two of Yin's tactics for internal validity are excluded from Table 21. *Pattern matching* is a subset of *explanation-building*. *Logic models* are a specialized technique for events that unfold over time.

7 Eclipse Foundation, platform, community, and ecosystem

This chapter introduces the top-level case study of the Eclipse field setting, comprised of the Eclipse Foundation, the Eclipse platform of open source software and complementary assets, the Eclipse developer community, and the Eclipse ecosystem. From case evidence, it induces a conceptual framework and preliminary constructs that are further developed in later chapters to answer the guiding research questions.

7.1 The Eclipse Foundation

As a legal entity, the Eclipse Foundation is a membership-based not-for-profit corporation, incorporated in the state of Delaware in the United States of America[70]. It applied for and received tax-exempt status under section 501(c)6 of the U.S. Internal Revenue Code as a trade association. Prior to incorporation in 2004, Eclipse was an industry consortium led by IBM (2001-2004), and prior to that, the Eclipse platform software was an internal research and development project at IBM (1998-2001).

According to the *Bylaws of the Eclipse Foundation* (2003, 2008; article 1.1), the Eclipse Foundation purpose is "to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services." The Eclipse Foundation is unusual in that it explicitly cultivates *both* an open source development community *and* a commercial ecosystem. According to Mike Milinkovich, the Executive Director of the Eclipse Foundation: "This duality is built into our bylaws, our organization and, I would assert, our DNA" (Milinkovich, 2010, p. 29).

[70] State of Delaware Division of Corporations file number 3757829, incorporated January 28 2004.

Likewise, the characteristics of vendor neutrality, extensibility, and accessibility are also embedded into the foundation's legal identity, The same opening paragraph of the *Bylaws* (2003, 2008; article 1.1) defines the *platform* as a

> *vendor neutral, open development platform supplying frameworks and exemplary, extensible tools... Eclipse Platform tools are exemplary in that they verify the utility of the Eclipse frameworks, illustrate the appropriate use of those frameworks, and support the development and maintenance of the Eclipse Platform itself; Eclipse Platform tools are extensible in that their functionality is accessible via documented programmatic interfaces.*

The Eclipse Foundation is governed by a *board of directors,* comprised of seats allotted to elected and appointed representatives from the membership according to rules specified in the Eclipse Foundation *Bylaws*. The board of directors appoints the Eclipse *Executive Director*, who reports to the board. The Executive Director leads an *Eclipse Management Organization (EMO)* comprised of a professional staff employed by the Eclipse Foundation and cross-project governance councils. The *professional staff* provide services to the Eclipse community and ecosystem, including provision and support of IT infrastructure, intellectual property management, management and maintenance of the development process, and activities to promote ecosystem development. In March 2011 there were three governance councils – an *architecture council*, a *planning council*, and a *requirements council* – comprised of representatives from the Eclipse membership and other appointees. Eclipse *members* provide the membership dues that fund the Eclipse Foundation. Table 22 describes the six Eclipse Foundation membership classes: five are open only to organizations, and one – the *committer* class – is open only to individuals.

Table 22: Eclipse membership classes

Membership Class	Description	Dues
Strategic Developer Members	*Strategic members* – both *strategic developers* and *strategic consumers* – are *organizations* that view Eclipse as a strategic platform and are investing developer and other resources to further develop the Eclipse technology. Both classes of strategic member receive a seat on the Eclipse board of directors, the requirements council, the planning council and the architecture council, and are expected to lead one or more Eclipse open source projects. *Strategic developers* are major *contributors* of technology who assign at least *eight developers* to full-time development of Eclipse technology.	Annual dues of 0.12% of revenue; US $25k minimum, US$250k maximum.
Strategic Consumer Members	*Strategic consumers* are major *users* of Eclipse technology. A strategic consumer member may reduce dues by assigning developers to Eclipse projects ($125k / developer; up to two developers).	Annual dues of 0.20% of revenue; US$50k minimum., US$500k maximum.
Enterprise Members	*Enterprise members* are *organizations* that rely on Eclipse technology as a platform for internal development projects, or act strategically building products and services on or with Eclipse, and want to influence and participate in the development of the Eclipse ecosystem. [This membership class was created in 2008]	US$125k annually
Solutions Members	*Solutions members* are *organizations* that view Eclipse as an important part of their corporate and product strategy offer products and services based on or with Eclipse, and want to participate in the development of the Eclipse ecosystem. Solutions members express public support for the Eclipse Foundation and the Eclipse platform, and plan to release a commercial Eclipse-based offering or use an Eclipse project to develop a commercial offering within 12 months. [The name of this membership class was changed from "Add-in provider" in 2008]	US$5k-US$20k annually depending on annual corporate revenue
Associate Members	*Associate members* are *organizations* that participate, and want to show support for, the Eclipse ecosystem. Associate membership is non-voting. Associate membership is free for not-for-profit organizations, standards bodies, universities, research institutes, media and publishing, government, and other organization types as defined by the Board of Directors. All other organizations, including for-profits, may become associate members by paying dues. Associate members can submit requirements, and participate in project reviews and membership meetings. [The rules for this membership class were revised in 2008]	For-profit organizations pay US$5k annually. Free for not-for-profits and other exempt organizations.
Committer Members	*Committer members* are *individuals* that are the core developers of the Eclipse projects and can commit changes to project source code. Committers may work for a member organization, or may complete the membership process independently if they are not.	No dues; elected by existing committers according to a *meritocracy*.

Compiled from the *Bylaws of the Eclipse Foundation* (2003, 2008), *Proposed Membership Changes* (2008), the Eclipse *Membership Agreement* (2010), and the Eclipse.org website (accessed August 2010).

The Eclipse Foundation does not employ any of the software developers who maintain and extend Eclipse software – instead, all development work happens within the Eclipse open source development community. Development work is organized into *projects* with roles and structures defined in Eclipse governance and process documents that apply across all Eclipse projects and are maintained by the Foundation.

7.2 A brief history of Eclipse

There are several existing accounts of the origins of Eclipse from various perspectives (e.g., des Rivieres & Wiegand, 2004; Thomson, 2004; Capek et al. 2005, Nackman, 2005; Samuelson, 2006; Wagstrom, 2009), including Harvard Business School teaching cases examining the circumstances surrounding IBM's strategic decisions to take take Eclipse outside the company (O'Mahony et al. 2005a, 2005b). This section presents a brief synopsis, drawing on prior work and insights from practitioner interviews.

Work that was to become the Eclipse Project began in the late 1990s as an internal R&D project at IBM. The context was a period of decline, transformation, and corporate turn-around at IBM. Following decades of strong financial performance driven by lucrative mainframe computer sales, IBM had begun posting substantial losses in the first quarter of 1991. From 1991 through 1993, IBM amassed a 3-year loss of nearly US$16 billion (Austin & Nolan, 2000, p. 1). According to CEO Louis V. Gerstner, Jr., who joined IBM in 1993 to either turn the company around or break it up into parts: “Only a handful of people understood how precariously close IBM came to running out of cash in 1993. Whether we would have had to file for bankruptcy, I cannot say” (Gerstner, 2002, p. 66). IBM faced “rapidly declining mainframe sales, much higher prices than those of our competitors, a lack of participation in the rapidly growing client/server (PC-centric) segment, and an alarming decline in the company's image" (Gerstner, 2002, pp. 43-44). Gerstner decided to keep the company together and to reinvent the business, transforming IBM over the next decade into the largest services business in the IT industry and opening up IBM products for interoperability with non-IBM products. Software – more specifically, the *middleware* that sits between the operating system and the application software that the user sees – would be core to the transformed IBM as the enabler for both services and interoperability.

Consistent with IBM's business direction, work began internally on an extensible platform for middleware tools that would be interoperable with other systems and able to run on a wide range of operating systems[71]. The project was assigned to IBM's Object Technology International (OTI) subsidiary, a company that IBM had acquired in 1996 and operated autonomously from IBM product lines[72]. The first application developed on the platform was a Java IDE that became the Eclipse JDT project.

On November 5, 2001, IBM announced "it is donating $40 million of software to a new independent open-source community. The Java-based open source software, code-named Eclipse, will enable developers to use software tools from multiple suppliers together, allowing developers to integrate business processes used to create e-business applications, such as those for Web services" (IBM press release, November 5 2001). IBM described Eclipse as the solution to integrate a fragmented tools industry:

[71] According to Lee Nackman, CTO of IBM Rational Software, the decision to go ahead with this development occurred in November 1998 (Nackman, 2005, p. 6).
[72] OTI was founded in 1988 in Ottawa Canada by a professor and three graduate students of the Object Oriented research group at Carleton University (Thomas, 1995).

> *Today, developers typically use tools and middleware from multiple vendors that are not designed to work together, so they are cumbersome and time-consuming to integrate. By using software tools that easily "plug into" Eclipse, developers can create higher-quality applications in less time and inherit technology developed by other vendors.*

Eclipse became open source on November 7 2001, when IBM published the source code to the Eclipse.org website under the *Common Public License* (CPL). On November 29, the *Eclipse Consortium* launched, with an initial board of stewards with representatives from Borland, IBM, Merant, QNX Software Systems, Rational Software, RedHat, SuSE and TogetherSoft (Eclipse.org Consortium press release, November 29 2001).

In January 2004, the Eclipse Consortium reorganized into the Eclipse Foundation. The launch of the not-for-profit foundation was formally announced at the inaugural EclipseCon conference (Eclipse Foundation press release, February 2 2004):

> *Eclipse has formed an independent open ecosystem around royalty-free technology and a universal platform for tools integration. Eclipse based tools give developers freedom of choice in a multi-language, multi-platform, multi-vendor environment. Eclipse provides a plug-in based framework that makes it easier to create, integrate and utilize software tools, saving time and money. By collaborating and exploiting core integration technology, tool producers can leverage platform reuse and concentrate on core competencies to create new development technology.*

In March, the Eclipse Foundation announced its founding board of directors (Eclipse Foundation press release, March 3 2004), and in June, it hired its first Executive Director (Eclipse Foundation press release, June 1 2004).

The following year, an industry analyst report from Forrester Research (Zetie, 2005, p.1) writes:

> *The battle to be the leading development tools integration framework ... is over and Eclipse has won... What started as grassroots adoption of a free and flexible IDE has quickly become the most widespread Java development environment and is beginning to spread across the rest of the application development life cycle. Debates will continue about which is the best Java IDE,*

> *but Eclipse has its eyes on bigger things: a platform to integrate disparate life-cycle tools to an unprecedented degree, and beyond that, to become an integration platform for other IT requirements.*

Subsequent growth included areas outside of the software development tools space, as a rich client platform (RCP) for applications and services (Gruber et al. 2005; see also McAffer et al. 2010a), and later as a runtime for servers (McAffer et al. 2010b).

7.3 Extant scholarly literature on Eclipse

An important early step in this research was a rigorous and systematic survey of published scholarly research examining the Eclipse field setting. This proceeded in three steps. First, the author performed a keyword search (keywords: "Eclipse" AND "open source") in four database search engines (title, abstract and keywords fields of Business Source Complete, ACM Digital Library, IEEE Xplore, and Google Scholar) to identify the list of candidate sources (Table 23). Additional manual filtering was performed on two of the search queries (ACM Digital Library and Google Scholar) which provided a large number of low-relevance results. Initial database searchers were performed on February 18 2008 and results were revised on May 30 2010. Second, the candidate articles were manually examined for relevance, seeking in particular empirical studies of the Eclipse ecosystem, governance of the Eclipse Foundation, and the organization of Eclipse projects. Third, the small group of salient articles[73] were organized into four streams (Table 24).

[73] Additional salient articles published after February 2008 (when the initial keyword searchers were conducted) were added as they were discovered. The author repeated the method on May 30 2010 and added six articles. Sources added after the original survey include six journal articles – West & O'Mahony (2008), Amoui et al. (2009), Goth (2009), Rahman et al. (2009), Shatnawi (2010) and Zhang & Kim (2010) – and Wagstrom (2009), a Ph.D. dissertation from Carnegie Mellon University.

Table 23: Results of keyword search on ("Eclipse" AND "open source")

Database	Number of hits on keywords		Notes
	February 2008	May 2010	
Business Source Compete (EBSCO Industries Inc.)	13	20	
ACM Digital Library	689	1306	Manually examined the first 300 hits returned, sorted by relevance; no salient articles were identified in hits 201-300, so hits 301-689 were not examined in detail.
IEEE Xplore	28	88	
Google Scholar	~6500	~20,400	Manually examined the first 100 hits, sorted by relevance; no salient articles were identified in hits 51-100, so subsequent hits were not examined in detail.

Table 24: Scholarly publications about the Eclipse field setting

Stream	Highlights	References
Eclipse platform, products, adoption, and utilization	Eclipse is both an open source development community and a commercial ecosystem. The Eclipse platform is an extensible framework for building, deploying, and managing software. The *Eclipse IDE for Java Developers* is the dominant IDE for Java development.	Goth (2009) Bi & Beidler (2007) Frost (2007) Yang & Jiang (2007) Murphy et al. (2006) Geer (2005) Goth (2005) Gruber et al. (2005) des Rivieres & Wiegand (2004)
The Eclipse community and Eclipse governance (empirical studies)	The Eclipse Foundation employs a structured *incubation process* to nurture newborn projects and communities.	Wagstrom (2009) West & O'Mahony (2008) Dueñas et al. (2007) Kidane & Gloor (2007)
Eclipse source code (empirical studies)	Several studies have mined and analyzed Eclipse project archives to study bug reports, change history, and comments.	Shatnawi (2010) Zhang & Kim (2010) Amoui et al. (2009) Rahman et al. (2009) Bettenburg et al. (2007) Zimmermann et al. (2007) Anvik et al. (2006) Ying et al. (2005; 2004)
Strategy of Eclipse Foundation Members (empirical studies, essays, conceptual papers)	IBM's embrace of open source is a pragmatic and sound competitive strategy.	Samuelson (2006) Capek (2005) West (2003)

The first of these four streams describes and critiques the artifacts of the Eclipse platform and outputs of the Eclipse ecosystem, including frameworks, tools, plug-ins and add-ons, open source and commercial products built on the Eclipse platform, and the adoption and utilization of these products. According to the articles in this steam, Eclipse is an open source development community situated within a vibrant commercial ecosystem (des Rivieres & Wiegand, 2004), and the Eclipse platform is an extensible framework for building, deploying, and managing software (Bi & Beidler, 2007; Yang & Jiang, 2007; Goth, 2009). The Eclipse IDE for Java Developers is the best-known Eclipse product, and it is the dominant IDE for software developed in the Java programming language (Geer, 2005; Murphy et al. 2006).

The second stream is comprised of empirical studies examining the organization and governance of the Eclipse community. Dueñas et al. (2007) describes and compares the structured incubation processes of the Apache Software Foundation (ASF) and the Eclipse Foundation, concluding that both are strong communities that "run fewer but bigger projects and deal with risk more effectively" (p. 91) than the SourceForge open source community, with incubation processes that are "intermediate between the cathedral and bazaar paradigms because their processes' degree of formalization is higher" (p. 91). Kidane & Gloor (2007) examine the correlation between group communication characteristics and productivity attributes on a sample of thirty-three Eclipse projects within three different top-level projects, employing software tools to automatically extract data from publicly available on-line source including Eclipse mailing list archives and the Eclipse Bugzilla database. West & O'Mahony (2008) include Eclipse in their sample of twelve open source projects initiated by corporate sponsors. With respect to the twelve communities examined, Eclipse was the most "open" (with regards to eleven community design parameters examined by the authors), it had community participation comparable to the earliest and most successful open source communities (including Apache, GNOME, and Debian), and it was one of only two communities to successfully transition from a sponsored community (controlled by IBM) to an autonomous community; the only other community in the sample to do so was Mozilla. Wagstrom (2009), an unpublished Ph.D. dissertation on interactions within software engineering communities that include multiple commercial firms, includes two related studies that include the Eclipse field setting. Through interviews with Eclipse Foundation staff and representatives from fifteen companies, the first Wagstrom (2009) study concludes that the structure and governance of the Eclipse community "drives values for the member companies" (p. 37) in five ways: (1) the "non-market nature of the [Eclipse] Foundation" (p. 37) which does not actively monetize Eclipse software, (2) the introduction of a standard set of processes across all projects so that "no firm has an undue advantage because of the formalities of the process" (p. 38), (3) the value of the Eclipse brand and joint marketing, (4) an organization structure that "provides a sense that everyone is playing by the same rules and acts as an equalizer for small firms" (p. 43), and (5) providing a platform for innovation by companies. The second Wagstrom (2009) study analyzes archival data extracted from Eclipse source code repositories and IP management logs to examine firm collaboration by computing metrics on projects that a firm is involved in, the firms involved in a project, and the distribution of contributions.

"More than half of the firms are involved in two or fewer subprojects, providing a testament to the degree that [the] Eclipse ecosystem is structured in such a way that firms can focus primarily on areas of expertise" (p. 60) and "make money specializing in a small component" (p. 84) Wagstrom (2009) does not explicitly define the terms "community" or "ecosystem," and appears to use the two terms synonymously throughout the study.

The third stream is comprised of empirical software engineering studies performing quantitative analysis of Eclipse design artifacts. Typically, these studies acquired data through automated data mining of publicly available project archives to study bug reports (Anvik et al. 2006; Bettenburg et al. 2007; Zimmermann et al. 2007; Rahman et al. 2009; Shatnawi, 2010; Zhang & Kim, 2010), change history (Ying et al. 2004; Amoui et al. 2009; Shatnawi, 2010), and source code comments (Ying et al. 2005). The emphasis of this stream is software engineering and development practices.

The fourth stream examines the business strategies of Eclipse Foundation member companies. Each of the three articles – West (2003), Capek et al. (2005) and Samuelson (2006) – examine IBM's open source strategy, which is salient to this research because of IBM's founding role and ongoing leadership role and participation in the Eclipse community and ecosystem.

West (2003) examines the hybrid strategies of Apple, IBM, and Sun Microsystems between 1995 and 2002, during which each company sought competitive advantage by combining open source and proprietary platform strategies in different ways. Regarding the IBM strategy of open source investment, West writes (p. 1272):

> *As he announced IBM's 2001 plans to invest US$1 billion in Linux, the then CEO, Louis Gerstner, predicted an end to the era of proprietary platforms that his company had spawned... While Gerstner might secretly prefer a return to IBM-controlled proprietary industry, the reality was that he found that an open source world – where hardware vendors and customers all had full control over crucial system software – was preferable to a proprietary industry controlled by Microsoft.*

West found that a three-stage model of proprietary platform evolution described the evolving hybrid strategies of all three companies:

(1) Complete proprietary platform solutions that are difficult to imitate and provide better margins, require vertical integration, and may be available only to one or two market leaders;
(2) Adoption of open standards shared with one or more competitors for technical and economic reasons;
(3) Open source.

The motivations for the companies to adopt open standards included the following:

(1) Market share lower than the minimum efficient scale necessary to support proprietary R&D;
(2) Not enough market power to resist buyer demands for open standards;
(3) "Tipping" of the standards contest in favor of the open standard, making it infeasible to establish (or maintain) a proprietary standard;
(4) A decision to accept commoditization of the particular architectural layer and shift competitive advantage to another layer.

As the companies moved to open standards, all three competed on implementations by adopting functional strategies for differentiation, including marketing, customer service, product design, engineering efficiency and leveraging previously established brand name reputations. The transition to open source was driven by similar factors. "Open source, however, eliminates the ability of vendors to compete based on implementations since the details of an implementation are visible to all" (p. 1279).

Capek et al. (2005), writing in the *IBM Systems Journal*, recount the history of IBM's open source investment and strategy, including the "key open source focus areas" of Linux, Apache, Eclipse, and the Globus Alliance Project (for grid computing). Open source software is "essentially an alternative business model which provides types of flexibility, opportunity, and benefits different than those provided by the conventional model. IBM was among the earliest of the major computer companies to embrace open-source software and was probably the first to realize that doing so could be consistent with our business goals" (p. 239). The authors summarize IBM's perspective (p. 254):

> *IBM views open source as a tool or technique to be used, where it makes sense to do so, to enhance our business and that of our customers. We strive to do this in a way that makes significant contributions to open source communities and projects and are often able to do that. We acknowledge that we benefit from the open-source efforts of others, but are, on balance, a net contributor.*

Samuelson (2006) provides another perspective on IBM's "pragmatic embrace of open source." IBM was once a vigorous advocate of very strong intellectual property (IP) rights for computer software. Three complementary explanations, each with some explanatory power, can account for IBM's embrace of open source.

(1) An anti-Microsoft strategy. Linux was the first OS with the potential to challenge Microsoft's dominant position.

> *Spending $100 million a year on Linux development is a bargain for IBM by comparison with developing a new OS from scratch. Investing in Linux allows IBM to be independent from Microsoft's licensing terms and willingness to reveal (or not) interface specifications for its platforms. IBM's support of Linux increases the chances that Linux will succeed in its competition with Microsoft in OS markets. (p. 22)*

(2) A consequence of changing business models in the software industry.

(3) Adoption of an open innovation strategy, the benefits of which include sharing cost and effort among many contributors, avoiding the substantial costs of launching a new product, bring customers into the development team, and speeding the pace of innovation.

In summary, West (2003), Capek et al. (2005), and Samuelson (2006) all argue that the motives underlying IBM's open source strategy are driven by business interests.

At least four other literatures complement the scholarly research literature on Eclipse. First, there is a large technical practitioner on using Eclipse, including an Eclipse book series from Addison-Wesley and a vast assortment of community-developed white papers, case studies, and blogs. Second, there is a large trade press literature about companies and products, including three Eclipse supplements to the *Software Development Times* (2004, 2006, 2007). Third, there are a series of research reports commissioned by the Eclipse Foundation, including the *Eclipse Community Survey* (2007, 2009, 2010, 2011). Fourth, the Eclipse Foundation staff frequently speak, publish, and blog about their own experiences and perspectives on open source communities and ecosystems (e.g., Milinkovich, 2006, 2007, 2008a, 2008b, 2008c, 2010; Campbell, 2007; Smith & Milinkovich, 2007; Skerrett, 2008a, 2008b, 2009, 2011; Beaton, 2008; Skerrett & Beaton, 2010), providing useful insights into participant beliefs and expectations.

7.4 The Eclipse platform

The *Bylaws of the Eclipse Foundation* (2003; 2008, article 1.1) identify three purposes of the Eclipse Foundation, the first of which is "to advance the creation, evolution, promotion, and support of the Eclipse Platform." The *Bylaws* describe the platform as follows:

> *The Eclipse technology is a vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools (the "Eclipse Platform"). Eclipse Platform tools are exemplary in that they verify the utility of the Eclipse*

> *frameworks, illustrate the appropriate use of those frameworks, and support the development and maintenance of the Eclipse Platform itself; Eclipse Platform tools are extensible in that their functionality is accessible via documented programmatic interfaces.*

In recent writing, the Eclipse Foundation Executive Director describes the Eclipse platform more broadly, as the combined base of technologies, architectures, designs and assets used to build market offers, components, products and services, legal and licensing frameworks, and processes which anchor economic community (Milinkovich, 2008a). This expanded view encompasses the Eclipse open source software tools and frameworks, and also the complementary assets (Teece, 1986; section 3.1) provided through the Eclipse Foundation, including operating rules that guide and enable action[74].

Eclipse software applications are comprised of modular *plug-ins* and a small *runtime kernel* (Figure 7). Each plug-in is a self-contained module with well-defined interfaces, and may include *extension points* where other plug-ins may be provisioned through application programming interfaces (APIs). Plug-ins can be removed, new plug-ins can be added, and existing plug-ins can be swapped out for other plug-ins. Gamma & Beck (2003, p. 4) use the analogy of an electrical powerstrip with multiple outlets:

> *Eclipse is a collection of places-to-plug-things-in (extension points) and things-plugged-in (extensions). The powerstrip is a kind of extension point. Multiple extensions (in this case, power plugs) can plug into it, and although the extensions are different shapes and have different purposes, they all must share a common interface.*

[74] Software in this context means intellectual property in electronic form, including source and binary code, documentation, courseware, reports and whitepapers (*Eclipse Standard Top-level Charter*, version 1.1; *Intellectual Property Policy*, 2008).

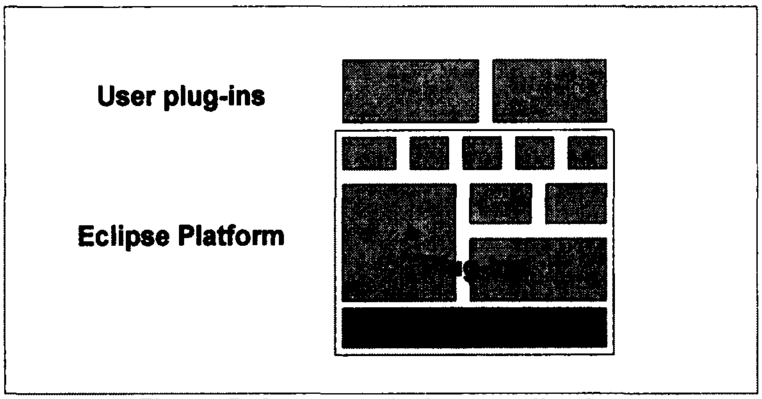


Figure 7: A generic Eclipse application

Presentations from Eclipse Foundation staff describe the Eclipse modular architecture as a deliberately crafted "architecture of participation" (e.g., Milinkovich, 2008a, 2008c; Skerrett, 2008a) in the sense of O'Reilly (2005a) and the related scholarly literature previously reviewed in chapter 4[75]. The specific arguments raised to support this claim include the following:

- The open APIs and commercial friendly licensing of the Eclipse platform create low barriers to entry for new developers.
- The ease of integration and extensibility enable incremental innovation built on past work.
- All Eclipse plug-ins are developed according to the same design rules available to everyone: "New plug-ins are first class citizens" (Milinkovich, 2008a, p. 20).
- Competition takes place on implementations where users decide winners.

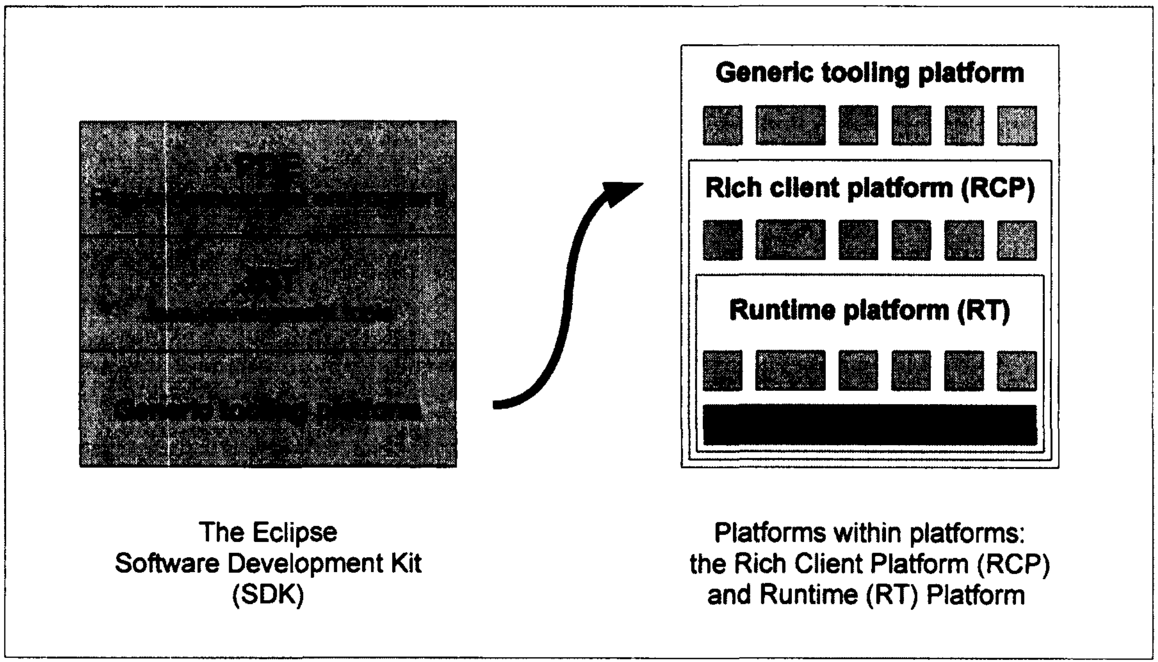
These points complement the traditional technical argument that strong modularity enables new work to proceed in parallel with few external dependencies.

Eclipse tools are tested configurations of plug-ins that accomplish specific tasks for users. The best-known and most widely used Eclipse tool is the Eclipse Java IDE used by millions of software developers to create, test, and maintain software written in

[75] Tim O'Reilly's opening keynote presentation at EclipseCon 2005 (O'Reilly, 2005b) helped disseminate his ideas on architecture of participation widely within the Eclipse community.

the Java programming language. Eclipse plug-ins are themselves written in Java, and developed using Eclipse tools; software developers creating Eclipse plug-ins are both *consumers and producers* of Eclipse software (Gamma & Beck, 2003; Carlson, 2005).

Eclipse tools are assembled from and built on *Eclipse frameworks*. According to Eclipse Executive Director Mike Milinkovich, "People inside Eclipse actually think of it primarily as for building extensible frameworks. And then we build a tool on top of it to show people how you can use those frameworks" (Anderson, 2004). As an example, Figure 8 depicts the *Eclipse Software Development Kit (SDK)* – an Eclipse software tool configured for building Eclipse plug-ins – alongside an expanded view of the nested platforms within the Eclipse generic tooling platform.



The Eclipse
Software Development Kit
(SDK)

Platforms within platforms:
the Rich Client Platform (RCP)
and Runtime (RT) Platform

Figure 8: The Eclipse Software Development Kit (SDK) and generic tooling platform

One view of of the Eclipse SDK is a stack of three layers (Gamma & Beck, 2003): (1) a *generic tooling platform* defining the common infrastructure for Eclipse tools, (2) the *Java Development Tools (JDT)* adding the capability of a full-featured Java IDE, and (3) the *plug-in development environment (PDE)* extending JDT with specialized support for developing Eclipse plug-ins. Each layer is a collection of configurable Eclipse plug-ins. A developer requiring a Java IDE but not the capability to create, package and deploy their own Eclipse plug-ins could remove the PDE layer, or acquire a version of the Eclipse Java IDE without the PDE layer installed. A developer requiring an IDE for a different computer programming language could swap out JDT for a different language layer – for example, *CDT (C and C++ Development Tooling)* for a C and C++ IDE, or *DLTK (the Dynamic Languages Toolkit)* for an IDE supporting the Ruby, TCL, JavaScript, and Python programming languages. A developer coding in both Java and C++ could have both JDT and CDT installed simultaneously.

The original Eclipse platform whitepaper, first distributed by the OTI division of IBM in 2001 prior to the launch of the Eclipse Consortium, described Eclipse principally as a *tools platform* (Object Technology International, 2001, p. 3):

> *to provide tool providers with mechanisms to use, and rules to follow, that lead to seamlessly-integrated tools. These mechanisms are exposed via well-defined API interfaces, classes, and methods. The Platform also provides useful building blocks and frameworks that facilitate developing new tools.*

In later years, the Eclipse generic tooling platform was successively refactored, repartitioned, and repackaged to address a broader range of software applications and potential adopters.

Contained within the generic tooling platform, beneath the tooling plug-ins, is the Eclipse *Rich Client Platform (RCP)*. Jeff Norris, supervisor of the Planning Software Systems Group at Jet Propulsion Laboratory, a consumer of RCP for space exploration and control applications developed for NASA, writes (McAffer et al. 2010a, p. xxi):

> *I've seen RCP applications for controlling nuclear reactors, scheduling trains, trading stocks, designing data centres, managing inventory, fighting terrorism, analyzing proteins, monitoring fishing boats, sharing files, and editing every programming language that I've ever heard of.... But what's more surprising than the diversity of RCP applications is everything that our applications have in common. For example, [America Power Conversion, a company providing infrastructure for industrial electronic equipment] uses the same graphical editor framework to arrange server racks in their data center design program that my team uses to manipulate Mars images in our rover operations program. My team built our spacecraft command editor with the same basic components used in most of the Eclipse programming tools. These commonalities allow us to combine our resources, learn from each other, and ultimately deliver better products to our customers.*

The RCP plug-ins provide functionality for client applications running on desktop computers and other client-side devices. McAffer et al. (2010a, p. 5) writes:

> *The Eclipse RCP is essentially the middleware function that you don't want to write because it is a means, not an end, for your domain. It includes facilities such as a flexible UI [user interface] paradigm, scalable UIs, extensible applications, Help support, context-sensitive Help, network updates, error handling, and much more.*

Contained within the RCP, beneath the RCP plug-ins, is the Eclipse Runtime (RT) platform. RT, consisting only of the Eclipse Equinox Runtime kernel and plug-ins providing various runtime services, is a framework for building high-performance application servers and other server-side software (McAffer et al. 2010b). Applications built on the Eclipse runtime platform include IBM's WebSphere Application Server and the open source Jetty application server. These are “headless” applications that do not require the graphical user interface and other end-user capabilities provided by the RCP.

According to Eclipse Foundation staff, growth in Eclipse membership has historically followed the creation and maturation of Eclipse projects in several overlapping waves (*Proposed Membership Changes*, 2008) . In the earliest days of the Eclipse Consortium, a wave of companies producing tools for Java software development joined the Eclipse Foundation to participate in Eclipse Java tools projects such as JDT and projects that built plug-ins on top of JDT. Later, a second wave of embedded tool companies joined to participate in CDT and other projects for mobile and embedded development that built on the generic tooling platform. The third wave of growth was participation in framework projects, the Rich Client Platform, and runtime projects. The next wave of future growth is expected to be enterprises joining to participate in Eclipse "industry verticals" such as mobile devices, health care, and the automotive sector. An example is the recent proposal to establish an Eclipse Automotive Industry Working Group to define, implement, and promote a common Equinox-based platform for automotive design time and runtime components on a service-oriented architecture (*Eclipse Automotive Working Group Charter*, 2011; eclipse.org-membership-at-large mailing list, Milinkovich, February 7 2011).

Eclipse software assets are distributed under the *Eclipse Public License (EPL).* The EPL is both a *free software license* approved by the *Free Software Foundation* as compliant with the *Free Software Definition* (Stallman, 1996a), and an *open source software license* approved by the *Open Source Initiative* as complaint with the *Open Source Definition.* It is deliberately crafted to be a business-friendly license enabling and promoting commercial adoption while fostering community development (Campbell, 2009). The recipient of EPL software can use, modify, copy and redistribute the original software and modified versions in both source code and object code form, and in some cases, the recipient may be required to release the source code for their changes under the same license. In open source software parlance, the EPL is termed a "weak copyleft license." This is in contrast to a "strong copyleft license," such as the GNU General Public License (GPL), that requires source code for all changes that are distributed to be released under the same license (Free Software Foundation, 1991). Specifically, a subsequent contributor to an EPL asset must distribute under EPL "i) changes to the Program, and ii) additions to the Program" unless those additions "(i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program." The exclusion of separate modules from the scope of the EPL, along with the design and modular architecture of the Eclipse platform, permits developers to often employ unmodified Eclipse software assets in proprietary projects without requiring EPL distribution of their own software modules. The EPL includes disclaimers of liability and warranty to protect contributors from litigation by users, and patent license provisions to protect users from litigation by contributors. Anyone distributing an EPL work agrees to grant recipients a license to any patents that they might hold that cover the modifications they have made.

Contributors to Eclipse software retain copyright ownership of their contributions (Skerrett, 2011). For a contribution to be accepted, it must normally be submitted through the Eclipse project infrastructure and thus governed by the *Eclipse.org Terms of Use* (2008), which include agreement to license the contribution under the EPL.

7.5 The Eclipse community

The second purpose of the Eclipse Foundation, as defined in the *Bylaws*, is to "cultivate ... an open source community" (*Bylaws*, 2003, 2008, article 1.1). The *Bylaws* do not elaborate further, but over time, more explanation of community and requirements for building community have been added to the *Eclipse Development Process* (EDP). The *Eclipse developer community* – a specific subset of the broader Eclipse community – is examined in its own subsection immediately following the body of this section.

An early and influential book by Gamma & Beck (2003) described the Eclipse community as a pyramid with six levels: "Moving up the pyramid requires investment and will be attempted by fewer people" (p. 1). The community constituents at each level of the Gamma & Beck pyramid are *individuals* interacting with Eclipse software. The wide base at the bottom is *users* of Eclipse software ("Currently this is restricted to programmers, but there is no reason in principle why Eclipse couldn't be used to structure other computing work", p. 1). The second tier is *configurers* – users who customize their experience by making changes envisioned by the Eclipse programmers. The third tier is *extenders* – programmers who create plug-ins with new functionality that change Eclipse in ways not envisioned by the Eclipse programmers. The fourth tier is *publishers* – programmers who create plug-ins and share them with others. The firth tier is *enablers* – programmers who create plug-ins with extension points for others to extend, and publish the plug-ins and extension points. Finally, the sixth tier is *committers* – programmers who have gained the trust of the existing community of committers and receive privileges to change the Eclipse source code[76]. These levels can also map to a circle, with enabler pointing back to user (p. 2):

> *What makes this circle interesting is the final arrow, from Enabler back to User. In Eclipse, you don't just invest more and more and receive more and more. When you take the step to become an enabler with your own extension points, you create for yourself the opportunity to be nourished by the work of others. Sometime later, someone may extend your contribution in ways you find useful. By creating and sharing your own extension points you can get the benefit of their work without further effort on your part.*

Although the community model most widely used by the Eclipse Foundation differs from this, the Gamma & Beck pyramid and circle provide useful insights into how the original Eclipse platform architects envisioned the Eclipse platform and the first Eclipse SDK. Erich Gamma, the first author of Gamma & Beck (2003), was the founding project lead of the Eclipse JDT project, an architect of the Eclipse platform when it was an IBM/OTI internal project, and one of the authors of the original Eclipse technical white paper (OTI, 2001). Gamma argues that the architects of Eclipse created the Eclipse plug-in architecture with tiered participation in mind, deliberately enabling upward movement in small incremental steps from user to programmer to committer (see also Gamma, 2008)[77].

The *Eclipse Development Process* recognizes three inter-related communities around each Eclipse project[78]: (1) a *user community* of individuals and organizations that use the software that the project creates, (2) an *adopter community* of individuals and organizations that extend Eclipse software with their own plug-ins, and (3) a *developer*

[76] Gamma & Beck's pyramid resembles in some ways the *onion model* of community-developed open source software projects (Crowston & Howison, 2003; Herraiz et al. 2006; Antikainen et al. 2007; Kilamo et al. 2010), explained and interpreted through a different metaphor.
[77] Prior experience with the Smalltalk programming language (Thomas, 1995) and development of Smalltalk tools at Object Technology International (OTI) (acquired by IBM in 1996, but operated as a distinct entity until 2003) are cited as the origin of these ideas (Gamma & Beck, 2003; Gamma, 2008).
[78] Chapter 8 closely examines and categorizes the operating rules comprising the EDP.

community of *committers and (non-committer) contributors* that maintain and extend the project's software. Rather than depict these three communities as hierarchical – for example, as three concentric layers of an onion as typical in the open source software literature (Crowston & Howison, 2003; Kilamo et al. 2010), or as three steps of a pyramid like Gamma & Beck (2003) – the Eclipse Foundation affords them equal status by depicting them as interconnected (Milinkovich, 2008a, 2008b, 2008c; Skerrett, 2011) and intersecting (Beaton blog, July 28 2010). All Eclipse projects are required to undertake and report annually on activities that develop each of these three project communities: "The Eclipse community considers the absence of any one or more of these communities as proof that the Project is not sufficiently open, transparent, and inviting, and/or that it has emphasized tools at the expense of extensible frameworks or vice versa" (*Eclipse Development Process*, version 2.5, section 2.3, p. 3). Within Eclipse practitioner parlance, the *Eclipse user community*, the *Eclipse adopter community*, and the *Eclipse developer community* are each said to comprise the respective communities for all Eclipse projects, and the *Eclipse community* (with no other adjectives) broadly encompasses all three Eclipse communities. Within each community, there are said to be many *niche communities* – smaller communities anchored around different technologies and clusters of related projects, where community members of the niche interact more strongly with one another than with community members of other niches.

Ian Skerrett, the Eclipse Director of Marketing, recommends five best practices for building a successful open source technical community (Skerrett, 2011; Skerrett blog, November 18 2010). (1) Engage a wider community by using a permissive or *weak* *copyleft license* such as EPL rather than a strong copyleft license like GPL. (2) Earn the trust of the community by *not requiring copyright assignment*; copyright to contributions should be retained by the originating contributor. (3) *Be truly open* by using public issue trackers, public code repositories, and public build systems, publish up-to-date project plans, and discuss technical issues in the open. (4) Have a *clear policy on trademarks* – how the trademarks can and cannot be used, allowing all organizations the same rights and privileges. (5) Implement a *vendor-neutral governance structure* with well-defined rules for making decisions. More generally, Skerrett (2008a) explains that *communities are about conversations*, and "the conversation starts with code or a compelling solution to a problem and shorter product cycles." Skerrett (2008a) quotes from Levine et al. (1999), arguing that the logic of *The Cluetrain Manifesto* applies to communities as to markets:

> *1. Markets are conversations. 2. Markets consist of human beings, not demographic sectors. 3. Conversations among human beings sound human. They are conducted in a human voice. 9. These networked conversations are enabling powerful new forms of social organization and knowledge exchange to emerge. 62. Markets do not want to talk to flacks and hucksters. They want to participate in the conversations going on behind the corporate firewall.*

This approach to community as conversation is perhaps most evident in the way Eclipse Foundation staff and Eclipse community leaders engage in frequent public dialogue directly with the community through various interactive technology channels. The *Planet Eclipse* blog aggregator tracks nearly 400 blogs about Eclipse, including blogs from all director-level Eclipse Foundation staff.

7.5.1 The Eclipse developer community

The Eclipse developer community of contributors and committers who directly maintain and extend Eclipse software is of central interest to this research. Wayne Beaton of the Eclipse Foundation staff explains (Beaton blog, July 28, 2010):

> *These are the individuals who provide patches and/or contribute new functionality to the various Eclipse projects. Contributors tend to participate directly in an Eclipse project by providing code, ideas, answers to questions in the forums, and more. Committers are a subset of the contributors with write access to the resources maintained by a project (committer access is provided on a project-by-project basis). The idea is that over time, a contributor is invited to become a committer and elected into that position based on credibility established over a period of time. At last count we had almost a thousand committers and thousands of contributors (more than 11,000 individuals have contributed at least one patch to an Eclipse project).*

"A thriving, diverse and active community of developers is the key component of any Eclipse Project" (EDP 2.5, section 2.3, p. 3), however, all three communities are valued and deliberately nurtured: "A strong community of users and adopters is essential for a strong contributor/committer community" (Skerrett, 2011, p. 12).

The Eclipse developer community is a project-based *meritocracy* governed by the *Eclipse Development Process* and the applicable project charter document[79]: "The breadth of a Committer's influence corresponds to the breadth of their contribution" (EDP 2.5, section 4.7, p. 8). In the Eclipse developer community meritocracy, it is individuals – not the companies employing those individuals – which have merit and status. Individuals who have earned the privilege to be elected as committers to a project may vote on project matters, participate in the nomination and election of new committers, and write to the project's resources. Commit privileges and other community roles attach to an individual rather than an individual's employer. For example, an individual's rights

[79]The roles and operating rules within the Eclipse developer community are examined in detail in chapter 8.

and responsibilities in the developer community do not change if that individual changes employers or other organizational affiliations; likewise, a contributor is said to receive no special community status from any particular organizational affiliation.

By all accounts, *most Eclipse committers are paid to contribute to Eclipse* – either as part of their job function with an employer, a contractor performing work for customers, or some other arrangement (Milinkovich, 2008b; Skerrett, 2008b; Wagstrom, 2009, p. 50). These resources are provided by organizations in the Eclipse ecosystem.

7.6 The Eclipse ecosystem

The third purpose of the Eclipse Foundation, as defined in the *Bylaws*, is to cultivate "an ecosystem of complementary products, capabilities, and services." In 2003, when this statement of purpose was first drafted, much of the research literature reviewed in previous sections of this document had not yet been written. According to Executive Director Mike Milinkovich, the Eclipse Foundation's early success in creating and nurturing a "a multi-billion dollar, worldwide ecosystem that spans hundreds of companies and thousands of products" was "accomplished in blissful ignorance of ecosystem theory" (Milinkovich, 2008a).

A close reading of Eclipse press releases, and presentations, articles, and blogs by Eclipse Foundation staff, reveals an evolving sophistication of ecosystem thinking that enfolded new ideas over time as they became available. According to Milinkovich (2008a), this happened is five stages, arriving at a current view of the Eclipse ecosystem as a system of coordinated economic production consistent with Moore (2006).

When IBM published the Eclipse SDK 1.0 source code and launched the Eclipse

Consortium in 2001, the plug-in architecture and permissive CPL license permitted anyone to create commercial offers and extend the platform in novel ways. However, IBM retained a privileged position as the organization at the centre of the system. To attract broad and deep participation by others, IBM worked towards placing a not-for-profit company at the centre of the system, resulting in the launch of the Eclipse Foundation in 2004 with the dual role of cultivating an open source community and commercial ecosystem. The second stage of Eclipse ecosystem thinking introduced the platforms literature (section 3.3; e.g., Gawer & Cusumano, 2002; Cusumano & Gawer, 2002) and O'Reilly's ideas on the architecture of participation (chapter 4; e.g., O'Reilly, 2001, 2004) to conceive of Eclipse as a participative platform. Although scholarly literature at the time focused exclusively on proprietary platforms such as those of Microsoft and Intel, it clearly and usefully articulated the central importance of clean and stable interfaces and of predictability as determinates of adoption. The third stage added insights from Iansiti & Levien's (2004a, 2004b) work on keystone advantage, diversity of ecological niches, and ecosystem health, the ecological analogy of the business ecosystem as a co-evolving network, and the role of the keystone as increasing network value by fostering linkages and recruitment for niche creation (sections 2.3 and 2.5). The fourth stage introduced Chesbrough's (2003a, 2003b) distinction between value creation and value capture, clearly articulating the benefits of business-friendly licensing and assurances of unencumbered intellectual property for adopting firms (section 3.2). The fifth stage and current view brought in Moore's (2006) formulation of the business ecosystem as a form of economic organization that coordinates continuous innovation across organizations and markets (sections 2.4 and 2.7). Each stage introduced a sharper vocabulary and new conceptual frameworks to more clearly explain and understand what, for the most part, Eclipse was already doing. According to Milinkovich (2008a, p. 39), "Eclipse has stumbled upon the best existing model we've seen for fostering corporate collaboration in innovation." Smith & Milinkovich (2007, p. 7) write: "This really is the best of both worlds: the openness, transparency and meritocracy of open source with the resources and commitment of corporations both large and small."

Table 25 illustrates this development with excerpts from archival documents.

Table 25: The changing meaning of "ecosystem"

Eclipse at IBM (~1998-2001) Origins	Skip McGaughey (IBM Software Group; founding chairperson of the Eclipse board of stewards) about creating the Eclipse Consortium (quoted by O'Mahony et al. 2005a): *We wanted to create an Open Source project that was different than any other Open Source project that had ever been done before, in that we wanted to have a commercial arm. So, we created this dual-edged or bi-polar organization that on one side would play by Open Source rules of engagement to develop the technology and that on the other side was the eco-system side, or the commercialization of the technology.*

Eclipse Consortium (Founded 2001)
Divergence
	Eclipse.org Consortium press release (December 16 2002):
	Eclipse is an open source eco-system of royalty-free technology and a universal platform for tools integration.
	Eclipse.org Consortium press release (April 15 2003):
	A worldwide "eco-system" of tools providers and consumers is forming around the Eclipse Platform, an open universal tools integration platform for multi-language, multi-platform and multi-vendor choice.
	Eclipse.org Consortium press release (November 25 2003):
	EclipseCon brings together the Eclipse ecosystem: developers, software architects, technical managers, systems integrators, thought leaders, and other software development tools producers and consumers using or interested in learning about Eclipse technology.

Eclipse Foundation (Founded 2004)
Refinement and restatement
	Bylaws of the Eclipse Foundation (2003, article 1.1):
	[The Eclipse Foundation purpose is] to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.
	Milinkovich blog (April 19 2005):
	The term ecosystem is sort of all-encompassing. It includes our committers, users, developers, book writers, service providers, product sales guys. You name it.

Today Convergence	Milinkovich (2008a, p. 42): The Eclipse ecosystem is a business ecosystem as defined by Moore (2006): "an intentional community of economic actors whose individual business activities share in some large measure the fate of the whole community."

In summary, the "Eclipse ecosystem" concept has shown great versatility, evolving over time to take on new meaning for practitioners. The over-all pattern is one of divergence from a broad notion of commercializing open source technology, followed by refinement, restatement, and enfolding other ideas, then synthesis and convergence around Moore's (2006) concept of a business ecosystem of economic actors. From these observations, sections 7.8 and 11.1 develop an operationalized "Eclipse ecosystem" construct and situate it within the ecosystem literature and the frameworks of chapter 2.

Astley & Zammuto's (1992) previously argued that *linguistic ambiguity* is a common feature of concepts that become widely adopted in management practice. It could be that the flexibility to reinterpret the Foundation purpose and to adapt to the shifting needs of the day may be a factor in the widespread adoption of the ecosystem concept by practitioners and its continuing ubiquity in Eclipse discourse.

A related "technology ecosystem" metaphor – of the "componentized software ecosystem" of Eclipse plug-ins (McAffer et al. 2010a, p. 5) or the "ecosystem of commercial and open source add-ons to the Eclipse platform" (Milinkovich blog, May 13 2010) – is common in some technical writing and informal community conversation. In the same way that a business ecosystem metaphor brings into sharp focus the ecological properties of business organizations (section 2.3), a technology ecosystem metaphor focuses on the interconnectedness, co-dependence, and co-evolution of the Eclipse software components created by the economic actors of the Eclipse business ecosystem[80].

[80] See also the previous discussion in section 2.6 on other ecosystem concepts in the management and engineering literatures. Subsection 11.1.1 revisits this theme to examine how these two ecological metaphors – one for the Eclipse *producers of innovation* and one for the Eclipse *products of innovation* – are deeply interconnected, and proposes that both can be reconciled with the business ecosystem construct.

7.7 Practitioner explanations of the Eclipse field setting

In archival documents, community conversations, and practitioner interviews, four partial explanations connect the Eclipse Foundation, community, platform and ecosystem. Each explanation was recurring – evidenced by multiple instances at different points in time. Citations provided are to evidence in the form of Eclipse documents, presentations by Eclipse Foundation staff, and archived conversations. Collectively, the similarities of these four explanations provide insight into practitioner beliefs about how the Eclipse ecosystem and surrounding systems operate.

The first connection is embedded in prescriptive advice from Eclipse Foundation staff to member companies and potential new members (Milinkovich, 2008c, p. 33)[81]:

> *Define* very *precisely what your competitive differentiators are for your customers or you're going out of business. Focus all possible energies there, and acquire everything else from open source software, or help build it in open source software. Build [the non-differentiating part of the offer] in and with open source, even if that means working with your direct competitors.*

By this logic, Eclipse software provides the non-differentiating open source platform for a wide assortment of commercial offers. Companies in the Eclipse ecosystem identify an attractive business niche, obtain and protect differentiating technology or capability, create an attractive offer by combining differentiating technology or capability with the Eclipse platform and other open source software, and co-evolve the platform in collaboration with other actors in the ecosystem. The underlying logic is "collaborate on the platform" (to reduce time to market, share costs, increase adoption, increase quality,

[81] There are several variants of this advice featuring different labels for the differentiating and non-differentiating components of the offer: value and platform (Milinkovich, 2008c), value and infrastructure (Milinkovich, 2007), and STM (stuff that matters) and STDNM (stuff that does not matter) (Milinkovich, 2006). The Eclipse distinction between differentiating and non-differentiating components of the offer echoes the *core-context analysis* recommended by Geoffrey Moore (2005).

and reduce risk) and "compete on products" (differentiating features, service, support, branding, and channels) (Milinkovich, 2006, p. 15). An example often cited is the Eclipse CDT project[82], where committers from nine different companies collaborate to produce an Eclipse open source C and C++ IDE, which is the base for more than a hundred commercial products competing in the marketplace. Some companies employing committers on CDT – such as Intel, Texas Instruments, and Freescale Semiconductor – are fierce competitors in their primary markets.

The second connection proposes a positive feedback loop of resource flows from community innovation to corporate profits and back to community innovation (Figure 9), variously called "Eclipse Economics 101" (Milinkovich, 2006, p. 11), or the "Eclipse Virtuous Cycle" (Milinkovich, 2008a, p. 14; see also Freeman-Benson blog, April 7 2009; Schaefer blog, April 7 2009). According to this explanation, committers in the developer community produce high quality open source projects, resulting in exemplary software tools adopted by large numbers of users, creating opportunities for vendors in the ecosystem to sell differentiated offers to users; vendors then re-invest some of their profits by employing committers to continue the cycle in anticipation of future benefits. The vendor offers are enabled by the Eclipse business-friendly licensing model and well-defined stable APIs.

[82] CDT is one of five case projects examined closely in chapter 8. During the study interval of June 2009 to November 2010, committers employed by nine different companies committed code to CDT (Table 39). The estimate of "well over a hundred products shipping that are based on CDT" is from Geelan (2008).

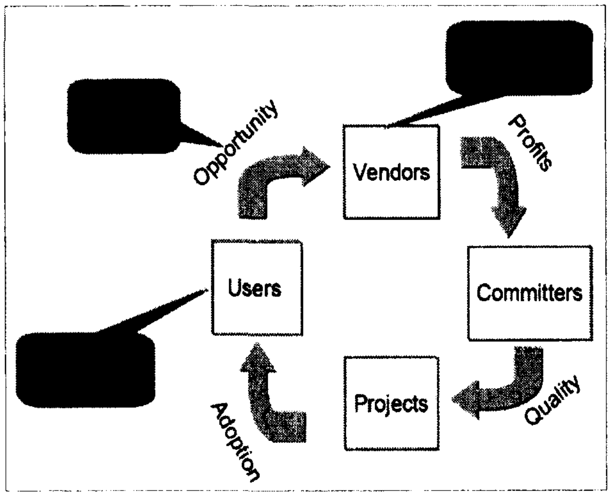


Figure 9: Cycle of profits and innovation (adapted from Milinkovich, 2008a, p. 14).

The third proposed connection is a maturity model of company involvement in open source software projects developed at Carleton University in 2006 (Carbone, 2006, 2007a, 2007b; Bailetti, 2007a, 2007b, 2009; Milinkovich, 2008a, 2008b; Wagstrom, 2009). Table 26 summarizes the five stages of company maturity in the context of Eclipse participation (adapted from Milinkovich, 2008a). According to Eclipse Foundation staff, this model is used internally within the Foundation to help members at all five stages get the most value from their participation in Eclipse. At the board of directors, it provides a framework for strategic planning, developing new programs to benefit members, and developing new programs to grow the membership. In the annual *Eclipse Community Survey* (2007, 2009, 2010, 2011), it provides an ordinal scale for measuring organizational participation. At all levels, it provides a common vocabulary for discussion of company participation. For example, at a member vote for a proposed *Bylaws* amendment in 2008, the maturity model provided a vocabulary to articulate why enterprise companies were expected to join the Eclipse Foundation if a new solutions provider membership class were to be created with dues and benefits that matched their needs (*Proposed Membership Changes,* June 24 2008): "The further along the open source maturity curve that an organization is, the more likely they would become a member. Many ISVs [independent software vendors] are in stages 2 – 4. Many enterprise users are moving from 0 – 1 into stage 2 – 3 and want our help to reach 4 – 5."

Table 26: Eclipse and the open source maturity model

Stages	1. Use	2. Contribute	3. Champion	4. Collaborate	5. Redefine
Business reasons for a company to engage with Eclipse	Explore. Reduce build costs. Decrease time to market.	Fill gaps in feature set of a company's product. Improve software quality.	Champion open source project. Steer new functionality and its evolution.	Gain positional advantage for company's product in a market segment.	Change value proposition of offer delivered to customers.
Benefits accrued by the Eclipse open source software community	Increased number of users and awareness of open source project.	More features and better quality.	Company pull for open source project. Leadership structure and governance.	Company commitment to health and advancement of open source project. New versions of open source software.	New company-driven resources establish links with other open source projects and launch new projects.

The fourth connection is an open source business logic introduced to the Eclipse community by Williams (2007), and recommended by Eclipse Foundation staff to new and potential Eclipse Foundation members[83]. According to Williams, there are three "levers" that companies can pull to drive stock price higher (p. 18): (1) increase revenue (at the same or lower cost structure), (2) reduce costs (while maintaining or increasing revenue), and (3) increase the multiplier which financial markets assign to a company's revenues and costs to determine its financial worth. Open source software – more

[83] William's EclipseCon 2007 presentation was widely and favourably discussed outside of the Eclipse community on several high-traffic blogs, including Slashdot (March 13 2007). See also Williams (2006), a previous presentation to the Eclipse Foundation board of directors.

specifically, the use and adoption of Eclipse open source software and active participation in the Eclipse ecosystem – can provide companies with all three benefits. Williams cites the example of Actuate Corporation[84]: between 2004 and 2007, Actuate stock appreciated more than stock of other business intelligence software companies. Williams attributes this to the multiple simultaneous expansions made possible by Actuate's leadership of the Eclipse BIRT project (launched in 2004 by Actuate): multiple types of revenue upside at lower cost on multiple product lines, an architecture for driving more product innovations faster than competition, credit for future products not yet developed, and the capability to win market share against competitors (p. 21):

> *Actuate found that BIRT not only generated direct revenue but accelerated the growth of their high-end proprietary product. ... In other words, your Eclipse involvement is about a lot more than just the direct revenue you're getting from Eclipse-based products & services.*

According to this logic, companies that try to use open source to solve the wrong problem, or produce community "blowback" for "anti-open source misbehaviour" (p. 16) are not rewarded by the financial markets.

Pragmatic business value is the unifying theme throughout this discussion. By the logic of these four practitioner explanations, economic actors in the Eclipse ecosystem which participate in developing open source software are neither performing a "good deed" nor an act of charity: participation is a business imperative motivated by pursuit of profits and competitive advantage over rivals, and contribution of resources to the Eclipse developer community is motivated by pragmatic business logic and the expectation of tangible business benefits.

[84] Actuate Corporation is one of eight case companies examined closely in chapter 10. BIRT, the Eclipse business intelligence and reporting tools project, is one of five case projects examined closely in chapter 8.

7.8 Synthesis and theory development

This section develops an inductive conceptual model, consistent with observations from the Eclipse top-level case, as a framework for further analysis. The institutional analysis and design (IAD) framework (Ostrom, 1990, 2005; Ostrom & Hess, 2007), surveyed in chapter 5, provides the technical vocabulary and conceptual apparatus. Resource dependence relationships (Pfeffer & Salancik, 1978) link the components together. Concepts from the business ecosystems literature (chapter 2), related management literatures (chapter 3), and the practitioner literature on participation (chapter 4) sharpen explanations and provide links to past scholarship.

An operationalized institutional view of the Eclipse ecosystem requires greater precision and specificity regarding terms, definitions, boundaries, constructs, and relationships than what was found in practitioner discourse. The research literature on managerial language (e.g., Astley & Zammuto, 1992) argues that this is neither unusual nor unexpected: because practitioners and researchers operate in different domains, each with its own own divergent history, focus of attention, terminology, and modes of discourse, requirements for precision and specificity of language are expected to differ.

The objective of this section is a theoretical abstraction that captures the main features needed to better understand the institutions of company participation, with constructs and associations that are congruent with the practitioner explanations of section 7.7 and recognizable to practitioners. Consistent with the institutional literature surveyed in chapter 5, it is not a perfect "snapshot" of institutional reality in all its complexity, nor a complete description of the micro-structure within each component.

7.8.1 Conceptual model

Following the Eclipse Foundation purpose "to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services" (*Bylaws of the Eclipse Foundation*, 2003, article 1.1), the developer community and ecosystem are treated here as separate *institutions* – the formal constraints, informal constraints, and enforcement characteristics that structure human interaction in a way "perfectly analogous to the rules of the game in a competitive team sport" (North, 1990, p. 4). The decision to model the community and ecosystem as two institutions rather than one is motivated by the institutional differences evident in the case data: the Eclipse developer community and the Eclipse business ecosystem differ with regards to salient actors, actor motivations and actor decision processes; they interact with the Eclipse platform in different ways, and they differ in the strength and mechanisms of influence exerted back on the governing foundation; most important theoretically, they differ in their emphasis of different aspects of innovation. Furthermore, the Eclipse Foundation treats the community and ecosystem as distinct entities in some contexts, for example, by creating separate and distinct roles for the Director of Community Development (with global scope) and the two Directors of Ecosystem Development (based in North America and in Europe, respectively)[85].

Table 27 compares the Eclipse developer community and the Eclipse ecosystem institutional constructs along seven salient dimensions.

[85] Section 11.4 discusses on possible alternative formulations and constructs to the model developed here.

Table 27: Comparison of developer community and business ecosystem institutions

Dimension	Institution of individual participation in the Eclipse developer community	Institution of company participation in the Eclipse ecosystem
Actors	*Individual actors*: software developers (committers and contributors) that maintain and extend the Eclipse platform software.	*Economic actors*: organizations (and some independent individuals) that develop products, technology and services that are economic complements to the Eclipse platform.
Actor motivations	Complex and heterogeneous; mix of intrinsic and extrinsic motivations.	Profit motivation; guided by market forces.
Actor decision processes	Individual decision-making.	Organizational and individual decision-making.
Interaction with Eclipse software	Eclipse software is both *input and output* of community activity. Trust and greater influence over software development are earned in a collaborative meritocracy.	Eclipse software is primarily an *input* to ecosystem activity. Contribution can be *direct* by donating existing code or other assets, or *indirect* by assigning staff and other resources to the community.
Influence on the Eclipse Foundation	Community representatives on board of directors and EMO councils; committer member powers.	Member powers vary by membership level: strategic members appoint a representative to the Board of Directors and EMO councils.
Innovation	Emphasis on the *creation* aspect of innovation.	Emphasis on the *commercialization* aspect of innovation.
Economic value	Emphasis on *value creation.*	Emphasis on *value appropriation* (value capture for the economic actor).

The Eclipse developer community and Eclipse business ecosystem are two components of an interconnected system with a common governance structure, shared platform, and co-dependencies for resources, and bound together through multilevel and nested interactions and shared actors (examined later in this section).

Figure 10 illustrates the conceptual model of the four components of the Eclipse field setting – the Eclipse Foundation, developer community, platform, and business ecosystem – and their relationships. Table 28 summarizes the two-step operationalization of each component, first, to specify as a formalized management construct with a corresponding scholarly literature, and second, to identify the salient institutional characteristics and situate those characteristics in the IAD conceptual apparatus.

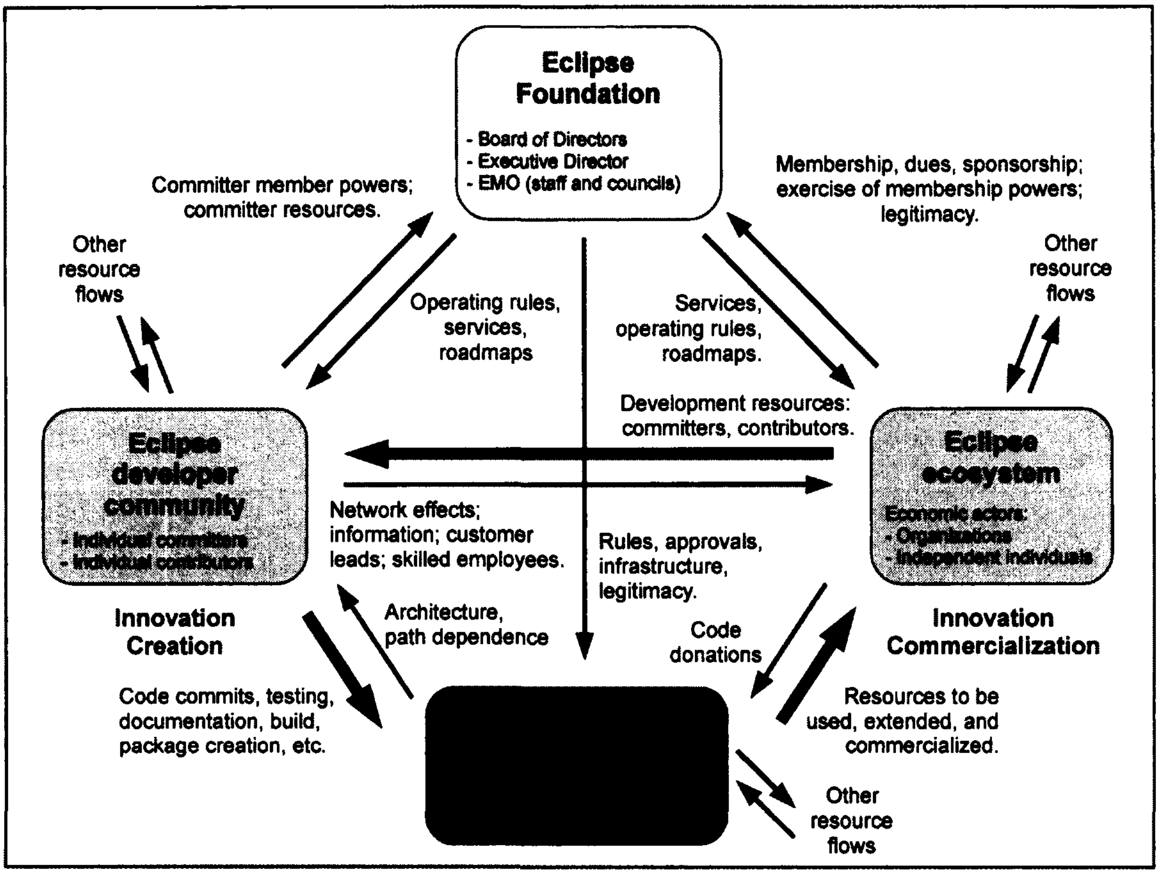

Eclipse Foundation
- Board of Directors
- Executive Director
- EMO (staff and councils)
Committer member powers; committer resources.
Membership, dues, sponsorship; exercise of membership powers; legitimacy.
Other resource flows
Other resource flows
Operating rules, services, roadmaps
Services, operating rules, roadmaps.
Development resources: committers, contributors.
Eclipse developer community
- Individual committers
- Individual contributors
Eclipse ecosystem
Economic actors:
- Organizations
- Independent individuals
Network effects; information; customer leads; skilled employees.
Rules, approvals, infrastructure, legitimacy.
Innovation Creation
Innovation Commercialization
Architecture, path dependence
Code donations
Code commits, testing, documentation, build, package creation, etc.
Resources to be used, extended, and commercialized.
Other resource flows

Figure 10: The Eclipse Foundation, community, platform, and ecosystem

Table 28: Operationalizing the Foundation, community, platform, and ecosystem

Component	Management construct	Position within IAD framework
Eclipse Foundation	The *keystone organization* of the Eclipse ecosystem (Iansiti & Levien, 2004a, 2004b) and an *open source software foundation* (O'Mahony, 2005).	A collection of *polycentric governance structures* that establish the operating rules of the community and ecosystem institutions. Also a *resource* performing services for the community and ecosystem.
Eclipse developer community	An *open source software community* (West & O'Mahony, 2008) of individuals who maintain and extend the Eclipse platform software.	An *institution* of individual participation in a *community*.
Eclipse ecosystem	A *business ecosystem* (Moore, 2006) of economic actors whose business activities are anchored around the Eclipse platform.	An *institution* of participation by economic actors in an *organizational field*.
Eclipse platform	A *platform* (Gawer, 2009a) of building blocks for developing complementary products, technology and services, and *open source software* and complementary assets (Teece, 1986).	A *commons resource* shared by the community and ecosystem institutions, and a bundle of *rules-in-use*.

Italicized words indicate specific technical concepts defined in chapters 2 through 5 and within this section.

7.8.2 Components

The *Eclipse ecosystem* is defined here as a *business ecosystem* adapted from Moore (2006): *an intentional collective of economic actors whose individual business activities, anchored around the Eclipse platform, share in some large measure the outcomes of the whole collective*[86]. This is consistent with the findings of section 7.6 on how the term "Eclipse ecosystem" is used within the practitioner community and a metaphor, label, and analogy (discussed further in chapter 11). As an operationalized construct in the parlance of institutional theory, the membership and scope of the Eclipse ecosystem are treated here as an *organizational field* – "those organizations that, in the aggregate, constitute a recognized area of institutional life" (DiMaggio & Powell, 1983, p. 143). According to Scott (2008, pp. 182), an organizational field encompasses:

- *a diverse array of organizations working within a given arena or domain;*
- *attention not only to the producer organizations, but to their exchange partners, customers, competitors, intermediary actors, regulators, and funding agents – their* organization set*; and*
- *recognition that organizations are particularly attentive to and influenced by the existence of organizations exhibiting the same general features and competing for the same resources – their* organizational population.

The organizational field is useful for examining "delimited systems ranging from markets to policy domains to the less structured and more contested arenas within which social movement struggle" (Scott, 2008, p. 182), and connects organization studies to the wider macrostructures of societies and world systems. The operational criteria for inclusion in the Eclipse ecosystem field follow directly from the definition above: (1) status as an

[86] The wording of this definition differs slightly from Moore (2006, p. 33) for reasons explained in section 3.6. Briefly, this definition uses "outcomes" in place of "fate" (to remove any connotations of predetermination of outcomes) and "collective of economic actors" in place of "community of economic actors" to clearly distinguish between the developer community and business ecosystem constructs. It also makes explicit the connection of business activities to the Eclipse platform.

economic actor, (2) intentionality, and (3) shared economic outcomes of business activities. Many of the most visible and highest-impact economic actors in the Eclipse ecosystem are *for-profit companies*, which consume Eclipse platform assets to provide Eclipse-based products, services, or technologies, to generate revenue directly by charging for the Eclipse-based offer, or indirectly by distributing an Eclipse-based complement to the offer for which customers pay[87]. Additionally, this definition also accommodates other forms of economic actor that are active in the Eclipse ecosystem, including university research groups, not-for-profit organizations, consultants providing Eclipse-based services, and individuals that engage in Eclipse-based business activity outside of any employment relationships[88]. Consistent with the practitioner explanations of section 7.7, ecosystem constituents are "profit-motivated" in the sense that pursuit of attractive profits is a motivation – but not necessarily the sole motivation – for action[89]. Chapter 10 develops an institutional representation of the Eclipse ecosystem from analysis of case evidence of participating companies.

[87] All eight company cases examined in chapter 10 fit this description.

[88] For example, an individual who develops, maintains, and distributes a software plug-in outside of the platform (thus an economic complement to the platform) and appropriates some economic benefit from that activity (whether or not that individual has an incorporated business) could be an economic actor within the business ecosystem organizational field, as defined here. In practice, the Eclipse ecosystem is comprised largely of for-profit companies and consultants with incorporated consulting companies; however, business ecosystems anchored around other platforms can and do include a large proportion of individuals behaving as economic agents, such as bloggers displaying Google ads in the Google ecosystem (Iyer & Davenport, 2008) or hobbyists distributing applications through a mobile "app store" (West & Wood, 2008). Nothing in this operationalization excludes economic actors who are not organizations.

[89] Chapter 10 presents the case evidence supporting the assertion of profit-motivated behaviour. Briefly, the observed behaviour of all case companies in this study was profit-seeking; as well, some informants reported parallel objectives for their companies, pursued simultaneously, that were compatible with seeking profits. No assumptions are required of exclusive profit-maximizing or hyper-rationality – only that pursuit of attractive profits is one motivation for action, in the same sense that Baldwin & Clark (2000, p. 12) argue that context acts "like a force" to "'pull' designs in the 'direction' of higher market value." This applies also to not-for-profit organizations that are unable to redistribute profits to shareholders, but are profit-motivated in the sense of obtaining revenues sufficient to cover the costs of operations.

The *Eclipse developer community* is an *open source software community* in the technical sense of the scholarly literature surveyed in section 3.6. West & O'Mahony (2008, p. 146) write:

> *[Open source software] communities are composed of individuals who collaborate toward a common goal but do not share a common employer and are not governed by an employment hierarchy. By using both online and offline means, open source software communities collectively produce software that is freely and publicly available – creating in effect a shared public good that can be used for either public or private purposes.*

The constituents of the Eclipse developer community are *individuals* – more specifically, the *individual contributors and committers who maintain and extend the Eclipse platform software*. Consistent with the findings of subsection 7.5.1 and the definitions of the *Eclipse Development Process* (examined in chapter 8), *contributors* are individuals who contribute to the platform is some way, by providing code, reporting bugs, discussing solutions, enhancing documentation, answering questions in the forums, and many other possible ways. *Committers* are the subset of contributors who have been nominated and elected by other committers to receive committer privileges, which include write-access to project resources[90]. Individual contributors and committers may be employed by business organizations, and those organizations may influence – perhaps quite strongly – the individuals' actions, but employers and other organizations in which individuals are involved are outside the scope of the developer community construct defined here[91].

[90] Subsection 8.2.1 examines the committer nomination and election process. Subsection 8.1.3 examines the operating rules, privileges, and responsibilities of committers and of other developer community roles.

[91] As argued in subsection 7.5.1, the Eclipse developer community is structured as a meritocracy where roles and reputation are conferred by other members of the community to individuals who demonstrate merit. Roles and reputation attach to an individual rather than an employer. An individual's roles, rights and responsibilities in the developer community do not change if the individual changes employers, and an individual is said to receive no special status from any particular organizational affiliation.

Likewise, consistent with the findings of subsection 7.5.1, the Eclipse developer community is a specific subset of the broader and much more inclusive practitioner notion of "Eclipse community" examined in section 7.5; users and adopters who are not contributors or committers as defined here are also outside the scope of the Eclipse developer community construct[92]. Chapter 8 develops an institutional representation of the Eclipse developer community from analysis of case evidence of Eclipse projects.

The *Eclipse platform* is a platform in the technical sense of the platforms literature (section 3.3; Gawer & Cusumano, 2002, 2008; West, 2003): *a set of building blocks providing a foundation on which firms and individual innovators can develop complementary products, technologies, or services* (Gawer, 2009a)[93]. It includes Eclipse software, as well as specialized and cospecialized complementary assets (section 3.1; Teece, 1986, 2009) for using and commercializing that software. This is consistent with the Eclipse Foundation view of the Eclipse platform as including not only software but also frameworks, processes, and other assets that anchor economic community (section 7.4; Milinkovich, 2008a), and the recommendation of Boudreau & Hagiu (2009) to include "non-price instruments" in the platform construct (section 3.3). Consistent with the specification of previous constructs, the individuals within the Eclipse developer community maintain and extend the Eclipse platform, and the economic agents within the

[92] In the present model, the user and adopter communities are implicit rather than explicit. For example, the Gamma & Beck (2003) community pyramid implies that today's users and adopters may be tomorrow's contributors and committers. Likewise, chapter 10 argues that large numbers of users and adopters motivate company participation in the ecosystem and resource flows from companies to the developer community and platform. Future research, discussed in chapter 12, could extend the model to make these interactions explicit and delve more deeply into the microstructures of components and resource flows.

[93] The word "platform" has a multitude of meanings within the Eclipse community, discussed further in subsection 11.1.1. Carefully weighing the advantages and disadvantages of various possible labels, "platform" was selected here to maintain consistency with extant management research.

Eclipse ecosystem consume and complement the platform. Within the IAD framework, the Eclipse platform is a *commons resource*, and a collection of *rules-in-use* that constrain and enable action. Subsection 7.8.4 develops an institutional representation of the Eclipse platform as a commons resource.

The *Eclipse Foundation* is both the *keystone* organization of the Eclipse ecosystem (section 2.5; Iansiti & Levien, 2004a, 2004b) and the *open source software foundation* providing governance, stewardship, and a legal identity to the Eclipse community (section 3.6; O'Mahony, 2005; Xie, 2008; Peters, 2009). As an operationalized institutional construct in the IAD framework, it is a collection of polycentric governance structures that shape the operating rules of the Eclipse developer community and Eclipse business ecosystem. It also acts to advance the platform, and some of the services that it performs have qualities of a IAD resources. Both the ecosystem construct and the community construct could include governance within their scope; however, the Foundation is broken out here as a separate entity because it is shared by both ecosystem and community, and it engages in activity outside of the scope of either construct. Chapter 9 develops an institutional representation of the Eclipse Foundation from analysis of case evidence of Eclipse polycentric governance structures.

7.8.3 Interactions between components

Each of the four components – the Eclipse Foundation, the Eclipse developer community, the Eclipse platform, and the Eclipse ecosystem – are related to each other component through flows of resources (Pfeffer & Salancik, 1978). Figure 11 is a subset of Figure 10 depicting only the components and relationships that comprise the *resource* *cycle* of innovation – a counterclockwise flow from the community to the platform to the ecosystem and back to the community. The scholarly literature on innovation distinguishes between the creation and commercialization aspects of innovation (section 3.1; Schumpeter, 1934; Freeman, 1982; Roberts, 1988, 2007; West and Bogers, 2010). Likewise, the open innovation literature distinguishes between value creation and value appropriation (or value capture) (section 3.2; Chesbrough & Rosenbloom, 2002; Chesbrough, 2003a, 2003b; Chesbrough et al. 2006). Within this conceptual framework, consistent with the practitioner explanations of the Eclipse field setting (section 7.7), the locus of innovation creation and value creation is the Eclipse developer community, and the locus of innovation commercialization and value capture is the Eclipse ecosystem. The Eclipse platform sits between as the outbound product of the developer community and inbound open innovation for the economic actors of the Eclipse ecosystem (section 3.2; West & Bolgers, 2010). Individuals within the Eclipse developer community maintain and extend the Eclipse platform, and economic agents within the Eclipse ecosystem consume and complement the platform. Economic actors in the ecosystem can participate in enhancing the platform assets in one of two ways – *directly* by contributing assets developed internally, or *indirectly* by providing development resources to the Eclipse development community. Consistent with practitioner explanations (section 7.7) and case evidence of Eclipse projects (chapter 8), it is the latter form of contribution – indirect contribution by providing development resources to the developer community – that is treated as the primary driver of platform innovation. There are also reciprocal resource flows in Figure 11 running in the opposite direction to the main cycle of production. The platform impacts the community by imposing constraints from past design decisions, particularly regarding APIs. Likewise, the Eclipse development community is a source for ecosystem resources, through network effects and as a source for information, customer leads, and recruiting of skilled employees.

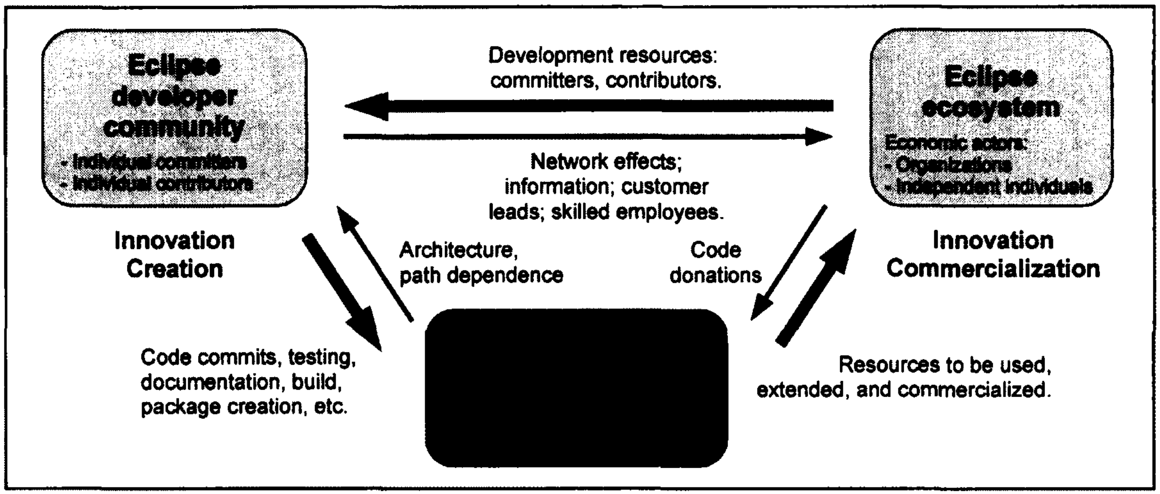

Eclipse developer community
- Individual committers
- Individual contributors
Innovation Creation
Development resources: committers, contributors.
Network effects; information; customer leads; skilled employees.
Eclipse ecosystem
Economic actors:
- Organizations
- Independent individuals
Innovation Commercialization
Architecture, path dependence
Code donations
Code commits, testing, documentation, build, package creation, etc.
Resources to be used, extended, and commercialized.

Figure 11: Production resource cycle (subset of Figure 10)

Figure 12 is a different subset of Figure 10 depicting only the resource relationships between the Eclipse Foundation and the three other components. The Foundation provides services, operating rules, and roadmaps to the community and ecosystem, and provides infrastructure, rules, and approvals to advance the platform. From the ecosystem, it draws organizational membership, membership dues, cash and assets from sponsorship, and legitimacy. From the community, it draws on committers and contributors to populate the councils and committees of the Eclipse Management Organization and perform some of the governance tasks delegated by the Executive Director. Ecosystem and community actors that are members of the Eclipse Foundation also exert influence through the exercise of membership powers specified in the governance documents. Chapter 9 examines these interactions in more detail.

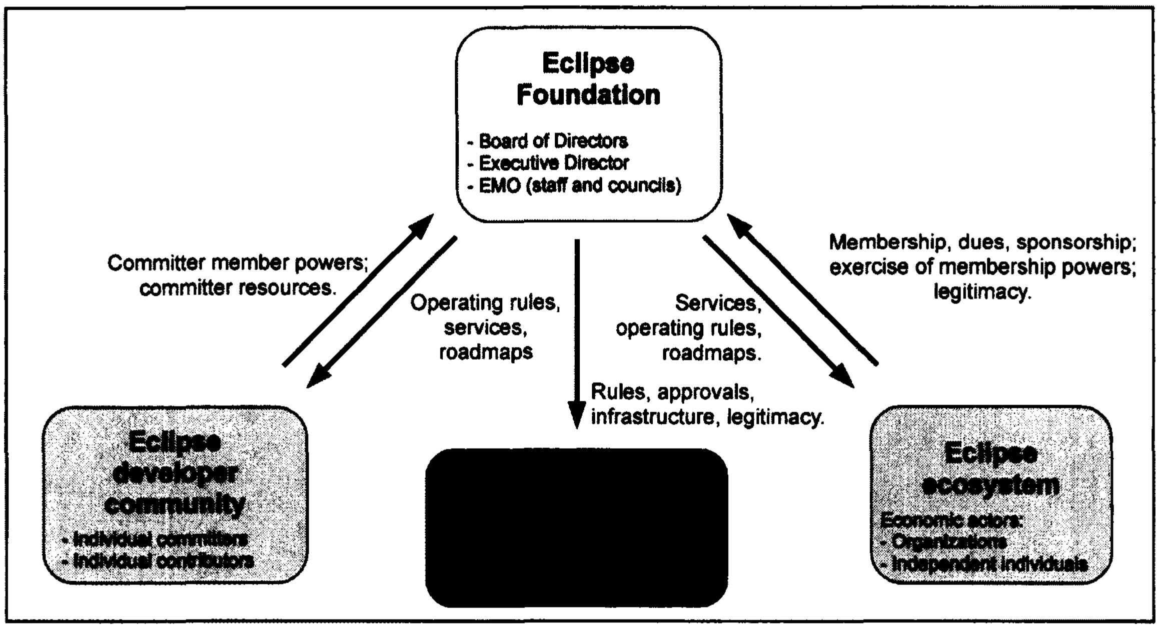

Eclipse Foundation
- Board of Directors
- Executive Director
- EMO (staff and councils)
Committer member powers; committer resources.
Operating rules, services, roadmaps
Rules, approvals, infrastructure, legitimacy.
Services, operating rules, roadmaps.
Membership, dues, sponsorship; exercise of membership powers; legitimacy.
Eclipse developer community
- Individual committers
- Individual contributors
Eclipse ecosystem
Economic actors:
- Organizations
- Independent individuals

Figure 12: Resource relationships of the Eclipse Foundation (subset of Figure 10)

The conceptual framework developed in this section provides a structure for further analysis. Chapters 8, 9, and 10 each develop IAD representations of the Eclipse developer community, Eclipse foundation, and Eclipse ecosystem, respectively, and the next subsection develops an IAD representation of the Eclipse platform.

7.8.4 The Eclipse platform as a commons resource

The analysis and discussion of the top-level case of the Eclipse field setting, and the conceptual development of this section provides the information to complete the IAD table shell for the Eclipse platform as an IAD commons resource (Table 29). The following paragraphs discuss and elaborate on the contents of Table 29.

Table 29: IAD representation of the Eclipse platform (March 2011)

IAD Element	Description
Underlying factors	
Rules-in-use	Property rights: The *Eclipse Public License* (EPL); no copyright assignment. • Business-friendly: weak copyleft; disclaimers of liability to protect contributors; patent license provisions to protect users. • Community-friendly: contributors retain ownership of contributions. • Comprehensive legal framework: Eclipse.org terms of use, intellectual property policy, trademark policy, membership agreement, committer agreement, etc.
Community attributes	n/a
Resource attributes	Artifacts Eclipse software platform tools, frameworks, and runtimes, distributed as source code and as executable packages: • Public goods: non-subtractable and non-excludable. • Supply-side: high fixed cost, near-zero marginal cost; large economies of scale. • Demand-side: large direct and indirect network effects; increasing returns. • Large: 33 million lines of code, 39 projects, and 490 committers in 2010 release. • Complex: The Eclipse annual release train is "arguably the best run, most predictable feat of software engineering on the planet" (Milinkovich, 2010: 29); complexity is managed through architectural modularity and design rules. • Predictable delivery: annual release always on time; published roadmaps. • Modular extensible plug-in architecture: small runtime kernel; open and stable interfaces; low barriers to entry for new developers; one set of design rules, available to everyone, governs the entire system. Complementary assets: the Eclipse brand; use of Eclipse trademarks; access to other participants; tutorials and videos contributed by the community; links to articles and books; committer guidelines; "webcast" events; legal resources; training; portal to community groups and other online resources. Facilities: Eclipse platform infrastructure • Platform artifacts are available for download at Eclipse.org anytime by anyone with a computer and an Internet connection. • Extensive and fully-supported IT infrastructure for Eclipse projects: developer mailing lists, user forums/newsgroups, project wikis, three supported source code repositories, Bugzilla issue tracker, MyFoundation portal, build servers. • Planet Eclipse blog aggregator. • Eclipse Marketplace Client for locating and installing plug-ins. • Eclipse Labs: project hosting and infrastructure for Eclipse-related projects outside the rules, processes, and endorsements of the Eclipse Foundation.
Action arena	
Actors	n/a
Action situation	n/a
Patterns of interaction	n/a
Outcomes	n/a
Evaluative criteria	n/a

The contents of this table are compiled from sources attributed throughout chapter 7.

Consistent with common practice in the IAD literature, this description of the Eclipse platform as a commons resources starts from its properties as an economic good (Ostrom, 1990), then considers size, complexity, and predictability (Ostrom, 2005), while enfolding insights from recent research on knowledge and digital assets as commons resources (e.g., Hess & Ostrom, 2003, 2007).

Following Hess & Ostrom (2003), Table 29 distinguishes between *artifacts* – discreet, observable, namable representations of ideas – and *facilities* that store artifacts and make them available. Eclipse platform *artifacts* include software source code, executable software packages of compiled object code optimized for various hardware platforms, and complementary assets such as documents and videos accessed through the Eclipse.org website. Eclipse platform *facilities* include an extensive information technology infrastructure accessed through the Eclipse.org website.

The Eclipse platform software inherits some of the economic properties of software goods (Messerschmitt & Szyperski, 2003), platform goods (Gawer & Cusumano, 2002), and information goods (Shapiro & Varian, 1999). In the traditional language of economics, Eclipse software platform artifacts are *non-subtractable in use,* meaning that utilization of the artifact by one economic actor does not make it unavailable to another, and *non-excludable*, meaning that one actor cannot easily exclude another from utilizing the artifact. These are the economic criteria of a *public good* (Ostrom & Ostrom, 1977). In general, public goods are largely immune from *dilemmas of over-utilization* (since utilization does not diminish the resource), but may be vulnerable to *dilemmas of under-production* (if there are costs to be paid privately that result in public benefits), or *dilemmas of enclosure* (where private interests seek to privatize the good through technology or imposition of restrictive property rights). Schweik (2007, p. 279) writes: "The challenge in [an open source software] commons is how to achieve collective action to create and maintain a commons or public good rather than the issue of protecting an existing commons from destruction (a public bad)."

On the supply side, software production has *high fixed costs* of creating new software and very low or *near-zero marginal costs* of replicating an additional unit of software already created. This results in *economies of scale* in supply that favour large producers and motivate small producers to band together, and strong incentives for re-use of existing assets and incremental innovation rather than greenfield development. On the demand side, open source software is available at low economic cost, but the high perceived value to the user of added services, such as provisioning, maintenance, operations, and customization, provides market incentives for specialty service providers and system integrations to offer these services. There are direct and indirect *network effects* in use. For example, an additional user of Eclipse software increases the size of the potential market for Eclipse plug-ins, increasing the incentives to create high-quality plug-ins – an *economic complement* to Eclipse software that increases the value of the platform to other users; similar arguments apply to many other complements. In general, the combination of strong network effects and economies of scale is expected to result in *increasing returns* (Arthur, 1994, 1996) that favour market convergence to a single platform or furcating convergence around a small number of co-existing platforms with distinct value propositions. The latter situation describes the current state of the IDE market, which has converged around two platforms: Microsoft Visual Studio for developing application software that runs on exclusively on Microsoft operating systems, and Eclipse for cross-platform development (Geer, 2005).

Many of these generic characteristics of the Eclipse platform are shared by software goods in general (Messerschmitt & Szyperski, 2003; Cusumano, 2004, 2010), and open source software more specifically. They are useful for distinguishing the Eclipse platform resource from, for example, a natural irrigation system or a public library, but other characteristics are needed to distinguish the Eclipse platform from other software commons.

In comparison to other software systems, the Eclipse platform is *large* and *complex*. The 2010 coordinated release had over 33 million lines of source code, including the work of 39 project teams and 490 committers (Eclipse Foundation press release, June 23 2010), and the 2011 release is expected to comprise 46 million lines of code and 62 projects. The entirety of the Eclipse platform is larger still, including also the Eclipse projects that choose not to participate in the coordinated release[94]. The complexity of the platform software arises, in part, from the breadth of its scope – from IDEs, to frameworks, to runtimes, for desktop computers, headless servers, and mobile phones, and across the entire breadth of industry verticals that comprises the software industry. "Apart from the [operating system], the IDE may now be the most complex piece of software running on your desktop computer" (Kersten blog, February 1 2011) – and the IDE is one of many application spaces addressed by Eclipse. However,

[94] Subsection 8.4.1 discusses the coordinated annual release, commonly called the Eclipse release train.

complexity is both mitigated and managed, in part, through a modular architectural design and well-specified public design rules (examined in section 7.4).

Also in comparison to other software systems, the Eclipse platform is *predictable*. The coordinated annual release of Eclipse software has delivered on-time for seven consecutive years on the fourth Wednesday each June. Roadmaps prepared midway through the development cycle inform the community and ecosystem what work is underway. Backwards compatibility is communicated through well-specified conventions. The Eclipse Foundation places high importance on predictability, as evidenced by staff presentations and self-assessments (e.g., Milinkovich, 2008a) and community dialogue within the projects and councils examined in chapters 8 and 9.

In comparison to the software artifacts, the elements of the Eclipse platform infrastructure vary more widely in their economic properties. Infrastructure resources with finite capacity are subtractable, and potentially vulnerable to over-utilization, resembling in some qualities the common pool resources of a natural resources commons. But unlike most biophysical commons discussed in section 5.2, information technology infrastructure can be upgraded to higher capacity when resources become available.

This chapter has presented the results of the top-level case of the Eclipse field setting, developed an emergent conceptual framework consistent with case results relating the Eclipse Foundation, platform, community, and ecosystem, and specified the IAD characteristics of the platform. The next three chapters present results from nested cases that delve more deeply into components of the conceptual framework, beginning in the next chapter with Eclipse projects and the institutions of individual participation.

8 Projects

This chapter examines activity at the project level of Eclipse in order to discover and articulate the institution that structures individual participation in extending and maintaining the Eclipse platform. It begins with the project rules specified in governance documents and communicated through community channels, then examines actual activity in Eclipse projects through direct observation and archival records, triangulated with participant interviews. From these observations, it develops an IAD representation of the Eclipse development community as an institution.

8.1 Project governance and organization

According to the *Eclipse Development Process* (EDP 2.5, section 4, p. 5), the activities of creating and maintaining Eclipse software occur within *Eclipse projects*:

> *A Project is the main operational unit at Eclipse. Specifically, all open source software development at Eclipse occurs within the context of a Project. Projects have leaders, developers, code, builds, downloads, websites, and more. Projects are more than just the sum of their many parts, they are the means by which open source work is organized when presented to the communities of developers, adopters, and users. Projects provide structure that help developers expose their hard work to a broad audience of consumers.*

Each project has a *project development team* (EDP 2.5, section 4.7, p. 8):

> *The Development team is composed of Committers and Contributors. Contributors are individuals who contribute code, fixes, tests, documentation, or other work that is part of the Projects. Committers have write access to the Project's resources (source code repository, bug tracking system, website, build server, downloads, etc.) and are expected to influence the Project's development.*

The development teams of all projects collectively comprise the Eclipse developer community of Eclipse contributors and committers, and individual participation in the community is thus structured, in part, by the project rules and operating procedures.

8.1.1 Project governance documents

The *Eclipse Development Process* outlines "how the Membership at Large, the Board of Directors, other constituents of the Ecosystem, and the Eclipse Management Organization (EMO) lead, influence, and collaborate with Eclipse Projects" (EDP 2.5, section 1, p. 2). It is a legally-binding governance document required by the *Bylaws of the Eclipse Foundation* (2003, 2008, sections 7.1-7.4, pp. 17-18); amendments to the EDP require a motion by the board of directors with super-majority (two-thirds) consent (*Bylaws,* 2003, 2008, section 3.9c, p. 5). The EDP serves several functions[95]:

- It articulates a set of *guiding principles* – an explicit statement of the values and beliefs motivating the processes and rules.
- It defines a *common vocabulary* for conversations about Eclipse projects.
- It specifies a set of *requirements* – actions and rules of behaviour mandated for all Eclipse Foundation projects: "Projects that fail to perform the required behaviors will be terminated by the EMO. In keeping with the Guiding Principles, the number of requirements must be kept to an absolute minimum" (EDP 2.5, section 3, p. 4).
- It endorses and directs the reader to additional *recommendations*, best practices, checklists, and other resources, but does not penalize projects for not following those recommendations.

It purposefully does *not* specify the *norms* of behaviour (EDP 2.5, section 2.4, p. 4)[96]:

> *Part of the strength of this document is in what it does not say, and thus opens for community definition through convention, guidelines, and public consultation. A document with too much structure becomes too rigid and prevents the kind of innovation and change we desire for Eclipse. In areas where this document is vague, we expect the Projects and Members to engage the community-at-large to clarify the current norms and expectations.*

[95] The emergent themes and category sets developed in this chapter and in subsequent chapters to organize and specify the rules-in-use of the Eclipse field setting differ from the labels and organization of Eclipse governance documents. For example, EDP 2.5 (2010) is structured in five sections: principles, requirements, structure and organization, roadmap process, and development process. From an institutional perspective, the mandatory rules for the roadmap process and the development process are both classified here as "requirements" along with the project operating rules specified in the requirements section.

[96] Section 8.6 reports the community norms evidenced in practitioner discourse and case project activity.

Other governance documents that structure project activity include the Eclipse *Intellectual Property Policy* (2008), the *Eclipse.org Terms of Use* (2008), the committer *Membership Agreement* (2010), and other legal paperwork regarding project contributions. The EMO cannot override or ignore the requirements of the EDP without endorsement of the board of directors; it is, however, instructed to "clarify, expand and extend this Process by creating a set of Eclipse Project Development Guidelines" (EDP 2.5, section 3.1, p. 4) maintained separately from the EDP; in practice, many of these guidelines are hosted at the *Development Resources* section of the *Eclipse Wiki*[97] and some – most notably the *Committer Guidelines* (2008) and *Eclipse Project Development* pages – are hosted at the Eclipse.org website. Recommendations and other non-mandatory resources are maintained in the Eclipse Wiki.

8.1.2 Guiding principles

The four _guiding principles_ of the EDP are (1) open source rules of engagement, (2) active community development, (3) deliberate cultivation of a commercial ecosystem, and (4) a process that is clear, concise, and evolving. These principles – a project-level interpretation of the Eclipse Foundation purpose defined in the *Bylaws* (2003, 2008) – are said to motivate the rules and processes of the EDP (EDP 2.5, section 2, p. 4).

The *open source rules of engagement* are openness to all participants, transparency of artifacts and information, and meritocracy – the same principles guiding community-managed open source software projects (EDP 2.5, section 2.1, p. 2). The

[97] The *Eclipse Wiki* (http://wiki.eclipse.org) is a hosted instance of the *MediaWiki* wiki software application hosted by the Eclipse Foundation staff at the Eclipse.org website. Registration is required is edit the wiki: some sections are moderated by the EMO so that only certain users can edit these documents directly; other community-developed pages require only Eclipse *Bugzilla* registration, available to anyone who agrees to the *Eclipse.org Terms of Use*.

second, third and fourth guiding principles are entwined (EDP 2.5, section 2.4, p. 4):

> *This document imposes requirements and constraints on the operation of the Projects, and it does so on behalf of the larger Eclipse community. It is an explicit goal of the Development Process to provide as much freedom and autonomy to the Projects as possible while ensuring the collective qualities benefit the entire Eclipse community.*

According to the EDP (EDP 2.5, section 2.2, p. 3), it is these guiding principles of active community development and the simultaneous cultivation of a commercial ecosystem that motivate additional demands on each project development team to make projects more attractive to others

8.1.3 Project organization

To establish a *common vocabulary* across all Eclipse projects, the EDP provides standard definitions of key terms and a standard project organizational structure. Eclipse projects are organized hierarchically such that project may contain zero or more projects. A *top-level project* sits at the top of the hierarchy, and possesses some special properties – in particular, a *project management committee* (PMC) that provides oversight and leadership for projects that fall under their top-level project (EDP 2.5, section 4.6.1, p. 7) and maintains a *project charter* that describes the purpose, scope, and project-specific operating rules for the top-level project (EDP 2.5, section 4.5, p. 7)[98]. As of March 2011, there were twelve top-level projects at the Eclipse Foundation.

A project with a *parent project* above it in the hierarchy may be called a *sub-project*. There is no inheritance of committers implied by the hierarchy: each project has

[98] The EMO provides a template for project charters (*Eclipse Standard Top-level Charter,* version 1.1), which includes standard text restating key requirements of the EDP that are mandatory for all projects. PMCs are encouraged to extend these requirements with additional project-specific rules and clarifications appropriate to the unique needs of each top-level project.

one set of committers, and committer status for an individual on one project does not imply committer status on any other project, including a parent project or a sub-project.

A project may designate a sub-project as an *incubator* for innovation outside of the main project. Incubators behave much like projects in the incubation phase of the project lifecycle, except that they do not graduate to become mature projects, and must include "incubator" in the project name (see also section 8.2.6 for more discussion of incubation and incubator projects). "Incubators are an excellent place to innovate, test new ideas, grow functionality that may one day be moved into another project, and develop new committers" (EDP 2.5, section 4.9, p. 10).

Committers may take on other roles within a project, including *project lead*, *PMC member*, and *PMC lead.* As with committer status, roles are assigned to individuals according to merit without consideration of organizational affiliation. Project leads and the PMC together comprise the *project leadership* and share three leadership responsibilities (EDP 2.5, section 4.6, p. 7): (1) ensure that the project is operating effectively by guiding the overall direction and by removing obstacles, solving problems, and resolving conflicts, (2) ensure that the project operates using open source rules of engagement, and (3) ensure that the project conforms to the EDP and Eclipse *Intellectual Property Policy.*

Table 30, Table 31, and Table 32 summarize the operating rules for project committers, project leads, and the PMC, respectively, as defined in the EDP.

Table 30: Operating rules of project committers (March 2011)

	Committers
Membership	"Becoming a committer is a privilege that is earned by contributing and showing discipline and good judgement. It is a responsibility that should be neither given nor taken lightly, nor is it a right based on employment by an Eclipse Member company or any company employing existing committers" (EDP 2.5, 2010, section 4.7, p. 8). The initial committers for a new project are appointed at the project *creation review.* Project contributors can become project committers through meritocracy: "The committers of a project have the exclusive right to elect new committers to their project – no other group, including a parent project, can force a project to accept a new committer" (EDP 2.5, 2010, section 4.1, p. 5). The *committer nomination and election process* is formally outlined in section 4.7 of the EDP, with additional clarification and requirements in the the Eclipse Wiki. See also subsection 8.2.1. Project leaders can revoke the committer status of a committer that has become inactive or disruptive.
Scope of authority	Committers have write-access to the project's resources and are expected to influence the project's development. The breadth of a committer's influence corresponds to the breadth of their contribution.
Responsibilities	• Sign a committer agreement to abide by the Eclipse *Bylaws*, *Committer Guidelines*, *Eclipse Development Process*, Eclipse *Intellectual Property Policy*, *Common Public License*, *Eclipse Public License*, and the *Eclipse.org Terms of Use*, and to complete and submit required questionnaires, employer consent forms, and other legal paperwork. • Active timely participation in developer mailing lists and user forums/newsgroups. • Monitoring and participation in discussions and votes on project matters. There are three voting responses to a project vote: +1 (yes), 0 (abstain) and -1 (no, or veto). • Proactive use of the bug tracking system to submit new problem reports, annotate existing reports, and provide updates on work relating to problem reports. • Committers may submit committer nominations for "contributors who have the trust of the project's committers" to become project committers. • All committers to a project have equal rights and responsibilities within that project.
Changes to the operating rules	The formal operating rules for committers are specified in the *Bylaws of the Eclipse Foundation*, the *Membership Agreement*, the various *Committer Agreement* documents, and the *Eclipse Development Process*. These documents may be amended according to rules outlined in the *Bylaws*.

Compiled from sources attributed in subsection 8.1.3, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Committer Guidelines, Committer Employer Consent Form*, *Common Public License*, *Eclipse Development Process* (EDP 2.5, 2010), *Eclipse Public License*, *Eclipse.org Terms of Use, Individual Committer Agreement, Intellectual Property Policy*, and *Member Committer Agreement.*

Table 31: Operating rules of project leads (March 2011)

	Project leads
Membership	The initial project leads for a new project are appointed at the project *creation review*. New project leads are elected by the project's committers, then approved by the project's PMC, and approved by Executive Director. The PMC can remove a project lead that becomes inactive or disruptive.
Scope of authority	Project leads manage an Eclipse project.
Responsibilities	• Ensure that the project's committers follow the EDP, and engage in the right sorts of activities to develop vibrant communities of users, adopters, and contributors. • Complete the new committer request form for newly elected committers. • Share project leadership responsibilities with the PMC (Table 32): "guiding the overall direction and by removing obstacles, solving problems, and resolving conflicts; operate using open source rules of engagement... and ensure that the project and its subprojects (if any) conform to the Eclipse Foundation IP Policy and Procedures" (EDP 2.5, section 4.6, p. 7). • Request project reviews as needed and provide review documentation.
Changes to the operating rules	Same as committers (Table 30).

Compiled from sources attributed in subsection 8.1.3, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Committer Guidelines, Common Public License, Eclipse Development Process* (EDP 2.5, 2010), *Eclipse Public License*, *Eclipse.org Terms of Use*, and *Intellectual Property Policy*.

Table 32: Operating rules of project management committees (PMCs) (March 2011)

	PMC members and PMC leads
Membership	The initial PMC members and PMC leads for a new top-level project are appointed and approved at the project *creation review* or *promotion review*. The creation of a new PMC requires an action by the Board of Directors with simple majority consent. New PMC members are elected by existing PMC members and leads, and approved by the Executive Director. New PMC leads are nominated by the Executive Director, and approved by an action of the Board of Directors with simple majority consent.
Scope of authority	Project management committees provide oversight and overall leadership for the projects that fall under their top-level project.
Responsibilities	• Ensure that the EDP is understood and followed by all subprojects. • Educate new committers on their responsibilities for safeguarding the intellectual property of Eclipse, and ensure that they have read, understood, and agree to follow the *Committer Guidelines*. • Maintain the *Project Charter* of the top-level project. • Approve the election of project leads by the committers of subprojects. • By unanimous vote, remove a project lead that has become inactive or disruptive. • Share project leadership responsibilities with project leads (Table 31).
Changes to the operating rules	Same as committers (Table 30).

Compiled from sources attributed in subsection 8.1.3, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Committer Guidelines, Common Public License, Eclipse Development Process* (EDP 2.5, 2010), *Eclipse Public License*, *Eclipse.org Terms of Use*, and *Intellectual Property Policy*.

8.1.4 Project resources

The Eclipse Foundation staff provide the development team of each project with a collection of infrastructure services. "The exact infrastructure provided by the Eclipse Foundation varies over time and is defined outside this process document" (EDP 2.5, section 4.2, p. 6); it presently includes a source code repository, hosting of a project website, developer mailing lists, user forums and newsgroups, space in the Eclipse wiki, a component in the Eclipse Bugzilla system for tracking issues and enhancement requests, space on a downloads server, and access to build resources. In Eclipse parlance, these are the *project resources* to which committers receive write access. Chapter 9 examines infrastructure and other services provided by Eclipse Foundation staff.

A project is not strictly required to make use of all the resources made available, and a project may elect to consolidate some services with a parent project or provide services for a sub-project. However, all committers to a project are expected to use the project's developer mailing lists for developer discussions, user forums and newsgroups for communicating with users, and Bugzilla for issue tracking and enhancement requests (see also subsections 8.2.3 and 8.2.4, and committer responsibilities in Table 30)

8.1.5 Revisions to the Eclipse development process

From section 8 (p. 19) of the EDP:

> *Due to the continued evolution of the Eclipse technology, the Eclipse community, and the software marketplace, it is expected that the [EDP] will be reviewed and revised on at least an annual basis.*

The original EDP document, Revision 1.0, was completed November 7 2003, approved by the Eclipse Consortium board of stewards, and came into effect at the launch of the

Eclipse Foundation in 2004. Some major sections of the EDP have been present since EDP 1.0 (2003), including project organization (a hierarchy with top-level projects, project management teams, project leads, and project charters), an iterative community process to produce and approve a roadmap for the Eclipse platform, and a project lifecycle with phases and reviews for the transitions between phases. EDP 2.3.1 (2007) was intended to "synchronize the process as-written and as-practiced," increase community participation, and introduce formal mentoring of new projects (*Abridged minutes of the Eclipse Foundation board of directors*, Q42006, December 13 and 14, pp. 13-15). It introduced the guiding principles and open source rules of engagement, the distinction between the user, adopter, and developer communities, and the distinction between requirements and guidelines. It revised the roles of the architecture council and planning council, and introduced detailed rules for the project reviews that transition a project between lifecycle phases. EDP 2.4 (2008) revised and simplified the guiding principles into their current form, introduced more information on committer and committer requirements, and added a new section on release management and release numbering. EDP 2.5 (2010) differs from EDP 2.4 in ten subsections; in explaining these revisions to Board of Directors and to the community, each change was said to be motivated by one or more of (1) better alignment of formal rules with informal norms ("the way things are actually done"), (2) greater clarity to reduce confusion, or (3) simplification and generalization by combining two or more concepts or rules into one concept or rule[99].

[99] For discussion of the changes and motivations for each change, see the *abridged minutes of the Eclipse Foundation board of directors* (2010Q1, March 22), explanations from Eclipse Foundation staff (Beaton

8.2 Mandatory requirements

The mandatory *requirements* imposed on all projects fall into a small number of constellations, including a common meritocratic process for nomination and election of committers, rules for intellectual property management, minimum standards of openness and transparency, obligations regarding community-building, participation in an Eclipse Roadmapping process to provide guidance to the community and ecosystem, a project lifecycle process with incubation and mentoring of new projects and community approval for transitions between between phases, and rules for release management. Each bundle of rules is discussed briefly below in its own subsection, followed by discussion of the enforcement mechanisms available to the Eclipse Foundation and the board of directors.

8.2.1 Committer nomination and election

The *committer nomination and election process* is the way in which individuals become committers on an Eclipse project (EDP 2.5, section 4.7, p. 8):

> *The election process begins with an existing Committer on the same Project nominating the Contributor. The Project's Committers will vote for a period of no less than one week of standard business days. If there are at least three (3) positive votes and no negative votes within the voting period, the Contributor is recommended to the project's PMC for commit privileges. If there are three (3) or fewer Committers on the Project, a unanimous positive vote of all Committers is substituted. If the PMC approves, and the Contributor signs the appropriate Committer legal agreements established by the EMO (wherein, at the very least, the Developer agrees to abide by the Eclipse Intellectual Property Policy), the Contributor becomes a Committer and is given write access to the source code for that Project.*

The steps of the process, including the notification of individuals whose action is required

blog, January 28 2010, February 2 2010, February 18 2010, March 5 2010, March 11 2010), and commentary on each approved change at the Eclipse.org website (EDP, version 2.5). Proposed changes to the EDP are reported and discussed using the same public infrastructure systems used to report and track bugs and enhancement requests for the Eclipse platform software. Bugzilla (#300000, #300002, #300006, #304878, #301065) are examples of EDP issue reports that were resolved by EDP 2.5.

and the tracking of those actions, is automated through the Eclipse Foundation IT infrastructure[100]. The specific paperwork comprising the committer legal agreements differs depending on several factors, including employment status and whether the individual is already a committer on one or more other projects. Employed individuals *must* obtain written consent from their employer (*Nominating and electing a new committer*, 2011):

> *If you cannot convince your employer to fill out the necessary paperwork, then you may not have write-access to the source code repository(s). This is the Board's position even if you are working on Eclipse projects on your own time. We realize that this prevents some talented and desirable people from being able to commit to the Eclipse projects but this is our IP risk reduction strategy.*

In all cases, the paperwork includes agreement to comply with the Eclipse *Intellectual Property Policy* (2008) and other Eclipse Foundation policies A new committer agrees also to a set of committer responsibilities summarized previously in Table 30.

8.2.2 Intellectual property management

All Eclipse Foundation member organizations and project committers – through the membership and committer agreements, respectively – agree to comply with the Eclipse *Intellectual Property Policy* (*IP Policy*, 2008), a legal document which sets forth how the Eclipse Foundation accepts content, redistributes content, and manages other intellectual property matters related to content (*IP Policy*, section I). According to interview respondents, the *IP Policy* is intended to help build trust by the organizations and individuals of the Eclipse ecosystem that Eclipse frameworks and tools are safely useable in commercial products with low risk of IP issues such as inadvertent patent or

[100] Specifically, the Eclipse *MyFoundation Portal* is an IT resource for process automation and workflow management at the Eclipse.org website. It is available to Eclipse Foundation members and committers.

copyright infringement. For the purposes of this research, there are five salient points.

First, the Eclipse Foundation accepts submission of content and redistributes content under the *Eclipse Public License* (EPL). Submission or redistribution of content under any license other than the EPL is possible only with a series of special approvals, including a resolution by the board of directors (*IP Policy*, section II and section III).

Second, the EMO accepts responsibility to ensure that each submission accepted for redistribution is committed to an Eclipse repository by an Eclipse committer, that sufficient license grants are obtained to allow redistribution, and that the *IP due diligence* is performed and records are kept (*IP Policy*, section II).

Third, the *IP due diligence process* (*IP Policy*, section IV) is a series of activities conducted on Eclipse content by the EMO to ensure compliance with the *IP Policy*. It covers several categories of submission, including original work by a committer, original work by a (non-committer) contributor, and derivative work from third-party sources. Eclipse software releases and release candidate builds may only contain content that has completed due diligence. Through a *Parallel IP Process* (2010), incubation phase projects may check content into an Eclipse source code repository with the expectation that due diligence will be completed before the project is permitted to pass a graduation review to become a mature project. Chapter 9 examines the activities undertaken by Eclipse Foundation legal staff to implement the due diligence process.

Fourth, there are IP requirements at project creation reviews, graduation reviews, and release reviews that must be completed for the project to pass the review (*IP Policy*, section IV and section V; see also subsection 8.2.6 of this document on the project incubation and lifecycle process).

Fifth, each committer must execute a *Committer Agreement* or have its employer execute a Committer Agreement on the committer's behalf (*IP Policy*, section VII). The committer agreement provides sufficient license grants to allow the Eclipse Foundation to distribute submitted content and it obligates the committer to comply with the Eclipse IP Policy (see also the committer responsibilities in Table 30). For example, committers agree to only commit code contributions that are submitted through Eclipse.org resources covered by the *Eclipse.org Terms of Use*, thereby ensuring that submissions are licensed under the EPL. Enforcing mandatory well-crafted committer agreements delivers, in part, on the EMO obligations of the second and third point.

8.2.3 Openness, access, and transparency

Following directly from the open source rules of engagement, "Project discussions, minutes, deliberations, project plans, plans for new features, and other artifacts are open, public, and easily accessible" (EDP 2.5, section 2.1, p. 2). In practice, project discourse occurs through the Eclipse.org IT infrastructure – developer mailing lists for development community discussions, the Eclipse Bugzilla issue tracking system for problem reports and enhancement requests, and the Eclipse wiki for collaborative multi-author documents. All posts are archived and publicly available to anyone. As part of the *Committer Agreement* signed by all committers, committers agree to monitor the mailing lists associated with their projects, respond in a timely fashion, and participate in committer votes about project decisions. "There are three voting responses: +1 (yes), -1 (no, or veto), and 0 abstain" (EPD 2.5, section 4.7, p. 9). Committers also agree to use the bug tracking system (EDP 2.5, section 4.7, p. 9):

> *Committers are responsible for proactively reporting problems in the bug tracking system, and annotating problem reports with status information, explanations, clarifications, or requests for more information from the submitter. Committers are responsible for updating problem reports when they have done work related to the problem.*

The mailing lists and bug tracking system are partially overlapping in purpose; projects may establish local conventions of where particular discussions are best conducted.

8.2.4 Community-building

"Projects are the level of communication with the larger Eclipse community and ecosystem... [and are] required to maintain an open and public communication channel with the the Eclipse community including, but not limited to, project plans, schedules, design discussions, and so on" and to make those communication channels easy to find (EDP 2.5, section 4.4, pp. 6-7). The community-building requirements place demands on projects in addition to those of openness and transparency. The openness, access and transparency requirements of the previous section ensure that anyone can participate ("access") and that information about project activity is public and easily accessible ("transparent"). Building communities – "a thriving, diverse and active community of developers... an active and engaged user community... [and] an active and engaged adopter/plug-in developer community" (EDP 2.5, section 2.3, p. 3) – requires active recruiting, time, effort, and creativity by the project leadership and project committers to encourage and nurture promising new committers with diversity of thought, and to reach out to and support users and plug-in developers[101]. In addition, "projects are required to

[101] The architecture council maintains a set of recommendations regarding community building (*Community development for Eclipse projects*, 2011). A project is said to be successful at building community when "a significant number of bugs raised against your project come from non-committers; non-committers are

report their status at least quarterly using the EMO defined status reporting procedures" (EDP 2.5, section 6, p. 11) so that information at the Eclipse.org website is current[102].

Large, active and engaged user and adopter/plug-in developer communities are said to be "key factors in creating a viable ecosystem around an Eclipse project, thus encouraging additional open source and commercial organizations to participate" (EDP 2.5, section 6, p. 3) "An active and engaged user community is proof-positive that the Projects, exemplary tools are useful and needed" and "an active and engaged adopter/plug-in developer community [outside of the Project's developers] is the only way to prove that an Eclipse project is providing extensible frameworks and extensible tools accessible via documented APIs." This is consistent with the Eclipse Foundation purpose to develop *both* exemplary tools for users and extensible frameworks.

8.2.5 The roadmap process

"The goal of the Roadmap is to provide the Eclipse ecosystem with guidance and visibility on the future directions of the Eclipse open source community" (*2010 Eclipse Roadmap*, version 5). The *Eclipse Development Process* explains the need for the roadmap process as follows (EDP 2.5, section 5, pp. 11):

> *This Development Process, the EMO, the Councils, and the Projects all acknowledge that the success of the Eclipse ecosystem is dependent on a balanced set of requirements and implementations. A Roadmap that provides too large a burden on the Projects will be rejected and ignored; similarly, a Roadmap that provides no predictable Project plans will be unhelpful to the business and technical plans being created by the ecosystem. A careful balance of demands and commitments is essential to the ongoing success of the Eclipse Projects, frameworks, and ecosystem.*

blogging about your project; articles, presentations, podcasts, webinars, etc. are being developed and presented by non-committers; [and] you cease to be the centre of the universe for your project."

[102] In practice, these requirements are met by updating the project metadata through the Eclipse *MyFoundation Portal*. Any committer can do this for any of their projects (*Project meta-data*, 2011).

The roadmap is prepared by the Eclipse Foundation councils and approved by the board of directors annually: "The EMO orchestrates interaction among and within the Councils to drive the Roadmap to convergence" (EDP 2.5, section 5, p. 10)[103].

Projects are obligated to participate by creating and maintaining a *project plan* in an EMO-defined file format (EDP 2.5, section 5, p. 11). The format and implementation details have evolved over time. The creation of the *2005 Eclipse Roadmap* (Version 1) was a manual process, but since 2008 the roll-up of project plans is partly automated with templates edited by the committers of each projects through the Eclipse *MyFoundation Portal* along with other project metadata[104]. Required project plan content includes expected deliverables of the next release, forecast milestones, compatibility with previous releases and explanations of incompatibilities, and cross-project dependencies.

8.2.6 Project incubation and the lifecycle process

The Eclipse Foundation has a standard project lifecycle process with six *phases* and managed transitions between phases (EDP 2.5, section 6, pp. 11-17). The six phases are (1) *pre-proposal*, (2) *proposal*, (3) *incubation*, (4) *mature*, (5) *top-level*, and (6) *archived*. New projects begin at the pre-proposal phase, and transition to the proposal phase when the EMO publishes a completed *project proposal* and announces it to the membership. All other transitions occur at *reviews* – open and transparent public review periods where the project leadership provides information to the membership and

[103] The interaction between governance groups to produce the roadmap, and the changes to the roadmap process over time are examined more closely in chapter 9 on Eclipse governance.

[104] In 2008, the board of directors instigated simplification of the roadmap process (*Abridged minutes of the Eclipse Foundation board of directors*, 2008Q2, June 18-19; Bugzilla #215301; see also *Project plan* , 2011; *Project meta-data* , 2011). In 2011, a proposed amendment to the *Bylaws* would replace the roadmap process with an annual community report written by Eclipse staff (examined in section 9.9).

receives feedback. The six review types are (1) *creation review* (for the transition from proposal to incubation), (2) *graduation review* (for the transition from incubation to mature), (3) *release review* (to update the membership and EMO prior to a release; discussed in more detail in subsection 8.2.7), (4) *promotion review* (for the transition to top-level project), (5) *continuation review* (to update the membership and EMO without a transition), (6) *restructuring review* (to make significant changes to a project), and (7) *termination review* (for withdrawal of a proposal or transition of an inactive or completed project to the archived phase). Each review type and each lifecycle phase has its own guidelines and check lists for the project leadership, maintained by the EMO (e.g., *Release Reviews*, 2011). The review requirements are tightly coupled to the requirements of the Eclipse *Intellectual Property Policy*, especially for creation, graduation, and release reviews (this was discussed in subsection 8.2.2).

According to interview respondents, it was originally envisioned that reviews would be synchronous events conducted by bringing together interested parties from the membership using teleconferences or other synchronous technology. Over time, this evolved to asynchronous reviews conducted over a *review period* with comments posted to a public discussion channel – a designated forum, mailing, or bug report – and a call held only if requested by at least one interested party. Version 2.5 of the EDP (2010) amended "review" to "review period" to better reflect actual practice (Bugzilla #304878).

Outcomes of a review are decided by the EMO: "The EMO(ED) approves or fails the review based on the public comments, the scope of the project, and the purposes of the Eclipse Foundation as defined in the Bylaws" (EDP 2.5, section 6.3, p. 15). From section 6.3 (EDP 2.5, p. 14):

> *The criteria for the successful completion of each type of Review will be documented in writing by the EMO in guidelines made available via the www.eclipse.org website. Such guidelines will include, but are not limited to the following:*
>
> 1. *Clear evidence that the project has vibrant committer, adopter and user communities as appropriate for the type of Review.*
> 2. *Reasonable diversity in its committer population as appropriate for the type of Review. Diversity status must be provided not only as number of people/companies, but also in terms of effort provided by those people/companies.*
> 3. *Documented completion of all required due diligence under the Eclipse IP Policy.*
> 4. *For Continuation, Graduation and Release Reviews, the project must have a current project plan, in the format specified by the EMO, available to the community.*
> 5. *Balanced progress in creating both frameworks and extensible, exemplary tools.*
> 6. *Showcase the project's quality through project-team chosen metrics and measures, e.g., coupling, cyclomatic complexity, test/code coverage, documentation of extensions points, etc.*

Review documents, comments from the community, and outcomes are publicly available.

The incubation phase in noteworthy in providing a supportive environment for a nascent project to develop the process, community, and technology of a fully-functioning open source project (EDP 2.5, section 6.2.3, p. 12; *Incubation phase*). Projects in the incubation phase have specific obligations and constraints, receive mentoring and support from the architecture council (EDP 2.5, section 6.1, p. 11), and are eligible to employ a *Parallel IP Process* (2010) that permits code to be checked into a repository after the IP Due Diligence process is underway but before it is complete. Obligations include clearly designating that the project and its software artifacts are in the incubation phase. As of EDP 2.5, a project may be designated as an "Incubator Project" and remain in perpetual incubation with no reviews (discussed previously in subsection 8.1.3).

8.2.7 Release management

Releases are "anything that is distributed outside of the committers of a project" (EDP 2.5, section 6.4, p. 17)[105]. The Eclipse rules for release management specify which projects can make a release, the requirements for making a release, and the release naming conventions. "Proper release management and reviews is a key aspect of Eclipse Quality" (EDP 2.5, section 6.4, p. 18).

Official releases are labeled with a release label of the form M.N according to well-defined labelling conventions where M and N are natural numbers (with additional letters and numbers under some circumstances). An Eclipse project in the *incubation phase* with two mentors may make interim pre-1.0 releases where M=0, e.g., "Eclipse EGit 0.11". "A project ready to make a 1.0 release is a project ready to graduate from incubation" (*Incubation Phase*, 2010). Any project in the *mature phase* may make a release where M>0, e.g., "Eclipse Equinox 3.6.2". All *official releases* for general distribution must successfully pass a *release review* (discussed in subsection 8.2.6)[106].

Projects may produce other software builds[107], such as *nightly and integration builds* for testing by the project developer community, and the *milestone* and *release candidate* builds for adoption and testing early adopters, which are not intended for

[105] This definition of "release", and many of the Eclipse Foundation release management requirements and recommended practices, are adapted from the Apache Software Foundation. "The Eclipse community has many of the same beliefs about releases as does the Apache community" (EDP 2.5, section 6.4, p. 17).

[106] There are additional recommendations regarding the version numbering of Eclipse plug-ins (*Version Numbering*, 2009); however, only the mandatory rules of the EDP rules are strictly required and enforced.

[107] There are additional release naming conventions for service releases (".1" or ".2" appended to the end of the release name), release candidate builds ("RC" and the release candidate number appended to the build name), and milestone builds ("M" and the milestone number appended to the end of the build name), and different naming conventions for integration builds (build name begins with "I" followed by the date), and nightly builds (build name begins with "N" followed by the date). The key point is that mandated conventions for release management across all Eclipse projects ensure that an Eclipse release label conveys a great deal of information to an informed consumer about the contents of that particular release.

general release. These are not considered *official releases* and do not require release reviews; as a consequence, they are not to be promoted outside their target groups.

8.2.8 Enforcement mechanisms

The Eclipse Management Organization and Executive Director have four project-level enforcement mechanisms available: (1) termination of a non-conforming project (EDP 2.5, section 3, p. 4), (2) removal of project leadership (EDP 2.5, section 4.6, p. 7), (3) unfavourable outcomes of the IP due diligence process (*IP Policy*, 2008, section IV), and (4) unfavourable outcomes of project reviews (EDP 2.5, section 6.3, p. 15). The application of these mechanisms – or more correctly, their *potential* application since they are rarely observed in practice – are the formal rules-based levers available for enforcing project requirements. Don Smith, the Eclipse Director of Ecosystem Development, explains the Executive Director powers as follows:

> *The Bylaws give the Executive Director the authority to step into a project and fix something that's broken, but that almost never happens. Most of the time, the potential for the ED to use this authority is enough to motivate the stakeholders of a problem to work things out for themselves.*

The EDP also specifies a grievance handling process by which a committer or organizational member can raise a concern up the leadership chain, eventually appealing to the board of directors (EDP 2.5, section 6.5, p. 18).

8.3 Guidelines and community norms

The mandatory requirements are a partial view of the rules-in-use. A second category of rules-in-use is comprised of guidelines and community norms that are widely adopted within the Eclipse community.

8.3.1 The Eclipse Way

The Eclipse Way (Gamma & Wiegand, 2005) was the title of a plenary EclipseCon talk that attempted to codify the development practices of the Eclipse Project, the top-level project containing the three core projects of the Eclipse Java IDE and the Eclipse SDK: the Eclipse platform project, Java Development Tools (JDT), and the Plug-in Development Environment (PDE)[108]. *The Eclipse Way* has no official status in Eclipse project governance, but it continues to be a popular label for Eclipse community norms and expectations (e.g., *Incubation Phase*, 2010), and Gamma & Wiegand (2005) is frequently cited in community discourse and writing about project best practices.

Gamma & Wiegand (2005) argue that the Eclipse Project development team consistently achieves quality and on-time delivery through five categories of "Eclipse practices" that collectively result in better feedback from the community[109]. First, there is a well-established *project rhythm* that partitions the year-long release cycle into 6-week milestones cycles[110]. Each milestone cycle is miniature software development cycle that culminates in the production of a *milestone build* – software that is intended to be good enough to be used by early adopters within the community. Each cycle also includes a retrospective of what went well and what did not, as an opportunity for learning. Build processes are fully automated with *nightly builds* to discover integration problems

[108] The authors, Erich Gamma and John Wiegand, were the founding lead of the Eclipse JDT project, and the founding lead of the Eclipse platform project, respectively.

[109] At the time of Gamma & Wiegand (2005), the Eclipse Project had released Eclipse 3.0 (June 25 2005), the first release following the launch of the Eclipse Foundation, and was approximately three months away from the release date of Eclipse 3.1 (June 27 2005).

[110] According to Gamma (2008), the Eclipse Way was the outcome of much experimentation: "We tried shorter iterations, like two weeks, but we settled on longer iterations for our distributed teams. Each six week iteration is comprised of one week of planning, four weeks of work, and a week to stabilize." Development of Eclipse 3.0 involved more than sixty IBM developers at eight different IBM locations: Winnipeg, Lexington, Ottawa, Toronto, Beaverton, Raleigh, Saint Nazaire, and Zurich.

between components and (weekly) *integration builds* for which all automated unit tests must be successful. The teams consume their own output, treating each build as a release candidate that is expected to work. In their daily software development work, each developer on an Eclipse component team uses the nightly build of their own component and the integration builds from the other components. Each milestone build has an accompanying explanation of what is "New and Noteworthy" to advertise recent work to the community, provide more incentives for early adopters to try milestone builds, and enable better and higher quality community feedback.

Second, there is *continuous testing*. Each build executes a test suite of automated unit tests that were developed as the software was created and modified.

Third, there is a clear *end game*. As the release date approaches, the project rhythm shifts from a development mode of six-week milestone cycles to a convergence process with a sequence of short test and fix passes that involve early adopters in the community[111]. There is shared responsibility and commitment around delivering quality – "we all sign off" (p. 32) – and a shared objective of "deliver on time, every time" (p. 42), dropping features to meet schedule and quality. Following the release, there is a short period of decompression to recover and to perform a retrospective of the last cycle: achievements, failures, process, and cross-team collaboration. On a release cycle of twelve months, typically milestones consume nine months, the endgame consumes two months, and decompression consumes one month.

Fourth, there is *early and incremental planning*. Release themes establish the big

[111] The Gamma & Wiegand (2005) "convergence process" of test and fix passes was the prototype for the release candidate (RC) builds of the annual release train (discussed in subsections 8.2.7 and 8.4.1).

picture, with input from the community and from the requirements council. The component teams define component plans. The PMC collates the initial project plan draft. The plan is final only at the end of the release; until then, it is updated quarterly to reflect progress on items, new items, and input from the community. The PMC meets at least once a week and all component leads and the PMC meet for a weekly planning call that covers status, planning, and identification of cross-component issues, with meeting notes posted to the developer mailing lists. Cross-component issues are addressed by establishing dynamic teams with participants from each effected component.

Fifth, Eclipse software is *built to last*, with deliberate preservation of architectural integrity (recognizing that decisions in one release impact the what can happen in the next release), and building a solid foundation that is scalable, high performing and stable. APIs are the interfaces between software artifacts and also the dependencies between component-centric project teams. "APIs don't just happen; we need to design them" (p. 47). Don't commit API before its time: "We would rather provide less API than desired (and augment) than provide the wrong (or unnecessary) API and need to support it indefinitely" (p. 45). APIs are developed with consuming clients: "Tension among components is healthy for coming up with good component interfaces/APIs" (p. 46).

Reflecting on *The Eclipse Way* some years later, Gamma (2008) calls it a mash-up of common agile software development practices such as short design iterations, open source development practices such as "release early and release often," and some scaling up practices for larger distributed teams, and it borrows freely from the processes of Extreme Programming (Beck, 1999) and from Scrum (Schwaber & Beedle, 2001).

The practices of Gamma & Wiegand (2005) evolved in at least three directions. First, these practices formed the basis for IBM's *open commercial development* process on the IBM Jazz project (Gamma, 2008) and other IBM projects developed outside of Eclipse (Gamma & Slobojan, 2008; Azad blog, July 7 2009)[112]. Second, they continued to evolve as guidelines and checklists for project reviews at the Eclipse wiki and the recommended best practices for Eclipse projects maintained by the architecture council. Third, some of these practices became codified as the release engineering process for the Eclipse annual release train (discussed in subsection 8.4.1).

8.3.2 Other guidelines and norms

Other project guidelines include recommendations, best practices, checklists, and other resources at the Eclipse wiki and Eclipse.org website. Some examples of guideline documents include the *Development resources*, the *New committer handbook, Community development for Eclipse projects, Starting a new project, Nominating and electing a new committer* and *Words of wisdom and bits of advice*. Section 13.1 provides full citations for each salient wiki document. Section 8.6 examines the adoption of recommendations within case projects.

Community norms on case projects are examined in section 8.6.

8.4 Opt-in programs

A third category of rules-in-use result from non-compulsory "opt-in" programs.

[112] According to Gamma (2008), IBM open commercial development is more than publishing the source code; it is an open, transparent process, from feature requests and planning through delivery. Community members can (1) download, tryout, and provide feedback on betas and incubators, including source code, (2) access, create, and update work items, (3) access milestone and component iteration plans, (4) access the development wiki, and (5) participate in discussions on the development community newsgroups. This resembles IBM's role in the Eclipse Consortium prior to the launch of the Eclipse Foundation.

Projects that voluntarily choose to participate in these Eclipse Foundation programs agree to additional operating rules and obligations.

8.4.1 The annual release train

Executive Director Mike Milinkovich calls the Eclipse annual release train "arguably the best run, most predictable feat of software engineering on the planet" (Milinkovich, 2010, p. 29). Participation in the annual release train is an option available to qualifying Eclipse projects – participating projects agree to an additional set of rules and processes, and are dropped from the train if they do not comply. Each year, the participation in the release train has increased, from ten projects in 2007, to 39 projects in 2010, to 62 projects in 2011. The release train has never missed a forecast release date – always the fourth Wednesday in June.

From the first release of Eclipse 1.0 in 2001, major releases of the *Eclipse SDK* – the software development kit comprising the Eclipse core platform, the Java development tools (JDT), plug-in development environment (PDE) – have occurred annually with two service releases between major releases. Major releases and service releases are provided as executable code ("binaries"), compiled and tested for various operating systems. Other Eclipse projects follow their own release plans, which may or may not coincide.

Beginning with the Eclipse Callisto release in 2006, other Eclipse projects have been invited to release in a coordinated schedule along with the Eclipse SDK – variously referred to as the *simultaneous release*, *coordinated release*, *annual release* or the *Eclipse release train.* Each major release occurs in June, with follow-up *service releases* in September (SR1) and February (SR2). Each release is named, to distinguish the identity of the release from the version numbering of the projects and plug-ins that participate in the release[113] (Table 33). Chronological timelines in later sections – case projects (Figure 14 in this chapter), foundation governance (Figure 15 in chapter 9), and case companies (Figure 18 in chapter 10) – label the release trains as points of reference.

Table 33: Eclipse platform versions and simultaneous releases

		Main Release		# of projects releasing together	# of packages released	First Service Release		Second Service Release	
#	Platform Version	Name	Date			Name	Date	Name	Date
1	1.0	Eclipse 1.0	Nov. 7 2001	3	1	n/a	n/a	n/a	n/a
2	2.0	Eclipse 2.0	Jun. 27 2002	3	1	Eclipse 2.0.1	Aug. 29 2002	Eclipse 2.0.2	Nov. 7 2002
3	2.1	Eclipse 2.1*	Mar. 27 2003	3	1	Eclipse 2.1.1	Jun. 27 2003	Eclipse 2.1.2	Nov. 3 2003
4	3.0	Eclipse 3.0	Jun. 25 2004	3	1	Eclipse 3.0.1	Sept. 16 2004	Eclipse 3.0.2	Mar. 11 2005
5	3.1	Eclipse 3.1	Jun. 27 2005	3	1	Eclipse 3.1.1	Sept. 29 2005	Eclipse 3.1.2	Jan. 18 2006
6	3.2	Callisto**	Jun. 26 2006	10	1	Eclipse 3.2.1	Sept. 21 2006	Eclipse 3.2.2	Feb. 12 2007
7	3.3	Europa***	Jun. 27 2007	21	4	Europa SR1	Sept. 28 2007	Europa SR2	Feb. 29 2008
8	3.4	Ganymede	Jun. 25 2008	23	7	Ganymede SR1	Sept. 24 2008	Ganymede SR1	Feb. 25 2009
9	3.5	Galileo	Jun. 24 2009	33	9	Galileo SR1	Sept. 25 2009	Galileo SR2	Feb. 26 2010
10	3.6, 4.0	Helios	Jun. 23 2010	39	12	Helios SR1	Sept. 24 2010	Helios SR2	Feb. 25 2011
11	3.7, 4.1	Indigo	Jun. 22 2011	62	11	Indigo SR1 (planned)	Sept. 23 2011 (planned)	Indigo SR2 (planned)	Feb. 24 2012 (planned)
12	4.2	Juno	Jun. 27 2012 (planned)	tbd	tbd	Juno SR1 (planned)	Sept. 28 2012 (planned)	Juno SR2 (planned)	Feb. 22 2013 (planned)

Compiled from various sources including *Eclipse Downloads* (http://eclipse.org/downloads), *Eclipse Project Downloads* (http://download.eclipse.org) and the *Eclipse Project Archives* (http://archive.eclipse.org).
* Build 2.1.3, a third service release to Eclipse 2.1, was released March 10 2004.
** The service releases for Callisto, the first simultaneous release, included only the SDK components.
*** Build 3.3.1.1, a "dot release" of the Eclipse Project, followed Europa SR1 on October 23 2007.

Coordinating activities for the simultaneous release is a major project management effort led by the Eclipse planning council (examined in section 9.6). The coordinated release occurs each year on the fourth Wednesday in June – Wednesday June 23 for the 2010 Helios release, Wednesday June 22 for the 2011 Indigo release, and it is planned for Wednesday June 27 for the 2012 Juno release. According to Gamma (2008),

[113] As with most community discussions, the conversation about naming each simultaneous release is conducted in Bugzilla; e.g., Bugzilla #108146 is the conversation about naming the Callisto release.

"We don't ship at Christmas and we don't ship in summer – thus, we ship in June."

Projects on the release train "opt in" to a set of simultaneous release requirements that are in addition to or more stringent than the project requirements of EDP (*Eclipse simultaneous release*, 2010). First, projects on the release train must complete some EDP requirements earlier than otherwise required, including project plans, IP logs, and release review materials. Second, projects agree to a suite of technical requirements, including version numbering, formatting and signing of bundles, provision of software bundles and metadata, and consistency of branding and license text, and agree to promptly fix cross-project problems identified by the Planning Council. Third, projects agree to "be a good Eclipse citizen ... and document it" (*Eclipse simultaneous release*, 2010) by reporting on compliance (or non-compliance) with a set of best practice items at each build milestone[114]. This makes mandatory many of *The Eclipse Way* (Gamma & Wiegand, 2005) practices that are otherwise recommended but not required of all Eclipse projects. A planning council exception process permits projects to request exemption from any particular requirement by providing sufficient rationale and notification.

This project rhythm of the release train (Figure 13) is that of the *The Eclipse Way* (Gamma & Wiegand, 2005). Throughout the year there are six *milestone builds*[115], labeled M1 through M6, occurring on Fridays at roughly six week intervals on a schedule published by the planning council (*Indigo simultaneous release plan*, 2010). As the release date approaches, the build schedule changes to weekly release candidate builds,

[114] The planning council maintains a consolidated report on the compliance status of each tracked item for all projects on the release train (*Simultaneous release compliance grid*, 2011).

[115] Milestone builds are also called stable builds; the former label is more common in communication among developers, the latter more common in communication to users.

labeled RC1 through RC4, occurring on sequential Fridays between the last milestone build and and the simultaneous release. Between the final RC4 build and the release is "quiet time" where no build are planned: "That time is reserved for final, in depth testing, and preparation for release. Emergency rebuilds might be considered, by following the usual Planning Council Exception Process, but only for serious, blocking regressions" (*Indigo simultaneous release plan*, 2010).

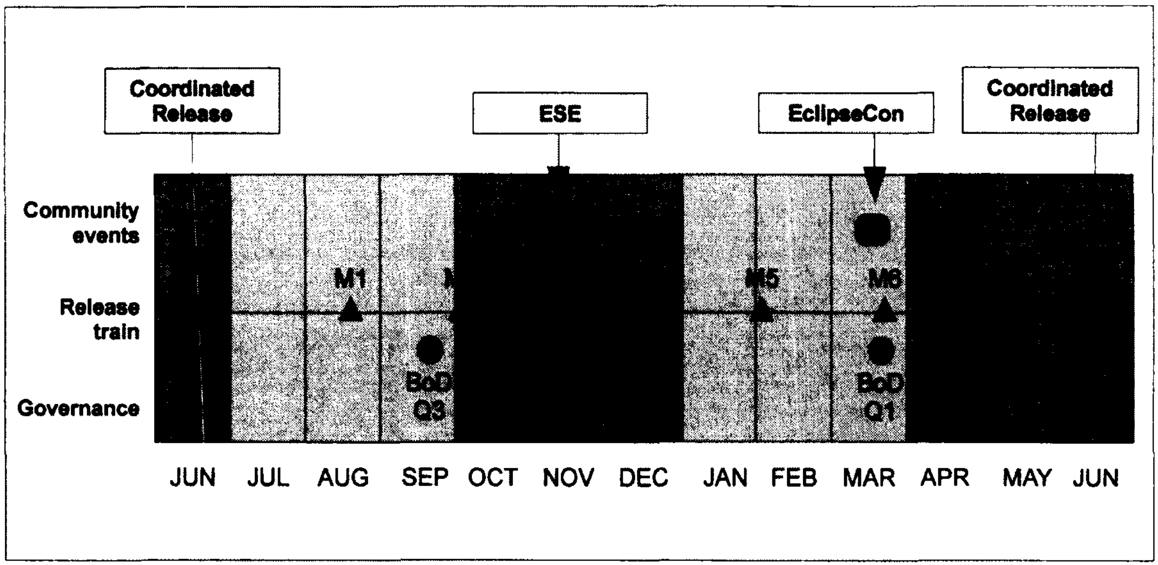


Figure 13: The Eclipse year structured by the annual release train milestones

The train metaphor, refers in part, to the sequencing of project deliverables. To accommodate dependencies between projects, each project has deliverables due at times offset from the end-date of the build. For the Indigo release, there are four delivery times, labeled +0 (first), +1, +2, and +3 (last), with projects self-selecting the most appropriate category[116]. "The actual time-offset represented by these intervals change over the course of the year of development, being several days at first [for milestone

[116] The components of the Eclipse Project – the platform project, JDT, and PDE – deliver first (+0), and the packages of the Eclipse Packaging Project (discussed in subsection 8.4.2) deliver last (+3).

builds], but then only one day near the end of the release [for release candidate builds]" *(Indigo simultaneous release plan,* 2010).

The milestones for the annual release train structure "the Eclipse year" for participating projects. Other events structuring the Eclipse year (also shown Figure 13) are EclipseCon, the annual North American conference held each March in North America, Eclipse Summit Europe (ESE), the annual European conference held each November in Germany, and the quarterly meetings of the Eclipse Foundation Board of Directors and the Eclipse Members Meeting near the end of each quarter.

8.4.2 Eclipse packages

Eclipse packages are tested configurations of the Eclipse platform with a specific set of Eclipse plug-ins intended for specific user applications. Beginning with the Europa simultaneous release in 2007[117], packages have been the preferred way of distributing Eclipse tools to the user community. The creation and maintenance of packages occurs within the Eclipse Packaging Project (EPP)[118]. Each package has a *package maintainer* who performs or coordinates the technical work of creating, testing and maintaining the package[119]. Projects may request inclusion in a package, but decisions about package content rest with each package maintainer[120]. For Eclipse projects, inclusion of the project's software in popular packages is a way to reach potential new users.

[117] The Eclipse Packaging Project launched in February 2007 (epp-dev mailing list, Ward, February 20 2007). Callisto, the 2006 simultaneous release prior to Europa, included advice on optional features for particular applications, e.g., plug-ins to download for development in C/C++, but did not provide packages.

[118] The EPP built all Helios packages for five target environments: Windows 32-bit, Windows 64-bit, Linux 64-bit, Linux 32-bit, and Mac OSX *(Project plan for Eclipse Packaging Project,* version 1.3).

[119] The Eclipse Wiki documents the steps to create a package *(EPP/How to create a package,* 2010).

[120] Requests occur through Bugzilla. For example, Bugzilla #330226 is the formal request to include the m2e project with the Eclipse IDE for Java Developers package.

Helios, the Eclipse 2010 simultaneous release, was distributed as twelve different packages. Table 34 summarizes the twelve Helios packages, with download counts recorded on September 23 prior to first service release[121]. Table 35 maps each Helios package against the various projects contributing content to that package.

Table 34: Eclipse Helios packages (2010, platform version 3.6.0)

Rank	Package name	Size (MB)	# of downloads	Package description
1	Eclipse IDE for Java EE Developers	206	1,376,226	Tools for Java developers creating Java EE and Web applications.
2	Eclipse IDE for Java Developers	99	692,102	The essential tools for Java development.
3	Eclipse Classic 3.6.0	170	607,664	The classic Eclipse download with source code and documentation. [*Eclipse SDK* (software development kit), with the Eclipse Rich Client Platform and the CVS, JDT, and PDE plug-ins]
4	Eclipse IDE for C/C++ Developers	87	271,636	An IDE for C/C++ developers.
5	Eclipse for PHP Developers	143	252,707	Tools for PHP developers creating Web applications.
6	Eclipse IDE for JavaScript Web Developers	108	65,720	Tools for JavaScript developers creating Web applications.
7	Eclipse IDE for Java and Report Developers	241	52,339	JEE tools and BIRT reporting tool for Java developers to create JEE and Web applications that also have reporting needs.
8	Eclipse Modeling Tools	252	42,647	A collection of Eclipse Modeling Project components with developer tools and source.
9	Eclipse for RCP and RAP Developers	191	40,508	Tools for developers who want to create Eclipse plug-ins, Rich Client or Rich Ajax Applications.
10	Pulsar for Mobile Developers	122	39,994	A tools platform for Mobile Developers that makes it easy to download SDK from different handset manufacturers.
11	Eclipse SOA Platform for Java and SOA Developers	187	22,353	Runtimes and tools integration platform for service-oriented architecture developers.
12	Eclipse IDE for C/C++ Linux Developers (*)	121	20,784	Augments the C/C++ package with additional tools specifically for Linux developers.

Download counts recorded on September 23 2010 prior to the first service release (Helios SR1).
(*) The Eclipse IDE for C/C++ Linux Developers package is available only for Linux (two versions: 32-bit and 64-bit). All other packages are available for Windows, Mac, and Linus, in both 32-bit and 64-bit versions.

[121] Downloads through the Eclipse.org website are a subset of Eclipse distribution. Other channels not reflected in Eclipse.org website analytics also distribute Eclipse software, including the *Eclipse Member Distro program* which encourages member companies to provide distributions through their own servers, and enterprise deployments which provision through enterprise infrastructures. The numbers in Table 34 are useful for order of magnitude comparison of the relative popularity of different packages, but they should not be interpreted as measures of Eclipse adoption.

Table 35: Eclipse packages and projects (Helios release, 2010)

		Packages											
		Classic (*) (Eclipse SDK)	Java	Java EE	C/C++	RCP/RAP/ Plugin (*)	Reporting (*)	Modeling (*)	PHP	Pulsar	SOA	Javascript	C/C++ Linux
	First included in release — Year	2001	2007				2008		2009			2010	
	Name	1.0	Europa				Ganymede		Galileo			Helios	
Project Name	Description — Rank in 2010	3	2	1	4	9	7	8	5	10	11	6	12
RCP/Platform	The base *Rich Client Platform* for building rich client applications.	●	●	●	●	●	●	●	●	●	●	●	●
CVS	Plug-ins for integrated software revision control with the *Concurrent Versions System* (CVS).	●	●	●	●		●	●	●	●	●	●	●
Mylyn	Plug-ins for task and application lifecycle management to improve developer productivity.		●	●	●	●	●	●	●	●	●		●
JDT	*Java Development Tools*: plug-ins for a Java IDE supporting Java application development.	●	●	●		●	●	●		●			
XML Tools	*Source Editing Project*: plug-ins for "web" languages including XML.		●	●		●	●		●	●			
GEF	*Graphical Editing Framework*: plug-ins to create a rich graphical editor.		●	●		●	●	●					
PDE	Plug-in Development Environment: plug-ins to create, develop, test, debug, build and deploy Eclipse plug-ins, RCP, and RAP products, and OSGi tooling for component development.	●		●		●	●	●					
WTP	*Web Tools Platform*: plug-ins for developing web applications.			●			●		●		●	●	
EMF	*Eclipse* Modeling *Framework*: plug-ins for model building and code generation.		⊙	●			●	⊙	●				
DTP	*Data Tools Platform*: plug-ins for developing and managing data-centric systems.			●			●						
EclipseLink	Plug-ins for a comprehensive open-source Java persistence solution.			●			●						
Java EE Tools	Plug-ins for development of J2EE artifacts			●			●						
RSE	Target Management plug-ins to configure and manage remote systems and services.			●			●						
CDT	*C/C++ Development Tooling*: plug-ins to create an industrial-strength C/C++ IDE.				●								●
ECF	Eclipse Communications Framework: plug-ins for distributed servers, applications and tools.					●							
BIRT	*Business Intelligence and Reporting Tools*: reporting system plug-ins for Java/J2EE apps.						●						
GMP	*Graphical* Modeling *Project*: plug-ins for graphical editors based on EMF and GEF.							●					
MDT	*Model Development Tools*: plug-ins for industry standard metamodels and modeling tools.							●					
PDT	PHP Development Tools: plug-ins for a PHP IDE.								●				
MTJ	*Mobile Tools for Java*: plug-ins to extend Eclipse Java development to mobile devices.									●			
Swordfish	Plug-ins for an extensible service-oriented architecture framework based on Equinox.										●		
Linux Tools	Plug-ins for a full-featured C and C++ IDE for Linux developers (builds on capability of CDT).												●

Table notes: ● Included ⊙ Partially included (*) The *RCP/RAP/Plugin*, *Classic*, *Reporting*, and Modeling packages include source code for some components.

8.5 Project case selection

Of the more than 120 active Eclipse Projects (*Executive Director Report*, March 2011), this research closely examines five case projects:

- *BIRT*, the Eclipse *Business Intelligence and Report Tools* project, provides an Eclipse-based open source reporting system for web applications. Its components include a report designer based on the Eclipse tooling platform, a server-side runtime, and a charting engine.
- *CDT*, the Eclipse *C/C++ Development Tooling* project, provides an integrated development environment for the C and C++ programming languages.
- *Equinox* is the software component model and runtime platform used by most Eclipse projects and the Eclipse platform. It is a core element of the Eclipse runtime (RT) platform – a generic standards-based framework for developing modular Java software. The Equinox p2 subproject is a software provisioning platform used by Equinox and also deployable separately.
- *JDT*, the Eclipse *Java Development Tools* project, provides an integrated development environment for the Java programming language. It was the first Eclipse tool developed on the Eclipse platform, first released with the Eclipse 1.0 SDK in November 2001. JDT was originally developed as a proprietary IBM internal project founded prior to the Eclipse Foundation and Eclipse Consortium.
- *Mylyn* is the Application Lifecycle Management framework (ALM) for Eclipse. It provides a task-focused interface for Eclipse IDEs and connectors to popular (non-Eclipse) software development tools, such as version control systems, code review tools, and bug tracking systems. Mylyn is included with most Eclipse IDE packages (Table 34). It was promoted to a top-level project in September 2010.

These five Eclipse projects provide variation in at least five dimensions.

The first dimension is breadth of company participation. Software development of the BIRT and JDT projects is performed largely by committers employed by one company which also leads the project. Software development on CDT and Equinox is performed by committers employed by multiple companies – approximately ten companies for CDT, and six companies for Equinox. In 2010 and 2011, the Mylyn project experienced rapid growth in company participation following a successful restructuring review and promotion to a top-level project[122]. At the Eclipse Helios release in June 2010, there were six Mylyn committers, all employed by one company. By March 2011, there were sixteen new Mylyn committers, and ten companies either employing committers or stating intent to do so.

The second dimension is age and maturity. The JDT, CDT, and Equinox projects were founded prior to the launch of the Eclipse Foundation in 2004. JDT is one of the original founding Eclipse projects (November 2001), CDT was added few months later (2002), and Equinox launched the following year (2003). BIRT was one of the first projects approved by the newly formed Eclipse Foundation board of directors in 2004[123]. Mylyn launched in 2005 in the Eclipse Technology Project incubator.

Table 36 situates each of the five case projects in a two-by-two table according the first two dimensions. Values for each dimension are provided in parenthesis.

Table 36: Positioning of project cases

		Origins	
		2001-2004 **Prior to the launch of the Eclipse Foundation**	**2004+** **Launched into the Eclipse Foundation**
Company participation	**Community-driven: committers from many companies**	• CDT (2002, 10 companies) • Equinox (2003, 6 companies)	• Mylyn (2005, 1 company in 2010, ~10 companies in 2011)
	Company-driven: committers from one or few companies	• JDT (2001, 1 company: IBM)	• BIRT (2004, 1 company committing code, 3 companies on PMC)

[122] The Mylyn restructuring review occurred in April 2010 prior to finalizing the project case selection (*Mylyn restructuring review*, April 7 2010). The announcement that Mylyn was promoted to a top-level project occurred in mid-September while longitudinal data collection for this study was underway (Kersten blog, September 16 2010; Milinkovich blog, September 16 2010).

[123] Discussion of the BIRT proposal was an agenda item on the first documented meeting of the Eclipse Foundation board of directors (*Abridged minutes of the board of directors*, Q2 2004).

The third dimension is the nature of the design artifacts produced by the project. The BIRT, CDT, and JDT projects each create tools for software development that build on the Eclipse tooling framework. Mylyn is a set of productivity plug-ins that work across multiple Eclipse tools, including JDT and CDT, and connect to other software systems. Equinox is the kernel of the Eclipse runtime platform. BIRT also includes non-tool components including a server-side runtime built on Equinox. The Eclipse software platform is said to comprise software tools, software frameworks, and runtimes, and the software produced by these projects reflects that full range of platform diversity.

The fourth dimension is the project organization. The JDT, CDT and Equinox projects are subprojects within a top-level project, but they differ in internal structure: JDT has a granular substructure with multiple subprojects, CDT is organized as a single project with no substructure, and Equinox has a granular substructure that includes an incubator subproject separate from core development. BIRT is a top-level project with multiple components, but those software components are all organized together in a flat project structure with no subprojects. Mylyn was initially incubated in the Eclipse Technology top-level project, then promoted to the Tools top-level project (as a sibling of CDT), then recently promoted to its own top-level project; its structure is still unfolding, and its inclusion here fortuitously permitted direct observation of those provisioning activities and processes.

The fifth dimension is the segment of Eclipse Foundation members to which the project most strongly appeals. Presentations by the Eclipse Foundation staff (e.g., *Proposed Membership Changes*, 2008) report four waves of membership growth: a first wave of Java tool vendors, a second wave of embedded tools vendors, a third wave of companies joining to participate in framework projects, rich client applications and runtime projects, and a fourth (anticipated) wave of enterprises joining to participate in industry verticals. JDT was instrumental to the first wave of membership growth, CDT to the second, and BIRT and Equinox to the third. Mylyn adds value to all segments.

Collectively this sample of projects includes broad variation and replication in the number of different organizations involved (expected to impact the action arena and committer diversity), founding circumstances (under different versions of the EDP, as well as prior to a codified development process), characteristics of the software artifacts, internal organization, and target user community. As argued in 6.2.3, this is intended to elucidate project similarities and differences in order to identify community norms of the developer community (characteristics that are the same across most or all projects of the sample), and local conventions (characteristics that differ significantly between projects).

Project case selection was finalized September 1 2010 in order to directly observe all case projects concurrently. Table 37 summarizes development activity on each case project at the time of case selection, sorted by project and participating company, including counts of commits (writes to the software repository) and committers (individuals with write access) employed by each company[124]. The table also indicates the organizational affiliation of the project lead (or co-leads) and which organizations are developed into company cases in chapter 10.

[124] The source for the Table 37 counts of commits and committers by project and organizational affiliation is the Eclipse Dash Project (http://dash.eclipse.org), accessed September 1 2010. "Unknown" is a Dash Project category indicating missing information from the database – common mainly during the Eclipse Foundation's first year of operation.

Table 37: Company and committer activity in case projects

	Commits				Number of committers (Sept. 1 2010)							
	Lifetime of project		Jan. – Aug. 2010									
Project / company	#	%	#	%	Total	Active	Participating	Inactive	Alumni	Emeritus	Project Lead	Company Case
BIRT	**195104**		**11153**		**109**	**26**	**2**	**25**	**56**	**0**		
Actuate	170350	88	11133	100		25	2	21		0	x	x
IBM	2397	1	0	0		0	0	3		0		x
Innovent	8	0	0	0		0	0	1		0		x
Individual	22346	11	18	0		1	0	0		0		
CDT	**123921**		**15144**		**36**	**22**	**2**	**1**	**6**	**5**		
Intel	41458	33	2504	17		5	0	0		1	x	x
IBM	31453	25	3688	24		2	1	0		2		x
QNX	21982	18	1363	9		1	0	0		2		
CodeSourcery	6778	5	68	0		1	0	0		0		
Nokia	4478	4	3011	20		5	1	0		0		x
Wind River	1881	2	0	0		0	0	0		0		
Ericsson AB	1416	1	840	6		1	0	1		0		
Freescale Semiconductor	1348	1	753	5		2	0	0		0		
Institute for Software (IFS)	412	0	59	0		1	0	0		0		
Mentor Graphics	148	0	16	0		1	0	0		0		
Individual	11571	9	2841	19		2	0	0		0		
Equinox	**398457**		**26343**		**57**	**11**	**8**	**23**	**15**	**0**		
IBM	109942	28	20498	78		6	5	10		0	x	x
Sonatype	30166	8	3144	12		1	0	0		0	x	x
EclipseSource	9816	2	60	0		1	0	1		0	x	x
Cloudsmith Inc.	2752	1	1377	5		0	1	1		0		
ProSyst Software	1405	0	73	0		1	0	2		0		
Compeople AG	489	0	0	0		0	0	1		0		
Tasktop	329	0	329	1		0	1	1		0		
Weigle Wilczek GmbH	203	0	0	0		0	0	1		0		
SAP AG	0	0	0	0		0	0	1		0		
Individual	241515	61	292	1		1	1	5		0		
Unknown	1427	0	520	2		1	0	0		0		
JDT	**290170**		**5630**		**49**	**13**	**3**	**19**	**13**	**1**		
IBM	239679	83	5603	100		12	3	16		1	x	x
Oracle	471	0	0	0		0	0	1		0		
Individual	20151	7	27	0		1	0	2		0		
Unknown	29682	10	0	0		0	0	0		0		
Mylyn	**97929**				**7**	**5**	**1**	**0**	**0**	**1**		
Tasktop	94705	97	5685	93		4	1	0	0	0	x	x
Individual	1315	1	418	7		1	0	0	0	1		
Unknown	1820	2	0	0		0	0	0	0	0		
TOTALS					**258**	**77**	**16**	**68**	**90**	**7**		

Notes: *Active* committers have made at least one commit to the source code repository in the past three months. *Participating* committers have made at least one commit in the past nine months, but not in the past three months. *Inactive* committers are neither active nor participating. *Alumni* are former committers who no longer have commit privileges. *Emeritus* are alumni specially recognized for exception past contributions.

8.6 Cross-case analysis

Table 38, Table 39, and Figure 14 compare the five case projects[125]. Table 38 examines software artifacts and project organization. Table 39 examines interaction and decision-making within the development community, including company participation. Figure 14 is a visual timeline of project reviews and software releases for each project. The next set of tables more closely examines specific forms of communication, interaction, and activity within each case project: commits to the project source code repository (Table 40), bug reports in Bugzilla (Table 41), posts to project user forums and newsgroups (Table 42), and activity on developer mailing lists (Table 43).

All five case projects would be deemed successful in all or most of the success dimensions comprising the three clusters of project success metrics identified in the *Eclipse Development Process* (EDP 2.5, section 2.3). Evidence presented at project release reviews shows active and engaged user communities around all five projects ("proof-positive that the Project's exemplary tools are useful and needed") and active and engaged adopter communities ("the only way to prove that an Eclipse project is providing extensible frameworks and extensible tools accessible via documented APIs"). It is the third EDP success metric, a thriving, diverse, and active developer community, where the projects show the greatest variation. All five developer communities are active with respect to code commits, bug reports, community contributions, and other R&D metrics, but some are developed by committers from one organization while others have attracted broader and more diverse participation by committers employed by multiple companies.

[125] These tables and figures are compilations of evidence from the project case notes documents constructed according to the method outlined in subsection 6.2.3. For example, the contributions row in Table 39 was obtained by manually examining project IP Logs for the most recent software release of each project.

Table 38: Comparison of project cases (part 1 of 2): artifacts and organization

	BIRT **Business intelligence and reporting tools**	**CDT** **C/C++ Development Tooling**	**Equinox**	**JDT** **Java Development Tools**	**Mylyn** **Application Lifecycle Tools**
Software artifacts	Reporting system for web applications, especially Java enterprise systems. Includes a report designer, a report runtime, and a chart engine. Extensible with user-created plug-ins.	Tool plug-ins that implement an Eclipse C and C++ IDE.	Component model used by Eclipse tools, frameworks, and runtimes. Equinox is the kernel of the Eclipse Runtime Platform, the Rich Client Platform, and the Eclipse IDE.	Tool plug-ins that implement a Java IDE. JDT allows Eclipse to be an IDE for itself.	Task and application lifecycle management (ALM) framework for Eclipse. Implements a task-focused interface with connectors to other software tools.
User segment	Software developers creating reporting applications.	Software developers creating C and C++ programs.	Software developers creating OSGi runtime applications.	Software developers creating Java programs.	Software developers using any programming language.
Adopter segment	Business intelligence consultants and vendors.	Embedded tools vendors. Embedded device providers.	Providers of runtime software.	Java tool vendors.	Software tool vendors.
Project origins	Created September 2004 as a new top-level project. Proposed by Actuate Corporation. Initial code contribution by Actuate Corporation.	Created January 2002 as the first project of the new Tools top-level project. Initial code contribution by QNX. Parsing code contributed by IBM.	Created February 2003. Incubated within the Eclipse Technology project. Initial code contribution by IBM. Graduated to Eclipse Project in September 2005. Moved to RT (runtime) top-level project at the RT creation review in March 2008.	Released November 2001 with the Eclipse SDK 1.0. Initial code contribution by IBM was part of the $40M IBM code donation at the launch of the Eclipse Consortium.	Created July 2005. Incubated within the Eclipse Technology project. Graduated to the Eclipse tools project November 2007. Promoted to a top-level project and restructured in September 2010.
Organization (March 2011)	Flat project structure: 3 code components (report designer, runtime, charting), each with mailing lists and Bugzilla components, organized together as one project; no subprojects.	Flat project structure: 1 subproject added in 2010 from a restructuring review of another project; no other partitioning.	Partitioned as 5 subprojects: framework, bundles, p2 (a provisioning system), security and incubator. All except p2 are managed together (shared mailing lists, leadership, meetings, etc).	Partitioned as 3 subprojects: core, debug, and UI (user interface).	Partitioned as 8 subprojects: builds, commons, context, docs, incubator, reviews, tasks, versions. Provisioning and activation of subprojects still underway.

Table 39: Comparison of project cases (part 2 of 2): community interaction

	BIRT **Business Intelligence and reporting tools**	**CDT** **C/C++ Development Tooling**	**Equinox**	**JDT** **Java Development Tools**	**Mylyn** **Application Lifecycle Tools**
Participant interactions [1]	Mailing lists used mainly to broadcast announcements. Weekly PMC meetings.	High-traffic mailing lists used for community discussion. Monthly project meetings. Biweekly working group calls.	High-traffic mailing lists used for community discussion. Biweekly project meetings.	Mailing lists used mainly to broadcast announcements. Weekly Eclipse PMC planning meetings.	High traffic mailing lists used for community discussion. Weekly project meetings attended by non-committers.
Participation					
Committers (Dec. 1 2010) [2]	109 lifetime total. 53 current committers. 31 active in study period.	36 lifetime total. 25 current committers. 25 active in study period.	57 lifetime total. 42 current committers. 22 active in study period.	49 lifetime total. 35 current committers. 19 active in study period.	13 lifetime total. 22 current committers. 12 active in study period.
Companies employing active committers (June 2009 - Nov. 2010)	Actuate (all active committers) IBM (PMC) Innovent (PMC)	Nokia (6), Intel (5), IBM (3), Freescale Semiconductor (2), Institute for Software (1), Mentor Graphics (1), Ericsson (1), QNX (1), Code Sourcesery (1).	IBM (5), EclipseSource (1) Cloudsmith (1), Prosyst (1) Sonatype (1), Tasktop (1).	IBM (all committers).	Tasktop (6), individuals (5), Ericsson (1). Organizations expected on newly provisioned subprojects: Cloudsmith, Ericsson, INSO, Itema, OBEO, Perforce, SAP.
Contributions (most recent release) [3]	23 code contributions from 9 contributors affiliated with at least 3 organizations.	1002 code contributions from 206 contributors affiliated with at least 50 organizations.	306 code contributions from 67 contributors affiliated with at least 18 organizations.	241 code contributions from 48 contributors affiliated with at least 8 organizations.	79 code contributions from 23 contributors. (More than 900 *total* contributions of all sizes)
Committer nominations (June 2009 - Nov. 2010)	1 new committer. (employed by Actuate).	6 new committers: 4 Nokia, 1 Freescale Semiconductor, 1 Texas Instruments.	12 new committers: 12 to incubator component. 4 to p2 subproject. (5 companies, 5 individuals).	6 new committers: 4 to the core component. 2 to the UI component. (all are IBM employees).	9 new committers: all to Mylyn Reviews. 4 SAP, 3 Ericsson, 1 Red Hat, 1 individual.
Community development (examples)	• *BIRT Exchange* community website. • "HelpWanted" tag on bugs well-suited to community contribution.	• Extensive and detailed "getting started" resources. • CDT summits. • Open project meetings with non-committer participation.	• *Eclipse RT Portal* with case studies and other runtime resources. • Equinox and RT summits. • Incubator for new ideas.	• "How to get involved" guidelines for new contributors.	• Public recognition for community contributions. • Incubator for new ideas. • Open project meetings with non-committer participation.

[1] All five case projects actively utilized the Eclipse Bugzilla public issue tracking system, and interacted with users through user forums and newsgroups.
[2] Aggregated from queries to the Eclipse Project Dashboard. "Current committers" is a snapshot on December 1 2010 of all individuals with commit privileges.
[3] Counts only "large" code contributions (>250 lines) from non-committers, as recorded in the project IP Log for the most recent release as of March 2011 (except as noted above).

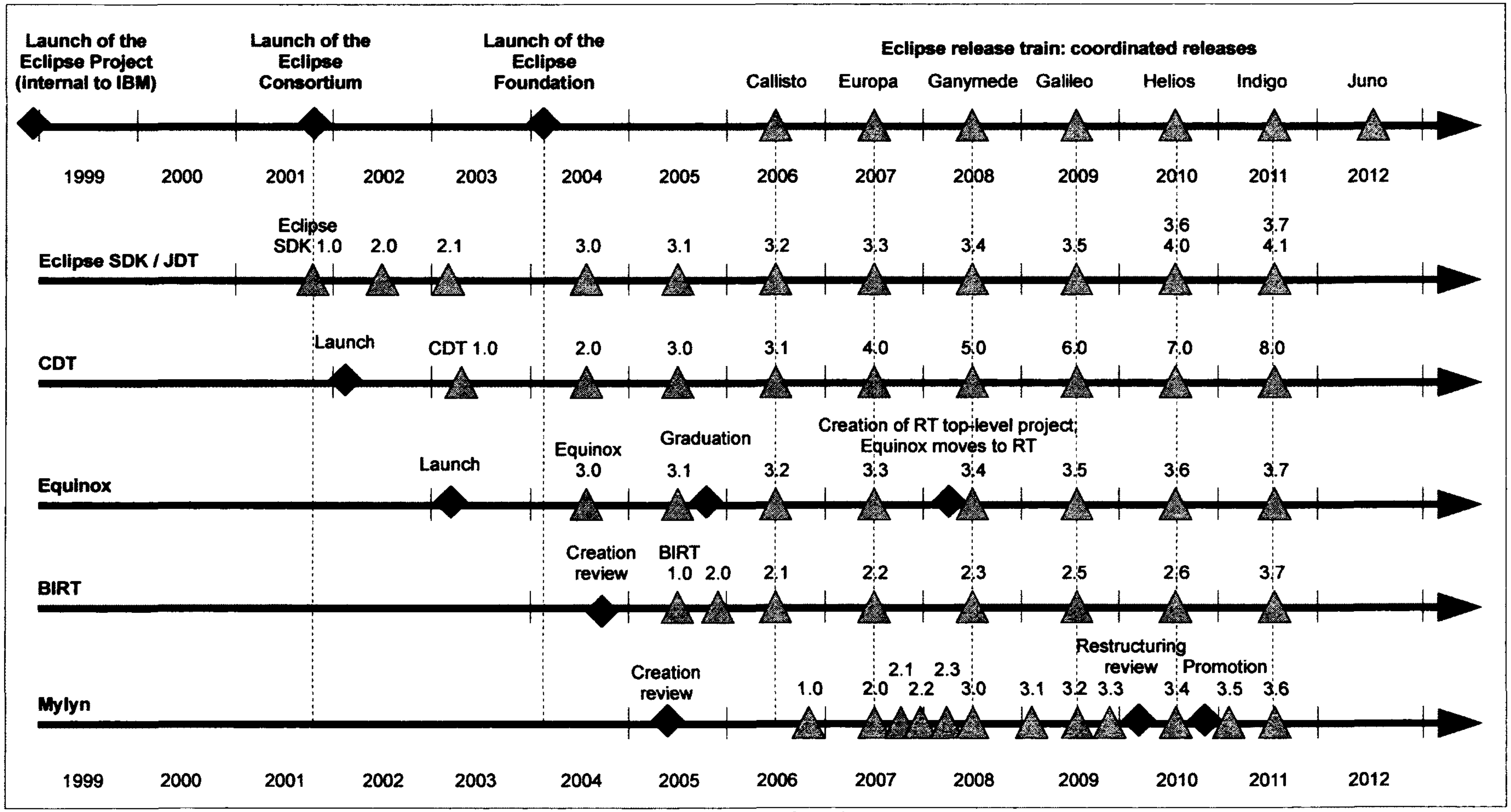


Figure 14: Timeline of project cases

Table 40: Project activity: Code commits to source code repository

	Commits						Committers
	Sept. 2010	Oct. 2010	Nov. 2010	3-month total	# of active committers	# of active firms	Companies that employ committers (September 1 – November 30 2010)
BIRT	430	1367	2055	3852	24	1	Actuate (24)
CDT	885	944	515	2344	19	9	Intel (5), Nokia (4), IBM (2), Ericsson, Freescale, IFS, QNX, Texas Instruments
Equinox	712	659	2490	3861	15	5	IBM (10), Cloudsmith, EclipseSource, ProSyst, Sonatype
JDT	508	452	337	1297	17	1	IBM
Mylyn	514	191	62	767	5	1	Tasktop

The Eclipse Foundation provides and maintains three software code repositories and version control systems available for Eclipse projects: Concurrent Versioning System (CVS), Subversion (SVN), and EGit (an Eclipse implementation of the Git distributed version control system). Of these five projects, BIRT, CDT, Equinox, and JDT employ CVS, Mylyn employs EGit, and none employs Subversion.

Table 41: Project activity: Bugzilla

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BIRT	128	222	688	1078	20232	21310
CDT	337	256	535	2651	11602	14253
Equinox	298	199	574	1749	8194	9943
JDT	402	389	801	3953	41603	45556
Mylyn	207	141	408	1285	7543	8828

Data in this table are obtained from Eclipse Bugzilla queries with the following parameters.
Changes: >= 2010-09-01, <= 2010-11-30 (chfieldfrom=2010-09-01&chfieldto=2010-11-30).
New: chfield=[Bug creation]. *Closed*: chfield=resolution (This query will count bugs that were closed, then later reopened and not yet resolved, slightly inflating the number of resolved bugs; for Equinox this was 4 of the 199 "resolved" bugs reported).
Open: bug_status = unconfirmed, new, assigned, or reopened.
Closed: bug_status = resolved, verified, or closed.

Table 42: Project activity: Forums (newsgroups)

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BIRT	Eclipse Projects >> BIRT	977	275	76371	24755
CDT	Language IDEs >> CDT	1032	350	22981	8100
Equinox	Eclipse Projects >> Equinox	212	75	7983	2472
JDT	Language IDEs >> JDT	650	228	27911	9470
Mylyn	Eclipse Projects >> Mylyn	107	39	4993	1494

All message and topic counts are exact counts from the forums, except for the numbers of messages posted to the BIRT, CDT, and JDT forums between September and November 2010, which are approximated by an extrapolation from the average number of messages per topic for a sample of 40 topics.

Table 43: Project activity: Developer mailing lists

Project	Mailing list name	Description	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BIRT			**18**	**2**	**5**	**25**	**0.3**	**1.22**	**5**
	birt-dev	For all BIRT committers	9	1	3	13	0.1	0.35	2
	birt-report-designer-dev	BIRT subproject committers	3	1	2	6	0.1	0.71	2
	birt-report-engine-dev	BIRT subproject committers	3	0	0	3	0.0	0	1
	birt-charting-dev	BIRT subproject committers	2	0	0	2	0.0	0	1
CDT	cdt-dev	General CDT developers list	**230**	**180**	**144**	**554**	**6.1**	**7.64**	**38**
Equinox			**133**	**71**	**129**	**333**	**3.7**	**3.93**	**15**
	equinox-dev	Equinox technical discussions	58	33	50	141	1.5	2.79	11
	p2-dev	p2 developer discussions	75	38	79	192	2.1	2.35	9
JDT			**18**	**27**	**33**	**78**	**0.9**	**1.23**	**5**
	jdt-dev	General JDT developers list	0	1	5	6	0.1	1.00	3
	jdt-core-dev	JDT core developers list	12	15	14	41	0.5	0.85	4
	jdt-debug-dev	JDT debug developers list	3	6	12	22	0.2	1.27	5
	jdt-ui-dev	JDT user interface developers	3	5	1	9	0.1	0.84	3
Mylyn			**35**	**143**	**138**	**316**	**3.5**	**5.57**	**30**
	mylyn-dev	Mylyn developer discussions	12	46	42	100	1.1	2.04	9
	mylyn-integrators-dev	Integrators adopting Mylyn	3	13	1	17	0.2	3.06	9
	mylyn-reviews-dev	Mylyn reviews project	18	84	95	197	2.2	5.48	29

This table omits inactive mailing lists, mailing list not directly concerning development, and new mailing lists provisioned during the study interval. Omitted lists include birt-news (inactive), birt-pmc (nearly inactive) jdt-apt-dev (inactive), jdt-doc-dev (inactive), mylyn-announce (a duplicate subset of the mylyn-dev list), and four new Mylyn subproject lists provisioned in November 2010: mylyn-versions-dev, mylyn-builds-dev, mylyn-docs, and mylyn-pmc.

All five case projects used all of the project infrastructure components provided the Eclipse Foundation, but each utilized those components in somewhat different ways. CDT, Equinox, and Mylyn were heavy users of the mailing lists for interactive discussion among developers, while BIRT and JDT employed their mailing lists primarily for distributing announcements to developers; interactive discussion was purposefully redirected to the issue tracker, and user questions were redirected to the user forums[126].

[126] Some CDT committers, in contrast, prefer that sophisticated technical user questions be posted to the mailing lists rather than the forums so that the question is more likely to be seen promptly by the developer best qualified to address the query (e.g., cdt-dev mailing list, Blackburn, March 31 2011).

CDT, BIRT, and JDT had the most traffic on user forums; Equinox and Mylyn forums were lower traffic. All projects used the Eclipse Bugzilla issue tracker actively as the main channel for focused discussion on specific technical matters. More so than other projects, the Mylyn project made extensive use of the Bugzilla voting feature; members of the community were encouraged to vote (by clicking a "vote" button) for bugs they most wanted fixed, and committers used this information when prioritizing tasks. As noted, BIRT and JDT used the issue tracker for nearly all developer discussions, while CDT, Equinox, and Mylyn held discussions of broader interest on the mailing lists.

The projects differed widely in their internal organization. CDT and BIRT are flat structures. JDT is partitioned into three subprojects with separate communication channels and leadership for each subproject. Equinox is partitioned into five subprojects but only one – the p2 provisioning framework – is managed under a separate project lead and mailing list; the other four subprojects are managed together under one project lead and mailing list. During the study interval, Mylyn was promoted from a subproject in the Tools PMC to a separate top-level project, and partitioned into eight subprojects, each with its own lead and communication channels. Equinox and Mylyn both include *incubators* – subprojects outside of the main code base intended for exploratory work. Jeff McAffer, co-lead of the Equinox project, explains the rationale for the Equinox incubator subproject:

> *On projects where the core group is very close, incubators are a good way to bring in new people. We take a somewhat laissez-faire approach to new committers in the incubator. It's a place where reasonably competent people can be easily voted in and show their stuff – both their technology and their personal skills – to become a member of the community. Then as the technology and the*

> *people mature, we'll graduate the work out to an appropriate place. It's not always Equinox – it could be some other place.*
>
> *When people show up and want to do new things, we put them in the incubator. It's a low bar for entry, they can't do any damage, and they get a place to prove themselves and get known. There's very little risk.*

This explanation of simultaneously fostering new participation while protecting the core is consistent with the project case data. Of the twelve new committer nominations to the Equinox project between June 2009 and November 2010 (Table 39), eight nominations were exclusively to the incubator subproject rather than the main project code base.

Committers on the case projects typically followed recommended guidelines and best practices, with some local exceptions. An example of a local exception is the decision by members of the BIRT PMC to use version numbers for the 2011 BIRT release that differed from the guidelines of the architecture council *(Version Numbering*, 2009; discussed in subsection 8.3.2). Following extensive discussion and counter-argument in the project and cross-project mailing lists, the BIRT PMC announced their decision[127]:

> *Based on the feedback from the cross-project list, we have discussed this further in the BIRT PMC. While good points were raised, we have concluded that we will go with 3.7 for the BIRT Indigo version number. It is not perfect but we think this is the better choice for our community.*

Consistent with *The Eclipse Way,* committer discussions on all projects showed reverence for published APIs, awareness of the potential for API changes to impact others, and reluctance to add new API without community review.

[127] The rationale for the BIRT version numbers, the counter-arguments raised in favour of the recommended version numbering guidelines by members of the planning council and architecture council are well documented in the archives (birt-dev mailing list, Clenahan, January 26 2011; cross-project-issues-dev mailing list, Williams, January 29 2011; McAffer, January 28 2011; Moir, January 28 2011) as is the outcome quoted here (cross-project-issues-dev mailing list, Clenahan, February 7 2011).

All five projects opted to participate in the annual release train and to be included in Eclipse packages. BIRT, CDT, Mylyn and JDT released only with the release train; Mylyn had two major releases each year – one with the release train and one mid-year.

Each project had its own unique local conventions, which were applied with high consistency within a project, but differed widely between projects. An illustrative example is the different ways in which project circulated information about software builds: four of the five case projects posted notices to the developer mailing lists but the timing, content, and level of scripted automation differed widely. Other local differences included coding conventions, frequency of project meetings (usually by conference call), the management of upcoming agendas and content of published minutes, decision rules for reaching consensus on developer discussions, decision rules for applying patches to source code, the expected minimum requirements for a contributor to earn a committer nomination, and which project communication channel to use for which discussion.

The discussion thus far has emphasized collaboration and cooperation, but there is also much evidence of competition, rivalry, and multiple parallel experiments within the technical domain. Competing solutions to a problem, either within a single project or between multiple projects, are especially common when the "right" solution is unclear[128].

8.7 Synthesis and theory development

The analysis and discussion of project governance rules and the project cases provides the information to extend the conceptual work of section 7.8 by completing the IAD table for individual participation in the Eclipse community (Table 44)[129].

[128] A full explanation requires case evidence and and arguments developed in chapters 9 and 10. Section 11.2 revisits this theme in more detail and enfolds prior literature.

[129] Table 44 is a compilation from sources cited throughout chapter 8 and project case notes documents.

Table 44: The institution of individual participation (March 2011)

IAD Element	Description
Underlying factors	
Rules-in-use	Development of the Eclipse platform occurs within *Eclipse projects*: • Projects share a common working vocabulary and organizational structure. • Project activity is structured by (1) *mandatory requirements* of the *Eclipse Development Process* (EDP), (2) *guidelines and community norms*, (3) opt-in *programs*, and (4) *local conventions* that vary between projects. Four *guiding principles* for developing the EDP: (1) practice the *open source rules of engagement* of *openness* to all participants, *transparency* of artifacts and information, and *meritocracy;* (2) actively develop *communities*; (3) deliberately cultivate an *ecosystem*; (4) be clear, concise, lightweight, and continuously evolve. *Mandatory requirements* for all projects: • Meritocratic committer nomination and election process. • IP management in compliance with the Eclipse *Intellectual Property Policy*: use of Eclipse Public License (EPL); contributors retain copyright; IP due diligence. • Openness and transparency of project activity and decision-making. • Active community building; open and public communication with users. • Participation in the Eclipse Roadmapping process; maintenance of project plans. • A project lifecycle process with distinct phases, incubation and mentoring of new projects, and managed transitions requiring community review and approvals. • Release management requiring community release reviews and approvals. • Committers agree to (1) abide by Eclipse Foundation policies, and (2) participate actively in project activities including discussions on developer mailing lists, questions on user forums, committer votes on project matters, and Bugzilla. Enforcement mechanisms include termination of non-conforming projects, removal of project leadership, rejection of IP, and unfavourable outcomes of project reviews. *Guidelines and community norms*: *The Eclipse Way*; guidelines for release numbering, community building, builds, etc.; reverence for committer elections, areas of expertise, expectations for conversation, maintaining APIs, etc. *Opt-in programs*: the *annual release train*; inclusion with *Eclipse packages*. *Local conventions*: coding conventions, use of communication channels, choice of build systems, expectations for committer nominations, decision rules, etc.
Community attributes	Technically-skilled users and developers of Eclipse software who are interested in technical problems and solutions. Contributions are typically related to paid work.
Resource attributes	*Artifacts*: open source software tools, frameworks, and runtimes on a modular plug-in architecture, distributed as source and packages; complementary assets. *Facilities*: an Eclipse project IT infrastructure including source code repository, developer mailing lists, user forums, Bugzilla issue tracker, and other assets.
Action arena	
Actors	Individuals with heterogeneous intrinsic and extrinsic motivations. • *Contributors* who contribute code, fixes, tests, documentation, and other work. • *Committers* with write access to project resources; expectations for contribution. Most committers are paid in some way to commit to Eclipse.
Action situation	Participation in the Eclipse developer community.
Interaction patterns	Virtual and technology-mediated; frequent; mainly asynchronous and persistent.
Outcomes	Sustained production of high quality software tools, frameworks, and runtimes.
Evaluative criteria	Evidence of (1) a thriving, diverse, and active *developer* community, (2) an active and engaged *user* community, and (3) an active and engaged *adopter* community.

The following discussion begins at the right of the IAD framework (chapter 5, Figure 2) at *outcomes* and works systematically through the *action arena* to the *rules-in-use* (approximately bottom-to-top of Table 44) – a typical order for IAD studies seeking to understand and explain observed outcomes (section 5.1; Hess & Ostrom, 2007).

The action situation of interest is *participation in the Eclipse developer community*: that is, the decisions taken by individual contributors and committers to participate in activities to maintain and extend the Eclipse platform. Interpreted through the lens of the IAD framework and the conceptual model of section 7.8, the Eclipse platform is an open source software commons freely available to all (section 7.8.4), and participation in the developer community is a form of collective action (section 5.1).

Activity within the Eclipse developer community is structured into approximately 120 active Eclipse projects (sections 8.1 and 8.5) – "the main operational unit at Eclipse" (EDP 2.5, section 4, p. 5). The EDP is explicit about evaluative criteria and desired outcomes: for each Eclipse project, the desired outcomes are (1) a thriving, diverse, and active developer community (i.e., *this* community), (2) an active and engaged user community as proof that the project's tools are useful and needed, and (3) an active and engaged adopter community as proof that the project's software is extensible and accessible (section 7.5; EDP 2.5, section 2.3, p. 3). Each project reports evaluative criteria to the community at each project review, held at least annually (section 8.2.6) and more often if the project releases more frequently (section 8.2.7).

The actors are of two types: individual *contributors* who contribute code, fixes, tests, documentation, and other work, and individual *committers* who can write to project resources, review and approve contributor contributions, and sign a committer agreement to participate actively and to comply with Eclipse Foundation policies (subsection 8.1.3; Table 30). Projects are structured as meritocracies (subsection 7.5.1), where the breadth of an individual's influence corresponds to the breadth of their contribution. Through a well-defined committer nomination and election process (subsection 8.2.1), existing committers can invite a meritorious contributor to become a committer.

Most interactions between participants are virtual rather than physical, and mediated by collaboration technologies such as emails posted to mailing lists, the Bugzilla issue tracker, wikis, blogs, and conference calls. Consistent with the research literature on participant motivation in open source software projects (section 3.6; e.g., Ghosh et al. 2002; Hertel et al. 2003; Ghosh, 2005; Lakhani & Wolf, 2005; Roberts et al. 2006; Subramanyam & Xia, 2008), the case data suggest heterogeneous motivations for Eclipse committers. Some interview respondents, especially long-time community leaders, strongly identified with the Eclipse community; a few respondents reported changing companies more than once in order to continue working on particular Eclipse projects. Some respondents worked primarily on specific features for an employer or for customers (discussed further in chapter 10); others reported minimal direction from an employer other than advancing the goals of particular projects.

The rules-in-use structuring individual participation can be usefully organized into four categories. The first category is *mandatory requirements* for all Eclipse projects (section 8.2), specified in the *Eclipse Development Process* and supporting documents. Motivated by the Eclipse Foundation purpose (*Bylaws*, 2003, 2008, article 1.1; section

7.1) and guiding principles (EDP 2.5, section 2, pp. 2-4; subsection 8.1.2), the requirements are said to be imposed on behalf of the Eclipse Foundation membership with the intent to make the Eclipse platform software produced by the projects more attractive to the user, adopter and developer communities, and to the companies of Eclipse ecosystem. The project requirements include a common meritocratic process for nomination and election of committers (subsection 8.2.1), rules for intellectual property management (subsection 8.2.2), minimum standards of openness and transparency (subsection 8.2.3), obligations of community-building (subsection 8.2.4), participation in an Eclipse roadmapping process to provide guidance to the community and ecosystem (subsection 8.2.5), a project lifecycle process with incubation and mentoring of new projects and community review of transitions between between phases (subsection 8.2.6), and rules for release management (subsection 8.2.7). The Executive Director and the Eclipse Management Organization can enforce these requirements through four mechanisms: termination of a non-conforming project, removal of project leadership, unfavourable outcomes of the IP due diligence process, and unfavourable outcomes of project reviews (subsection 8.2.8). During the study period, each case project conformed to mandatory requirements without evidence of enforcement.

The second category is *guidelines and community norms* (section 8.3). Guidelines include recommendations, best practices, checklists, and other resources, that are made available to the development teams of Eclipse Projects by the Eclipse Foundation staff, councils, and members of the Eclipse community. The release management practices of *The Eclipse Way* (subsection 8.3.1) include a project rhythm of

6-week milestones, continuous integration with nightly, weekly, and milestone builds, and automated testing – first codified by Gamma & Wiegand (2005) and continuing to evolve in the Eclipse wiki. Adherence to guidelines is itself a community norm: there are normative pressures to conform, but projects may make different choices with no formal penalty. Other community norms discovered through direct observation of the similarities between the case projects (section 8.6) include reverence for the committer nomination and election process, attention to APIs, respect for the areas of expertise of other committers, and expectations for early and ongoing conversation.

The third category, *opt-in programs* (section 8.4), comprises voluntary processes that entail additional obligations. The most prominent example is participation in the *annual release train* (subsection 8.4.1) – a coordinated software release available to projects that commit to a predictable release engineering schedule and other requirements that include many of the practices of *The Eclipse Way*. Projects that participate in the release train are eligible for inclusion in *Eclipse packages* (subsection 8.4.2) – tested configurations of Eclipse plug-ins addressing the needs of particular target user groups, such as Java developers, C++ developers, or enterprise report designers.

The fourth category, *local conventions*, was examined by identifying differences between the five Eclipse case projects (section 8.6). Case projects varied widely along several dimensions. Top-level projects, through the creation of a project charter, define their own decision-making conventions (within the constraints of the EDP) and levels of trust expected for committer nominations. Projects establish coding conventions, patterns of engagement, and purposing of communication channels. Outside the scope of mandatory rules, guidelines and community norms, and participation in optional programs, projects display considerable diversity in local operational rules.

For decision-making on project matters, the central notion across all projects is the open and transparent committer vote (EDP 2.5, section 4.7, p. 8):

> *The decision making process of the Project relies on active committers who respond to discussions and vote in a constructive and timely manner... Committers are required to track, participate in, and vote on, relevant discussions in their associated Projects and components. There are three voting responses: +1 (yes), -1 (no, or veto), and 0 (abstain).*

However, projects establish their own local conventions for when and how to call for a vote, and how to interpret results. Within the community-driven case projects, much of the decision-making happened on developer mailing lists and Bugzilla bug reports. Except for committer elections, calls for formal project-wide votes were relatively rare. More common was a tacit norm often called "silent consensus" (Fogel, 2006) or "lazy consensus" (Gardler & Hanganu, 2011) in the open source practitioner literature. With silent consensus, agreement is assumed unless an objection is raised. Typically, a committer posts an explanation of an issue or opportunity, with some analysis of possible options, and a description of the action they propose to take if no one comments or objects. If a few days pass with no comments or objections, the committer moves ahead with the work, and replies back to their original post with updates or results.

This chapter examined activity at the project level of the Eclipse field setting in order to specify the IAD characteristics of the institution of individual participation. The next chapter is a similar examination of governance-level activity within the various polycentric governance structures of Eclipse.

9 Foundation governance

This chapter examines the polycentric governance structures that shape the operating rules of the community and ecosystem institutions. In the institutional conceptual framework of section 7.8, the Eclipse Foundation is a hybrid of two management constructs: it is both the *keystone organization* of the Eclipse ecosystem, and the *open source software foundation* providing governance and a legal identity for the Eclipse community. This chapter begins by examining the rules specified in Eclipse governance documents, then examines participant behaviour through direct observation and archival records, triangulated with insights from participant interviews. From these observations, it develops an institutional representation of Eclipse governance using the conceptual apparatus of the IAD framework.

Business ecosystem governance and open source community governance are each complex topics, and the deliberate placing of both functions within one organizational entity introduces potential for new interactions and complexity. As a tactic to manage the scope of data collection, analysis, and presentation of results, each governance structure is examined within a separate section focusing on the answers to the governance case study questions identified in section 6.2.4 – topics of membership, scope of influence, changes to rules, participant interactions, and impact on other institutional structures – and the information needed to complete the governance table shells (Table 16) developed in subsection 6.2.2. Section 9.9 examines how the Eclipse governance structures have changed over time, and section 9.10 examines the interactions between groups.

9.1 Governance documents

The governance documents that specify the formal structure of the Eclipse Foundation provide a useful starting point for data collection and analysis. The principle governance document is the *Bylaws of the Eclipse Foundation* (2003, 2008). The *Bylaws* make reference to several other governance and operating documents for additional rules and policies. These supporting documents include the Eclipse *Membership Agreement* for the membership dues, powers, and obligations of each membership class, the Eclipse *Intellectual Property Policy* regarding software code contributions, the *Antitrust Compliance Policy* governing member competitive behaviour, the *Eclipse Development Process* (EDP 1.0, 2003; EDP 2.3.1, 2006; EDP 2.4, 2008; EDP 2.5, 2010) for how the various parts of Eclipse "lead, influence, and collaborate with Eclipse Projects" (EDP 2.5, section 1, p. 2), and the annual *Eclipse Roadmap* and *Release Plan*. The rules defined in these supporting documents are more easily amended than the rules of the *Bylaws*. Amendments to the *Bylaws* require consent of a super-majority (at least two-thirds of the members in good standing) of the board of directors, and consent of at least two-thirds of the membership at-large in a membership vote. The original *Bylaws*, completed in November 2003 for the launch of the Eclipse Foundation in January 2004, have been amended only once, in July 2008, primarily to revise the Eclipse Foundation membership classes[130]. In March 2011, the Executive Director began community consultations for a second possible amendment, with target for completion in summer 2011. At the time of

[130] Table 22 in section 7.1 summarized the six Eclipse Foundation membership classes of the 2008 Bylaws. Amendments to the Bylaws in 2008 included (1) creation of a new *enterprise member* membership class, (2) renaming the *add-in provider* class to *solutions member* to better reflect the commercial offers provided by the companies in the membership class, and (3) fix several minor errors and inconsistencies in current Bylaws (*Proposed Membership Changes*, 2008).

this writing, those discussions are still underway[131]. Amendments to supporting documents typically require only a simple majority or super-majority of the Board of Directors without requiring a vote of the membership. For example, the *Eclipse Development Process* was amended in 2007 (EDP 2.31), 2008 (EDP 2.4), and 2010 (EDP 2.5), and discussions are currently underway about possible amendments in 2011[132].

As defined in the *Bylaws*, the *Eclipse Foundation* is a membership-based not-for-profit trade association with a *board of directors*, an *Executive Director* reporting to the board, a *professional staff* reporting to the Executive Director, three councils (an *architecture council*, a *planning council*, and a *requirements council*), and a *membership at-large* with multiple classes of membership. The staff and councils together comprise the *Eclipse Management Organization (EMO)*, under the direction and leadership of the Executive Director. Each governance structure is examined in its own section.

9.2 Board of directors

The Eclipse Foundation *board of directors* directs the the business and technical affairs of the Foundation (*Bylaws*, section 3.1), including appointment and oversight of an Executive Director (*Bylaws*, section 5.3). Article 3 of the *Bylaws* specifies the composition, power and duties, and decision-making process of the board. The board is comprised of both appointed and elected directors according to well-specified rules of composition (*Bylaws*, section 3.5). The three categories of director are (1) *strategic*

[131] Discussions of the proposed amendments began with presentations by the Executive Director to the Eclipse councils, board of directors, and the Eclipse membership at EclipseCon 2011 (e.g., eclipse.org-membership-at-large mailing list, Smith, March 20 2011; *Abridged minutes of the Eclipse Foundation board of directors*, 2011 Q1, March 21; *Executive Director Report*, 2011 Q1) and continue through public communication channels (e.g., Bugzilla #342328).

[132] Bugzilla #342328. See also subsection 8.1.5 for discussion of past revisions to the EDP and section 9.9 to situate these revisions with respect to other governance changes.

member directors appointed by the strategic member companies of the Eclipse Foundation, (2) *sustaining member directors* elected annually by the enterprise member companies and solutions member companies using a single transferable vote process, and (3) *committer member directors* elected annually at membership-at-large elections. The number of strategic member directors is determined by the number of strategic members of the Eclipse Foundation: each strategic member appoints one director. The number of sustaining member and committer member directors is determined by a formula: one seat plus an additional seat for every additional five seats beyond one allocated to strategic members. Elected members serve one-year terms with no restrictions on renewal. Associate members of the Eclipse Foundation are not represented on the board of directors. In March 2011, there were 18 directors on the Eclipse Foundation board of directors: 12 directors appointed by strategic members, and 6 elected directors (3 sustaining member representatives and 3 committer member representatives).

Most board actions are approved by a *simple majority* of attending members, with each director receiving one vote on each matter submitted to a vote of the board. Examples of actions requiring a simple majority assent of the board include creating new top-level projects and working groups, approving an annual roadmap (from the process discussed in subsection 8.2.5 and section 9.9), approving an annual budget for the Eclipse Foundation, approving the promotion plans of the EMO or any committees, and passing resolutions that instruct the Executive Director to take specific actions. Approval for some specific actions (listed explicitly in the *Bylaws*, section 3.9) requires greater support – either super-majority consent or unanimous consent. *Super-majority consent* of at least two-thirds of the board is required for actions amending governance documents, selecting or terminating the Executive Director, terminating a member's agreement, selecting standards organizations through which to standardize Eclipse APIs, and entering into formal affiliations with other organizations. *Unanimous consent* is required for actions amending the EPL or regarding the use of a license other than the EPL.

The board meets quarterly at face-to-face meetings, with regular monthly conference calls between quarterly meetings. Each year, the Q1 board meeting is co-located with EclipseCon – the annual face-to-face event of the Eclipse community. The Q1 meeting closes out the previous year, welcomes new committer and sustaining members from the annual membership elections, and (in all years except 2008 and 2009) approves the roadmap. Since 2005, the Q2 meeting has been a two-day event with a strategy discussion to establish strategic goals and propose initiatives to achieve those goals. In some years, the Q4 meeting has also been a two-day event. The Q4 meeting typically includes resolutions to re-appoint board offices and the Executive Director for the following year, and approve the next year's budget.

Board meetings operate under the Chatham House rule[133]. Section 11.7 of the *Bylaws* requires that "abridged versions of any and all minutes from Board and Membership Meetings" be posted promptly to the Eclipse website[134]. Because many

[133] The 2002 refinement of the Chatham House Rule reads as follows: "When a meeting, or part thereof, is held under the Chatham House Rule, participants are free to use the information received, but neither the identity nor the affiliation of the speaker(s), nor that of any other participant, may be revealed" (http://chathamhouse.org.uk). Confidentially of board discussions was discussed explicitly at the 2007 Q2 meeting of the board (June 20-21): "The understanding would be that no Board member would ever attribute a statement to someone else."

[134] The full minutes are not publicly available. However, the abridged minutes are said to comprise everything in the full minutes except for personal information falling under the Chartham House Rule and detailed financial information. For example, the abridged minutes indicate the outcome of all votes

matters of interest for this research require board confirmation or other board involvement as a matter of course, abridged minutes of board meetings were a useful data source at all levels of analysis of this research – particularly given that board meetings were one of few areas of Eclipse that were closed to direct observation[135].

Data collection and analysis included coding of board resolutions and discussions to identify common themes of activity, establish timelines of events, and to inform cases at other levels of analysis following the method described in subsections 6.2.4 and 6.2.8. Quarterly meetings agendas typically included the following five topics.

1. *General business.* Board meetings open with general business, typically including resolutions to approve full and abridged minutes of previous meetings, discussion of issues and corrections to previous minutes, changes in board membership and other matters impacting the current meeting, and the schedule of upcoming meetings.

2. *Committer and sustaining member issues.* Each meeting explicitly includes time for committer member representatives and sustaining member representatives to bring forward to the board issues from their constituents, and for other board members to raise issues or pose questions for the representatives to take back to their constituents.

3. *Management reports.* Each quarterly meeting includes presentations of management reports and subsequent discussion, typically including reports on membership, marketing, projects, operations, and (since 2007) key performance indicators (KPI) of the EMO. The reports are delivered by the Executive Director, the

(unanimous, super-majority, majority, etc.), but not voting details of how each director voted.

[135] The other noteworthy area of Eclipse that is closed to direct observation is the IP due diligence process, discussed in subsections 8.2.2 and 9.4.2.

Treasurer, or other individuals "present at the invitation of the board" such as Eclipse Foundation staff. Membership, marketing, and project reports are typically included with the abridged minutes; operations and KPI reports are not.

4. *Other business* may include proposals for new policy resolutions, changes to existing Eclipse policies, approval of the annual roadmap (typically in Q1), renewal of board officers (in Q4), new programs, preparation and planning for upcoming events, exceptions to the IP policy, creation of new projects or working groups, amendments to project charters, activities of board committees, presentations of commissioned research reports on the Eclipse community and ecosystem, and changes to governance documents.

5. *Invited speakers*. Some meetings include special presentations from invited guest speakers on topics of interest to board members. Past speakers have included software industry analysts and academic researchers.

Matters that receive frequent attention from the board of directors include intellectual property, growing the Eclipse membership, promoting Eclipse through marketing and public relations (including EclipseCon and other events), Eclipse facilities and IT infrastructure, and Eclipse projects.

Table 45 summarizes the operating rules for individuals serving on the Eclipse Foundation board of directors.

Table 45: Operating rules of the board of directors (March 2011)

	Board of directors
Membership	The board is comprised of *appointed* and *elected* directors in three categories: • Strategic member directors *appointed* by the strategic member companies of the Eclipse Foundation; each strategic member appoints one director. • Sustaining member directors *elected* annually by the enterprise member companies and solutions member companies using a single transferable vote process. • Committer member directors *elected* annually using a single transferable vote process at annual membership-at-large elections. Associate members are not represented on the board of directors. The number of sustaining member and committer member directors is determined by a formula: one seat plus an additional seat for every additional five seats beyond one allocated to strategic members. Elected members serve one-year terms with no restrictions on renewal. In March 2011, there were 18 directors: 12 strategic member directors (appointed), 3 sustaining member directors (elected), and 3 committer member directors (elected).
Scope of authority	The board directs the business and technical affairs of the Eclipse Foundation.
Actions requiring simple majority consent	Board actions are normally approved by a simple majority of members in good standing. Examples include the following: • Creating new top-level projects and PMCs (nominated by the Executive Director). • Approving a release roadmap. • Approving an annual budget for the Eclipse Foundation. • Approving the Executive Director's compensation package. • Appointing board committees and confirming appointments to standing committees. • Approving promotion plans of the EMO or any membership-at-large committee. • Instructing the Executive Director to take specific actions at the behest of the board.
Actions requiring super-majority consent	Some specific actions require super-majority consent of no less than two-thirds of the members in good standing. • Amending governance documents, including the *Bylaws of the Eclipse Foundation,* the *Membership Agreement*, or the *Certificate of Incorporation*. • Approving changes to governance documents including the *Eclipse Development Process* (EDP), the Eclipse Foundation *Antitrust Policy*, or the *IP Policy*. • Approving changes to the membership contribution requirements. • Approving the appointment of a Project Management Committee lead when, at the time of approval, the appointment of that lead would result in more than fifty percent of the PMC leads being affiliates of the same organization. • Terminating a member's agreement. • Approving or changing the name of the Eclipse Foundation or the Eclipse Platform. • Selecting standards organizations through which to standardize Eclipse APIs. • Selecting or terminating the executive director. • Selecting outside legal council. • Entering into any formal affiliation with another organization.
Actions requiring unanimous consent	Unanimous consent is required for actions amending the terms of the EPL or actions regarding the use of a contribution or distribution license other than the EPL.
Changes to the operating rules	The rules governing the board of directors are specified in the *Bylaws of the Eclipse Foundation*. Amendments to the *Bylaws* require consent of at least two-thirds of the membership at-large, and may require unanimous consent under some circumstances.

Compiled from sources attributed in section 9.2, particularly the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Eclipse Development Process* (EDP 2.5, 2010), and board resolutions documented in the *Abridged minutes of the Eclipse Foundation board of directors*.

9.3 Executive Director

The Eclipse Foundation *Executive Director* (ED) manages the business affairs of the Eclipse Foundation on a day-to-day basis, provides periodic reports to the board of directors and to the membership at-large, and leads the Eclipse Management Organization (*Bylaws*, section 5.3a, section 7.1).

The formal powers invested in the Executive Director are broad. The Executive Director (or a delegate acting on behalf of the ED)[136] is the arbiter of community consensus in matters such as project release reviews (subsection 8.2.7) and other community project reviews (subsections 8.2.6) (EDP 2.5, section 6.3, p. 15):

> *The EMO(ED) approves or fails the review based on the public comments, the scope of the project, and the purposes of the Eclipse Foundation as defined in the Bylaws.*

The Executive Director selects the chairs of the Eclipse councils, nominates PMC leads (for confirmation by the board), approves new PMC members (elected by the existing PMC), approves new project leads (elected by the project's committers and approved by the PMC), appoints individuals to the Eclipse councils, hires and manages Eclipse Foundation staff, and represents the interests of the Eclipse Foundation on external standards organizations and industry working groups. Together with the board of directors, the Executive Director determines which new projects will be accepted at Eclipse[137]. For existing projects, the ED has a range of possible actions available to enforce conformance to the *Eclipse Development Process* (examined in subsection 8.2.8). Concentration of formal power within one individual was one of three concerns cited by

[136] In Eclipse governance documents, including the Bylaws and EDP, the label "EMO(ED)" refers to an individual or group to which the Executive Director has delegates some specific powers of the ED office.

[137] The EMO(ED) decides the outcomes of creation reviews, which are required to create a new project. Board approval is required to create a new top-level project and PMC (section 9.2).

Sun Microsystems in their open letter of January 2004 explaining their reasons for not joining Eclipse[138]:

> *The organization's bylaws have given the director an unusual amount of power to form projects and assign resources. Will the director be an impartial guardian of the community (or be partial)?*

Nonetheless, there are strong structural incentives for the Executive Director to build consensus on issues and to act of behalf of the Eclipse membership. In particular, the ED operates under the oversight of the board of directors, on a one-year term appointment renewable each December for the following year with a successful action by the board. The board also establishes ED compensation. The Eclipse Foundation has had only one Executive Director in its history, Mike Milinkovich, announced in June 2004 (Eclipse Foundation press release, June 1 2004) four months after the formation of the first Eclipse Foundation Board of Directors (Eclipse Foundation press release, March 3 2004), and renewed each year, typically with unanimous support.

As a requirement of the *Bylaws* (section 5.3c), the ED produces quarterly *Executive Director Reports*, available back to October 2004. Since 2005 Q1, the reports have followed a standard format including welcome of new Eclipse Foundation members, a membership update, download statistics and website analytics about Eclipse software, financial information, and upcoming activities for the coming quarter. The Q1 report coincides with the Eclipse Annual General Membership Meeting (co-located with

[138] Sun's open letter to Eclipse Membership was published publicly on January 30 2004 (available online: http://today.java.net/pub/a/today/2004/01/30/letter.html). O'Mahony et al. (2005b) provides context and insightful commentary. On January 27 2010, Sun Microsystems was acquired by Oracle Corporation, a strategic developer member of the Eclipse Foundation, and one of the most active companies employing committers on Eclipse projects. In May 2011, Oracle ranked second in Eclipse Dashboard reports in the metric of number of active committers employed; only IBM employed more active Eclipse committers.

EclipseCon), which also includes results of the board of directors elections, and other presentations of interest to the membership[139].

Table 48 summarizes the operating rules for the Eclipse Executive Director.

Table 46: Operating rules of the Executive Director (March 2011)

	Executive Director
Membership	The Executive Director is an individual appointed by the board of directors on a one-year renewable term. The Executive Director may not be an employee, officer, director or consultant of any member of the Eclipse Foundation. Selecting or terminating an Executive Director is a board action requiring super-majority (two-thirds) consent of the Board.
Scope of authority	The Executive Director manages the day-to-day business affairs of the Eclipse Foundation, reporting to the board of directors, and subject to oversight by the board.
Duties to the Membership	• Preside over Membership Meetings, including an Annual General Meetings held each year, and special meetings called by the membership. • Provide quarterly *Executive Director Reports* including project status, financial information, membership information, and other material information. • Provide written *Update Reports* to the membership regarding material developments.
Duties to the Board	• Execute contracts, agreements, membership certificates, and other instruments, upon approval and direction from the board. • Report to the board all matters affecting the Eclipse Foundation that should be brought to the board. • Perform other duties assigned by the board.
Powers	• Hire other employees as deemed appropriate. • Organize and lead Eclipse platform development according to the *Eclipse Development Process.* • Select the chairs of the architecture, planning, and requirements councils. • Nominate project management committees (PMCs) and their leaders (for confirmation by the board of directors). • Approve new project leads (elected by project committers).
Changes to the operating rules	Operating rules that are specified in the *Bylaws* can be changed only by amendment of the *Bylaws*. Within the scope of the *Bylaws*, the board of directors has considerable discretion over directing the duties of the Executive Director.

Compiled from sources attributed in section 9.3, particularly the *Bylaws of the Eclipse Foundation* (2003, 2008) and the *Eclipse Development Process* (EDP 2.5, 2010).

[139] These data were coded in the same manner as the board of directors minutes to identify timelines of events and to inform cases at all levels of analysis following the method of subsections 6.2.4 and 6.2.8.

9.4 Foundation staff

The Eclipse *Bylaws* empower the Executive Director to "hire other employees as deemed appropriate" (*Bylaws*, section 5.3a) to form an Eclipse Management Organization under the direction the Executive Director (*Bylaws*, section 7.1). As of March 2011, there were twelve full-time staff providing services in four areas: (1) IT infrastructure, (2) intellectual property management, (3) development process, and (4) ecosystem development (Eclipse.org website, accessed March 30 2011; Skerrett & Beaton, 2010). Each area is described below.

9.4.1 IT infrastructure

From the Eclipse.org website (accessed March 30 2011):

> *The Eclipse Foundation manages the IT infrastructure for the Eclipse open source community, including CVS/SVN code repositories, Bugzilla databases, development oriented mailing lists and newsgroups, download site and web site. The infrastructure is designed to provide reliable and scalable service for the committers developing the Eclipse technology and the consumers who use the technology.*

The Eclipse IT infrastructure includes an extensive and modern open source development infrastructure for Eclipse Foundation projects, communication services for the Eclipse community such as the *Planet Eclipse* blog aggregator, the *MyFoundation Portal* for Eclipse member services, and the Eclipse.org website. In the framework of Hess & Ostrom (2003), these IT resources comprise the *facilities* component of the Eclipse commons (subsection 7.8.4), .

The Eclipse open source development infrastructure includes three choices of source code repository[140], a customized and branded instance of the Bugzilla issue

[140] A source code repository, also called a source code management (SCM) system, is a software application that stores software revisions and allows many developers to work on a project at the same time. In March 2011, the three source code repositories supported in Eclipse were CVS, Subversion, and git.

tracking system (*Eclipse Bugzilla*)[141], developer email lists, online user forums and newsgroups, and a wiki (*Eclipsepedia*)[142] (see also project resources in subsection 8.1.4).

As noted in section 9.2, enhancements to the Eclipse IT infrastructure is a frequent discussion topic at the board of directors. *Eclipse Labs* is an example of a recent new capability to emerge from those discussions. From the Eclipse Labs announcement (Milinkovich blog, May 13 2010):

> *In the open source world, there are two options if you want to start an Eclipse oriented project:*
>
> *1) propose a project with the Eclipse Foundation or*
> *2) start a project on one of the existing forges, ex. Google Code, SourceForge, Codehaus, etc.*
>
> *For some projects, the IP due diligence and development process expected of Eclipse projects is not warranted. However, creating an Eclipse project on a forge makes it difficult to gain visibility in the Eclipse community.*

Eclipse Labs provides a third alternative for hosting open source projects related to Eclipse[143]. Open source projects hosted at Eclipse Labs are not Eclipse Foundation projects and cannot be called "Eclipse Projects", but they do get greater visibility through the Eclipse Marketplace, and some may choose to "graduate" to Eclipse projects through a project proposal and a creation review (Milinkovich blog, May 13 2010):

> *We need to make it as easy as possible for someone to open source their awesome Eclipse based technology. Not all projects need to be hosted at the Eclipse Foundation and in fact I am hoping more projects will start at Eclipse Labs and then, if they choose, graduate to the Eclipse Foundation.*

[141] An issue tracking system (or bug tracker) is a software application that manages problem reports and enhancement requests (collectively called "bug reports"), and the development activities related to resolving those bug reports. Bugzilla is an open source bug tracker developed by the Mozilla project.

[142] Eclipsepedia is a branded instance of MediaWiki, an open source wiki application developed by the Wikimedia Foundation (http://wikimediafoundation.org).

[143] Eclipse Labs is a customized instance of Google Code Project Hosting, hosted by Google (http://code.google.com/a/eclipselabs.org/hosting). It provides infrastructure services including code repositories, bug tracking, and a project wiki. Projects are expected to be Eclipse plug-ins or built with Eclipse technology, and are thus restricted to a set of open source licenses compatible with the EPL.

Five months after launch, there were more than a thousand projects hosted at Eclipse Labs (Skerrett blog, October 13 2010).

9.4.2 Intellectual property management

Eclipse Foundation legal staff manage the Eclipse IP policy, the IP due diligence process for contributions to Eclipse projects, and other legal policies and services to project members and adopters. From the Eclipse.org website (accessed March 30 2011):

> *This process includes analyzing selected code contributions to try to ascertain the provenance of the code, and license compatibility with the EPL. Contributions that contain code licensed under licenses not compatible with the EPL are intended to be screened out through this approval process and thus not added to an Eclipse project. The end result is a level of confidence that Eclipse open source projects release technology that can be safely distributed in commercial products.*

According the Eclipse Director of Intellectual Property (Campbell, 2007), the process to clarify the IP trail of an open source package has two investigative steps. The first step looks at the code and the project documentation for possible nesting (many open source packages contain other open source packages as nested components). The second step examines each nested package from three perspectives: the *provenance* of who wrote the code and how the rights were obtained, *license compatibility* of whether or not the package license is compatible with the distribution license (usually EPL), and a *risk assessment* of any potential complications. The process is partly manual and partly automated by specialized software tools (Campbell, 2007, p. 16):

> *The Eclipse Foundation uses a variety of tools to scan the source code of the code that will be distributed in either source or binary form. These tools scan the code for keywords related to a number of areas such as words with legal significance, open source license references, copyright references, and inappropriate language, ... [and] look for code that may have been copied from another open source project.*

Eclipse IPzilla is a hosted instance of Bugzilla, available exclusively to Eclipse committers and strategic and enterprise members, for tracking IP due diligence requests and approvals for Eclipse code contributions[144].

9.4.3 Development process

Eclipse Foundation staff provide services and support to Eclipse projects to implement and automate some aspects of the *Eclipse Development Process*, including organizing project reviews and automating aspects of the development processes, such as the process of creating and maintaining the project plans required from each Eclipse project to support the Eclipse roadmap (subsection 8.2.5).

9.4.4 Ecosystem development

"A unique aspect of the Eclipse community and the role of the Eclipse Foundation is the active marketing and promotion of Eclipse projects and wider Eclipse ecosystem" (Eclipse.org website, accessed March 30 2011).

The Eclipse Foundation websites include the *Eclipse.org* home page with community news and press releases, the *Eclipse Live* portal for Eclipse-related videos, webinars, podcasts and other member-provided multimedia content, and the *Eclipse Marketplace* – a catalog of Eclipse-based plug-in extensions and solutions, receiving an average of 200k visitors and 760k page views per month in 2010. The Eclipse Foundation commissioned and maintains the *Eclipse Marketplace Client* (MPC), an Eclipse plug-in that provides an interface within Eclipse tools for browsing and installing

[144] IPZilla (http://dev.eclipse.org/ipzilla) – the tool for tracking Eclipse IP due diligence requests – is the second area of Eclipse that is closed to direct observation; the other noteworthy area is the meetings of the board of directors, discussed in section 9.2 (see also footnote 135). Access to the information contained in IPzilla is said to be a valuable benefit of the strategic and enterprise membership classes.

software from the Eclipse Marketplace. The MPC plug-in is distributed in all Eclipse 3.6 packages (except classic) to give visibility to plug-ins created by Eclipse Foundation members and to make those plug-ins easier for users to locate and install.

The Eclipse Foundation staff provide various optional marketing services to Eclipse Foundation members. These "opt-in members programs" are said to create value for members by driving web traffic to member websites and organizing face-to-face regional events (Skerrett & Beaton, 2010). An example of a program to drive traffic to member websites is the *Members Distro Downloads Program*, which directs users seeking Eclipse software to the download sites of participating companies[145]. Other examples are premium Eclipse marketing programs providing promoted downloads, promoted MPC solutions, and advertising on Eclipse.org. An example of face-to-face regional events is the *Eclipse Training Series* of instructor-led courses on Eclipse-related topics conducted by the staff of Eclipse member companies[146]. The Fall 2008 series included classroom events in 31 countries (Eclipse Foundation press release, September 9 2008). More recently, a greater proportion of training series events are offered online to participants in multiple countries, with fewer regional classroom events.

Table 47 summarizes the operating rules for the Eclipse Foundation staff. According to titles and job descriptions at the Eclipse.org website (accessed December 2010), six staff members primarily support ecosystem development, three support IT infrastructure, two support IP management, and one supports the development process.

[145] Eclipse members distro downloads program website: http://www.eclipse.org/membership/special_programs/member-downloads-program.php

[146] Eclipse training series website: http://www.eclipse.org/community/training/classes.php

Table 47: Summary of operating rules of the Eclipse Foundation staff (March 2011)

	Eclipse Foundation Staff
Membership	The Executive Director may hire other employees as deemed appropriate.
Scope of authority	Eclipse Foundation staff are part of the Eclipse Management Organization (EMO) under the direction the Executive Director, who may delegate authority to staff. The Eclipse Foundation staff provides four areas of services to the Eclipse community: 1) IT infrastructure, including o the Eclipse.org website o infrastructure for Eclipse projects (source code repositories, the Eclipse Bugzilla issue tracker, developer mailing lists, forums and newsgroups, the Eclipse wiki) o release train infrastructure (e.g., build servers and services) o member infrastructure (e.g., MyFoundation portal) o community communication infrastructure (e.g., the *Planet Eclipse* blog aggregator, blog hosting services) 2) Intellectual property (IP) management o the Eclipse IP due diligence process (automated and manual actions by Eclipse legal staff to ensure the pedigree of the IP contained in Eclipse projects). 3) Development process o organization and management of project reviews o automation of project planning, administration and reporting tasks 4) Ecosystem development o marketing services that drive web traffic to member websites o organization of face-to-face regional events, including *EclipseCon*, *EclipseCon Europe*, regional Eclipse *Demo Camps* and Eclipse *Stammtisch* (Mueller blog, January 14 2008), and the *Eclipse Training Series* o activities to grow the Eclipse Foundation membership
Changes to the operating rules	Eclipse Foundation staff report to the Executive Director.

Compiled from sources attributed in section 9.4, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Eclipse Development Process* (EDP 2.5, 2010), the Eclipse.org website, and presentations by Eclipse Foundation staff (Campbell, 2009; Skerrett & Beaton, 2010; Skerrett, 2011).

9.5 Architecture council

According to the *Bylaws of the Eclipse Foundation* (2003, 2008, article 7), the Eclipse architecture council (AC) is responsible for the development, articulation, and maintenance of the Eclipse platform architecture. Its membership is comprised of an individual appointed by each project management committee (PMC), individuals appointed by Eclipse Foundation strategic members, and individuals appointed by the Executive Director.

In March 2011, there were 51 AC members: 36 appointed by the Executive

Director, 8 appointed by strategic members, and 7 appointed by PMCs. There were 26 Eclipse member organizations employing AC members: IBM employed 12 AC members, seven organizations employed either two, three or four members, and 17 organizations employed one member. Three AC members were not associated with a member company. Appointment to the architecture council is one of the highest status positions in the Eclipse meritocracy (section 7.5): "a highly visible confirmation of the Eclipse community's respect for the candidate's technical vision, good judgement, software development skills, past and future contributions to Eclipse" (EDP 2.3.1, section 4.4).

Initially in 2004, the primary activity of the architecture council was the creation of an architecture plan "providing an explicit description of the architecture, … protecting the architecture from inadvertent corruption... [and] plotting the evolution of the Architecture in response to the Purposes and Roadmap" (EDP 1.0, section 1). In 2007, the primary activity changed from creation of an architecture plan to mentorship of new projects (EDP 2.3.1, section 4.4). In 2011, there is a proposal under discussion with the membership and board of directors for the architecture council to formally expand its role to include maintenance of the EDP (currently maintained by Eclipse Foundation staff within the development process function; subsection 9.4.3) and to more formally recognize its role in "monitoring, guiding, and influencing the software architectures used by projects" (Beaton blog, May 20 2011).

The Eclipse governance rules provide the architecture council with no formal powers. In the words of one long-time participant, "AC has influence but no power" (*Minutes of the architecture council*, March 10 2011). In the technical language of the

French & Raven (1959) power typology, the AC has no coercive power or reward power, but it has considerable expert power and legitimate power deriving from the skills and experience of AC members and the high social status which AC members hold within the community. The AC directly impacts the rules-in-use of the developer community by producing *guidelines* (subsection 8.1.1, section 8.3 and Table 44) and other informational resources, but does not enforce their adoption. It indirectly impacts the community and ecosystem by making recommendation to the Executive Director, to Eclipse Foundation staff, and to the board of directors (through committer representatives and strategic member representatives who serve on both the architecture council and board of directors). The architecture council provides a channel for community requests into Eclipse governance (Oberhuber, 2008): individuals within the community can contact the architecture council to request an opinion, propose an enhancement, or escalate an issue.

The AC meets monthly by conference call, with minutes posted publicly to the Eclipse wiki. Other interactions include discussions and votes on the eclipse.org-architecture-council mailing list (publicly archived, but only AC members can post), Bugzilla for focused discussions on specific issues, one-on-one communications between AC mentors and their projects (mainly about the development process and community-building), and occasional face-to-face meetings – most notably on the Sunday before EclipseCon each year. According to the chair of the architecture council (Oberhuber, 2008), AC members are "linked in" with people who can find answers quickly and affect change, learn from others on the AC, and bring that information back to their projects. Some examples of recent AC activities (and associated Bugzilla discussions) include creation of a security policy at Eclipse (Bugzilla #337004), extension of the Eclipse.org IT infrastructure to include a Maven repository for software artifacts built at Eclipse[147] (Bugzilla #283745), and response to a call for help from the community when an Eclipse project did not at first respond to requests for a maintenance release but the AC persuaded project leads through arguments and authority (Bugzilla #251304). Interactions between AC members are similar to the interactions between committers on a community-driven project such as CDT or Mylyn (subsection 8.6): council members participate in discussions and activities that interest them, and decisions occur through lazy consensus (section 8.7), or calls for council member votes (Table 30). Observed outcomes in the case data include strong consensus on a recommendation or course of action, acknowledgment of a majority position and documentation of dissenting views, and agreement that there is no AC consensus on an issue or question.

Table 48 summarizes the operating rules for individuals serving on the AC.

[147] Apache Maven is an open source software build automation framework. In software development, the *build process* refers to the end-to-end process of converting human-readable software source code into executable software artifacts. Sonatype – one of the case companies examined in chapter 10 – leads the Apache Maven project, hosted at the Apache Software Foundation (http://maven.apache.org).

Table 48: Summary of operating rules of the architecture council (AC) (March 2011)

	Architecture council
Membership	Membership is comprised of individuals of three types: (1) PMC. Each PMC is entitled to appoint one AC member. (2) Strategic members. Each strategic developer is entitled to appoint one AC member (except for members leading a PMC or with another appointee to the council). (3) Appointed. The Executive Director can appoint additional individuals on a two-year renewable term. Since EDP 2.3.1 (2007), the existing members of the architecture council recommend new appointees to the ED through an internal election process. In March 2011, there were 51 AC members: 36 appointed by the Executive Director, 7 appointed by PMCs, and 8 appointed by strategic members, employed by 26 different organizations.
Scope of authority	"The Architecture Council is responsible for development, articulation and maintenance of the Eclipse Platform Architecture" (Bylaws, 2003, 2008, section 7.2).
Responsibilities	From 2004-2007 (when EDP 1.0 was in effect), the primary activity of the architecture council was creating an ***architecture plan*** to provide an explicit description of the architecture, protect the architecture from inadvertent corruption, and plot the evolution of the architecture in response to the purposes and roadmap. Since 2007 (EDP 2.3.1), the primary activity has been ***mentorship*** – by assigning experienced mentors to advise new projects, and by providing resources and recommendations to the community. • Individuals can contact the architecture council to request an opinion, propose an enhancement, or escalate an issue. • The architecture council (through its members) makes recommendations to the Executive Director, Eclipse Foundation staff, and the Board of Directors. • The architecture council creates and maintains information resources, including recommendations on best practices.
Changes to the operating rules	The operating rules are specified in the *Bylaws* and the *Eclipse Development Process*. Changes to the *Bylaws* require consent of at least two-thirds of the membership at large. Changes to the EDP require confirmation by a super majority (2/3) of the Board of Directors.

Compiled from sources attributed in section 9.5, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Eclipse Development Process* (EDP 1.0, 2003; EDP 2.3.1, 2006; EDP 2.4, 2008; EDP 2.5, 2010), meeting minutes, Bugzilla bug reports, posts to the eclipse.org-architecture-council mailing list, AC pages at the Eclipse Wiki, and the *Eclipse.org* website.

9.6 Planning council

According to the *Bylaws of the Eclipse Foundation* (2003, 2008, article 7), the Eclipse planning council (PC) is responsible for the development and maintenance of a platform release plan. Like the AC, PC membership is comprised of an individual appointed by each project management committee (PMC), individuals appointed by Eclipse Foundation strategic members, and individuals appointed by the Executive

Director. However, the PC membership is smaller and its composition is different. In March 2011, there were 21 PC members: 10 appointed by PMCs, 7 appointed by strategic members, and 4 appointed by the Executive Director.

Initially in 2004, the primary activity of the planning council was the creation of a platform release plan that "supports the Roadmap, ... enumerates the themes and priorities of the release, and contains each Project Plan" (EDP 1.0, section 1). In 2007, the responsibilities specified in the Bylaws were reinterpreted such that the primary activity became coordinating the simultaneous release (EDP 2.3.1, section 4.4) (previously examined in subsection 8.4.1).

Eclipse governance rules provide the PC with authority over the processes and content of the annual release train – a bundle of rules-in-use that section 8.4 and Table 44 labeled an "opt-in program." From the *Eclipse Development Process* (EDP 2.31, 2007):

> *The Planning Council is responsible for cross-project planning, architectural issues, user interface conflicts, and all other coordination and integration issues. The Planning Council discharges its responsibility via collaborative evaluation, prioritization, and compromise.*

Eclipse projects that choose to participate in the release train "opt in" to a set of simultaneous release requirements that are in addition to or more stringent than the project requirements of EDP. Most high-profile Eclipse projects choose to participate.

The PC meets monthly by conference call, with minutes posted publicly to the Eclipse wiki. Other PC interactions include discussions and votes on the eclipse.org-planning-council mailing list, discussion on the cross-project-issues mailing list, and occasional face-to-face meetings – most notably on the Sunday before EclipseCon each year. Unlike the AC, representation is largely hierarchical. PC members represent their respective PMCs, projects and committers by bringing forward issues for discussion and resolution, and disseminating information back to their constituents. Much of the mailing list activity is exception requests from project leads for waivers of PC release train rules, which are approved or rejected by PC member vote[148]. Much of the technical work required is release engineering – "releng" in the software development jargon – to produce tested builds on a pre-defined schedule for consumption by other projects.

Table 49 summarizes the operating rules for individuals serving on the PC.

Table 49: Summary of operating rules of the planning council (PC) (March 2011)

	Planning council
Membership	Membership is comprised of individuals of three types – same as AC in Table 48. In March 2011, there were 21 AC members: 10 appointed by PMCs, 7 appointed by strategic members, and 4 appointed by the Executive Director. Council members are employed by 13 different member organizations and the Eclipse Foundation: all except IBM and the Eclipse Foundation employ one member each. IBM employs 4 members, the Eclipse Foundations employs 2 members, and 3 members are not employed by member companies.
Scope of authority	"The Planning Council is responsible for responsible for the development and maintenance of a Platform Release Plan" (Bylaws, 2003, 2008, section 7.3).
Responsibilities	From 2004-2007 (when EDP 1.0 was in effect), the primary activity of the planning council was creating a ***platform release plan***. Since 2007 (EDP 2.3.1), the primary activity has been coordinating the simultaneous release (also known as the coordinated release or the Eclipse release train). The PC maintains informational resources about the release train (schedules, guides, checklists, etc.), approves (or denies) requests for changes to the train or exceptions to rules, conducts retrospectives, and organizes community polls to name the next train.
Changes to the operating rules	Same as architecture council (Table 48).

Compiled from sources attributed in section 9.6, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Eclipse Development Process* (EDP 1.0, 2003; EDP 2.3.1, 2006; EDP 2.4, 2008; EDP 2.5, 2010), meeting minutes, posts to the eclipse.org-planning-council mailing list, posts to the cross-project-issues-dev mailing list, PC pages at the Eclipse Wiki, and the *Eclipse.org* website.

[148] A typical exception request email includes an explanation and justification of the exception requested, often with links to an email thread where the request was discussed and approved by the PMC and other background and technical information. PC members may ask questions and engage in discussion, or may simply vote +1/0/-1 on the request.

9.7 Requirements council

According to the *Bylaws of the Eclipse Foundation* (2003, 2008, article 7), the Eclipse requirements council (RC) is responsible for reviewing and categorizing incoming requirements, and proposing a coherent set of themes and priorities that will drive the annual roadmap. Its membership is comprised of an individual appointed by each strategic member, and individuals appointed by the Executive Director. In March 2011, according to the Eclipse.org website, there were 14 members of the requirements council: 11 appointed by strategic members of the Eclipse Foundation and three appointed by the Executive Director. However, the requirements council has been largely inactive since 2007[149].

Initially in 2004, the primary activity of the requirements council was the creation of themes and priorities to "drive the Roadmap" (EDP 1.0, section 1). In 2007, the text of the EDP changed to state that the requirements council is "primarily responsible for the roadmap" (EDP 2.3.1, section 4.4). Beginning in 2007, facing declining community participation in RC meetings, the RC chair (a member of the Eclipse Foundation staff) maintained the themes and priorities as a page in the Eclipse wiki, with calls for participation and requests for comments posted to the AC and PC mailing lists. In 2011, a proposal is under discussion with the membership and board of directors to remove the requirements council from the Bylaws, and to replace the annual roadmap with an annual community report produced by the EMO.

Table 50 summarizes the operating "rules-in-form" for the RC.

[149] The most recent published RC meeting minutes are dated June 2007. There is a published agenda for Q4 meeting, but no record that the meeting occurred.

Table 50: Summary of operating rules of the requirements council (RC) (March 2011)

	Requirements council
Membership	Individuals of two types: (1) Strategic members. Each strategic developer and strategic consumer is entitled to appoint one member. (2) Appointed. The Executive Director can appoint additional individuals. In March 2011, there were 14 RC members: 11 appointed by strategic members of the Eclipse Foundation and three appointed by the Executive Director. Council members are employed by 10 different member companies: all except IBM employ one member each (IBM employs 2 members).
Scope of authority	The requirements council is responsible for "reviewing and categorizing incoming requirements, and proposing a coherent set of themes and priorities that will drive the Roadmap" (*Bylaws*, 2003, 2008, section 7.4).
Responsibilities	From 2004-2007 (when EDP 1.0 was in effect), the primary activity of the requirements council was creating a ***themes and priorities*** document. Since 2007 (EDP 2.3.1), the requirements council has been said to be "primarily responsible for the roadmap" in addition to the themes and priorities document. Activity on the RC declined in 2007, and the RC became inactive in 2008. Version 5 (2010) and version 6 (2011) of the themes and priorities document were drafted by the council chair, with requests for comments posted to mailing lists.
Changes to the operating rules	Same as architecture council (Table 48).

Compiled from sources attributed in section 9.7, including the *Bylaws of the Eclipse Foundation* (2003, 2008), the *Eclipse Development Process* (EDP 1.0, 2003; EDP 2.3.1, 2006; EDP 2.4, 2008; EDP 2.5, 2010), meeting minutes, RC pages at the Eclipse Wiki, and the *Eclipse.org* website.

9.8 Membership at-large

Section 7.1 and Table 22 previously examined benefits and obligations of the six membership classes of the Eclipse Foundation. The *membership at-large*, referring collectively to all organizational and individual members of the Eclipse Foundation at all membership classes (*Bylaws*, section 6.1), also plays a role in Foundation governance. Other than affiliate members[150], the membership at-large is entitled to vote on membership matters. On each matter submitted to a vote of the membership at-large, each member is entitled to one vote, with two important restrictions. First, that multiple affiliates of one entity constitute one member only for voting purposes (*Bylaws*, section

[150] Affiliate membership is the lowest tier of Eclipse membership (Table 22), available with no membership dues to not-for-profits and other qualifying organizations. According to the Eclipse *Membership Agreement* (2010, section 1.5), affiliate members of the Eclipse Foundation are not represented on the board of directors (section 9.2) and are not permitted to vote with the membership at-large.

6.2) – a rule that comes into play following company acquisitions and also prevents any company from obtaining greater influence through multiple memberships of subsidiaries or similar means[151]. Second, that committer members who are employed by the same organization receive only one collective vote (*Bylaws*, section 6.12a). Section 6.5 specifies a list of "major decisions" that require approval by a vote of two-thirds of the membership at-large: (1) approving or changing the name of the Eclipse Foundation or the Eclipse platform, (2) approving or amending the *Membership Agreement*, or (3) amending the *Bylaws* or the *Certificate of Incorporation*. Membership votes are conducted electronically (*Bylaws*, section 6.12) through the *MyFoundation Portal*.

Strategic members, sustaining members, and committer members have representation on the Eclipse board of directors (see section 9.2 and Table 45). Each strategic developer and strategic consumer are entitled to appoint one director. Sustaining members – including enterprise members and solutions members – and committer members vote in annual elections to select their representation on the board.

9.9 Timeline

This section briefly examines how the polycentric governance structures of the Eclipse Foundation have collectively changed over time. Figure 15 is a visual timeline of major events, changes in formal operating rules, and changes within Eclipse governance structures, from November 1998 (the launch of Eclipse as an internal project within IBM; Nackman, 2005) to March 2011, and looking ahead to proposed future changes[152].

[151] "Affiliate" is defined precisely in section 6.2 of the *Bylaws* to limit potential for abuse.

[152] The timeline of Figure 15 is a compilation from various sources cited in chapters 7, 8 and 9, and chains of evidence documented in the case notes documents for the Eclipse board of directors, Eclipse Foundation staff, architecture council, planning council, and requirements council.

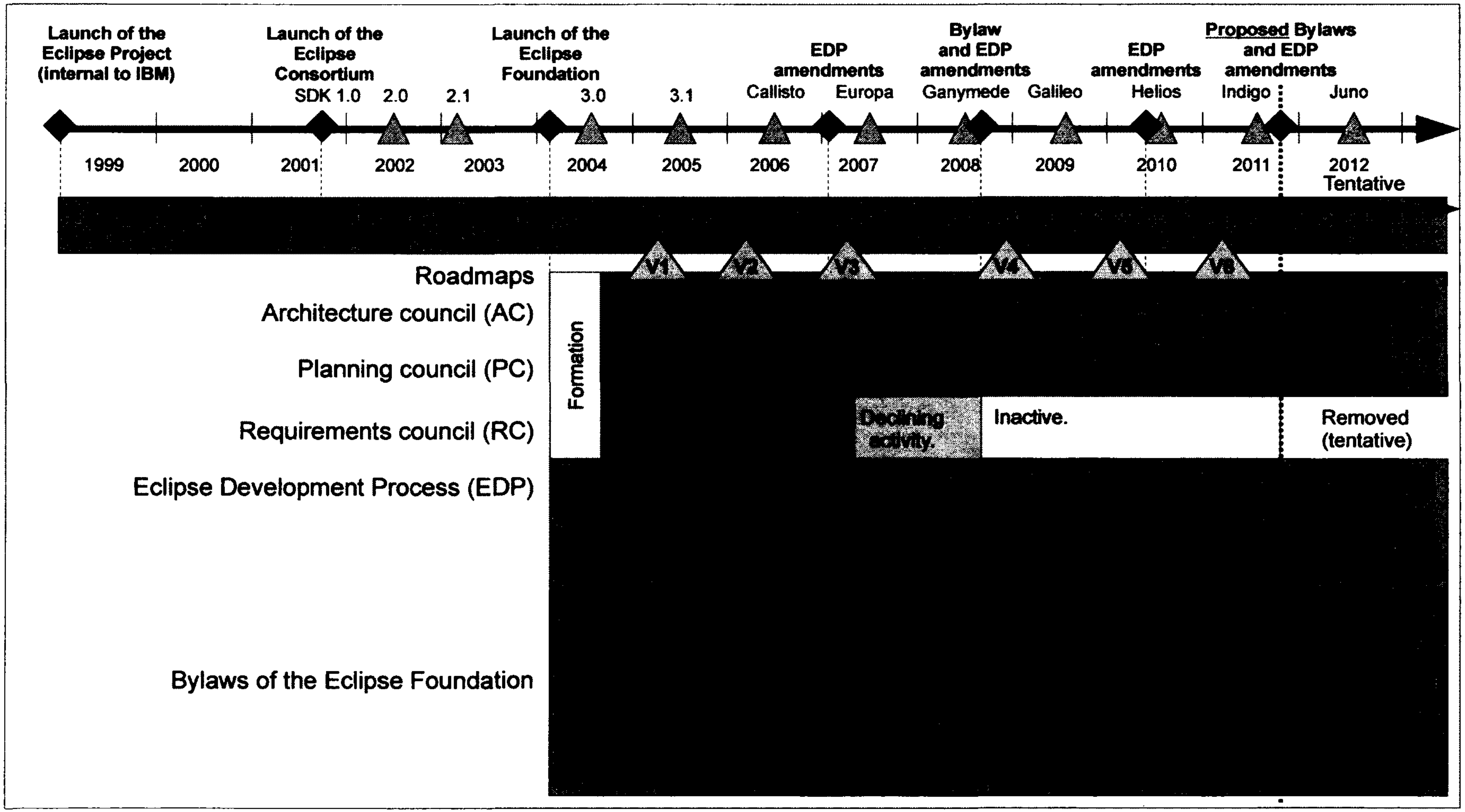


Figure 15: Timeline of polycentric governance structures and operating rules

O'Mahony et al. (2005a) and O'Mahony et al. (2005b) recount the internal decisions at IBM leading to the launch of the Eclipse Consortium in 2001 and the Eclipse Foundation in 2004 as *Harvard Business School* teaching cases. According to O'Mahony et al. (2005b), the initial Eclipse Bylaws were "crafted" (p. 3) as the outcome of an eight-month iterative process orchestrated by IBM staff on the Eclipse Consortium board of stewards[153], beginning in April 2003 and completed by November[154]. Tasks within the "search for a governance model" (p. 3) included identifying the problems with the existing consortium governance structure, evaluating alternative governance models (said to include the Apache Software Foundation, Sun Microsystem's Java Community Process (JCP), the US Congress, and the United Nations), conversations with Consortium members and potential members, and careful legal review. The major design decisions included tiered membership with five membership tiers, one vote for each member company regardless of the number of contributing developers, a board of directors with representation of all membership classes, and an Eclipse Management Organization with staff and councils led by an Executive Director.

This account begins in 2004 where O'Mahony et al. (2005a, 2005b) leaves off. A comprehensive treatment of governance changes over time is outside the scope of the guiding research questions motivating this research. The intent is here is rather to identify patterns salient to understanding company participation in the Eclipse field

[153] O'Mahony et al (2005b) credits IBM staff on the Eclipse Consortium board of stewards with leading this process – particularly Skip McGaughey, the founding chairperson of the board and start-up specialist at the IBM Software Group, and Dave Bernstein, chair of the Eclipse Independence Subcommittee and former VP of Development at Rational Software, acquired by IBM in 2003.
[154] The founding *Bylaws of the Eclipse Foundation* (2003), approved by the Eclipse Consortium board of stewards, were last updated on November 10 2003.

setting – in particular, the extent to which the contents of the IAD tables for individual participation and company participation have changed or remained stable.

At the launch of the Eclipse Foundation in 2004, the Eclipse roadmap process (subsection 8.2.5) was the central organizing feature of Eclipse governance. Each year, the requirements council was to propose "a coherent set of themes and priorities that will drive the roadmap," the Executive Director was to lead the councils "to produce a roadmap," the architecture council was to create an architecture plan "in response to the … roadmap," and the planning council was to create a platform release plan that "supports the roadmap" (*Bylaws*, 2003, article 7) – all to be reviewed by the Eclipse community, and reviewed and approved by the board of directors. The initial version of the *Eclipse Development Process* (EDP 1.0, 2003, section 2) summarized the roadmap process as four steps: (1) the membership determines the purposes of Eclipse, (2) the EMO proposes a roadmap (or update) consistent with the purposes, (3) the board of directors approves the roadmap, and (4) development teams work to the roadmap.

Table 51 compares all six extant versions of the *Eclipse Roadmap*, beginning with version 1, developed in the Eclipse Foundation's first year of operation and approved by the Eclipse Foundation Board of Directors in early 2005, up to version 6, approved in 2011. Together, Figure 15 and Table 51 provide a useful guide to changes within Eclipse governance.

Table 51: Summary of Eclipse Roadmaps, version 1 (2005) – version 6 (2011)

	Version 1	Version 2	Version 3	Version 4	Version 5	Version 6
Time horizon	2005	2006	2007	2008/2009	2010-2011	2011-2012
BoD approval	February 28 2005	March 20 2006	March 5 2007	December 12 2008	March 22 2010	March 21 2011
Strategic goals	n/a	3 strategic goals: • Define an open development platform embodying technology leadership and innovation. • Spur growth and adoption of Eclipse technology. • Enable a commercially successful ecosystem.	4 strategic goals: • Define a vendor-neutral open development platform demonstrating technology leadership and innovation. • Invest in recognition. • Foster growth & adoption of Eclipse technology. • Cultivate commercial success of the ecosystem.	4 strategic goals: • Establish Eclipse as a leading provider of OS runtime technologies. • Maintain leadership in OS tools platforms. • Create value for all membership classes. • Foster growth of the ecosystem, particularly in verticals.	7 strategic goals: the 4 goals of V4 + 3 new goals: • Run a good ship. • Grow a diversified revenue model. • Ensure adequate resources are invested in the core technology platform.	7 strategic goals; the first 6 are unchanged from V5; the 7th goal is the following: • Establish Eclipse web technology as a leading OS web application platform.
Themes and priorities	V1 by requirements council. 4 fundamental principles. 8 themes: • Scaling up. • Enterprise ready. • Design for extensibility. • Embedded development. • Rich client platform. • Simple to use. • Multi-language support. • Broad community appeal. 4 out-of-scope inputs.	V2 by requirements council. 4 fundamental principles. 8 themes: • Scaling up. • Enterprise ready. • Design for extensibility. • Embedded development. • Rich client platform. • Simple to use. • Multi-language support. • Broad community appeal.	V3 by requirements council. 4 fundamental principles. 4 theme categories. 10 active themes: • Platform support beyond Java on Windows. • Rich client platform. • Embedded device s/w. • Ease of use. • Improving the "out of the box" experience. • Technology trends • Scaling up. • Enterprise ready. • Design for extensibility. • Multi-language support.	V4 by requirements council. 4 theme categories. 10 active themes: • Platform support. • Rich client platform. • Rich internet applications. • Embedded device s/w. • Ease of use. • Technology trends • Scaling up. • Enterprise ready. • Design for extensibility. • Consistent multi programming language support. Known issues section.	V5 by Eclipse councils. 2 theme categories. 10 active themes: • Platform support. • Rich client platform. • Rich internet applications. • Embedded device s/w. • Ease of use. • Technology trends • Scaling up. • Enterprise ready. • Design for extensibility. • Consistent multi programming language support.	V6 by Eclipse councils. 2 theme categories. 10 active themes: • Platform support. • Rich client platform. • Rich internet applications. • Embedded device s/w. • Ease of use. • Technology trends • Scaling up. • Enterprise ready. • Design for extensibility. • Consistent multi programming language support.
Platform release plan	V1 by planning council.	V2 by planning council. Callisto (2006, 10 projects).	V3 by planning council. Europa (2007, 21 projects).	V4 by planning council. Galileo (2009); automated.	V5 by planning council. Helios (2010); automated.	V6 by planning council. Indigo (2011); automated.
Architecture plan	V1 by architecture council.	V2 by architecture council.	n/a	n/a	n/a	n/a

Sources include the *2005 Eclipse Roadmap* (Version 1), *2006 Eclipse Roadmap* (Version 2), *2007 Eclipse Roadmap* (Version 3), *2008 Eclipse Roadmap* (Version 4), *2010 Eclipse Roadmap* (Version 5), *2011 Eclipse Roadmap* (Version 6), and *Abridged minutes of the Eclipse Foundation board of directors*, as well as *Executive Director Reports*, council meeting minutes, wiki documents, and email posts the council mailing lists.

The first documented board of directors meeting in June 2004 established six objectives set for the new Executive Director for the upcoming third quarter (*Abridged minutes of the Eclipse Foundation board of directors*, 2004Q2, June 25):

1. Grow the membership base, including expanding the number of strategic members.
2. Improve the marketing effectiveness of the EMO.
3. Implement the new development process. Specifically, create the Requirements, Planning and Architecture Councils and organize their initial meetings.
4. Plan for migrating the Eclipse website from the current IBM infrastructure to one which is managed directly by Eclipse.
5. Establish the Eclipse administrative services such as banking and payroll.
6. Plan for migrating the Eclipse distribution license from the CPL to the EPL.

Consistent with the third objective, all three councils met for the first time in face-to-face meetings on consecutive days, August 31 to September 2, in Scottsdale Arizona, followed by an agenda of monthly conference calls, and face-to-face meetings, approximately quarterly, that continued into 2006.

The first two years of operation were a period of rapid growth in membership and projects. By the Annual General Meeting in March 2006, the Eclipse Foundation had grown from 50 member organizations to 130, and from 3 top-level projects to 8, with dozens of new subprojects. The scope of activity also continued to expand. In 2001, the platform had been about software tools (especially the Eclipse Java IDE) and frameworks for creating software tools. By 2004, the platform included rich client applications and frameworks for building rich client applications. Over the next two years, there was growing activity in Eclipse runtimes – software that executes on server hardware rather desktop computers or user devices. As described in section 7.4, the expanding platform scope was achieved by successively refactoring, repartitioning, and repackaging the platform software to address a broader range of software applications and adopters.

As the scope and scale of activity grew, some things worked well while others did not. Already by 2005, there were concerns from the architecture council about maintaining consistent quality and processing the volume of new proposals (*Minutes of the architecture council*, May 2 2005):

> *There is some concern that the quantity of new proposals has made it difficult to provide appropriate qualitative reviews, hence there is a risk that the quality of Eclipse projects could slip. Eclipse is known for its high quality frameworks and tools and projects, and thus it is important for us to take steps to retain that.*
>
> *At the same time, we don't want to be too restrictive. We want to keep the hassle-factor low so that we encourage new ideas to come to Eclipse, but we want to raise the bar (or, perhaps the bar is already high enough, but just not documented) for making a release. We want Eclipse projects to have an open, transparent, inclusive, and welcoming process; we want Eclipse projects to have a wide community of tool users and framework users.*

More specifically, some processes that had been extensions of how things were done on the Eclipse platform project by IBM staff didn't scale well across all projects. Discussions began on the first revisions to the Eclipse development process. Community discourse in 2006 and 2007 reveals active debate around all proposed amendments and also larger questions of the "right" amount of process, the appropriate role for governance groups to be involved in projects, and the extent to which roles should be enforced through coercion or community norms. The 2007 Q4 meeting of the board of directors discussed the roadmap process and potential governance changes to "refocus the energies of the Requirements Council". From the abridged minutes:

> *The Bylaws require that each year the Requirement, Planning and Architecture Councils collaborate on the Eclipse Roadmap. This document has historically required a significant amount of effort from the Councils for an unclear return on that investment in time and energy. Several Board members found the document quite valuable in order to explain Eclipse to their management and how Eclipse fits*

> *into the company's strategic direction. After some discussion it was decided that there would be a roadmap, that the EMO will write the preamble ... the Planning Council would be responsible for validating and selecting set of themes and priorities consistent with those prepared by the Requirements Council which each project in the annual release train would be expected to use in their plans...*

Discussion of the roadmap process continued at the board of directors in 2008 Q2. Also in 2008, the first revision to the Eclipse Bylaws came into effect (discussed in section 9.1), as did the second revision to the EDP (discussed in subsection 8.1.5 and section 9.1). At the 2011 Eclipse Annual General Meeting, the Executive Director began community discussions on a second amendment to the Bylaws (*Executive Director Report*, Q1 2011, p. 38):

> *The Eclipse Foundation has been in existence for seven years. Most of the ideas that went into the original Bylaws have worked well and have stood the test of time. A few have not. Let's tidy up the Bylaws to ensure that they reflect current practices.*

The most significant changes proposed are replacing the roadmap process (subsection 8.2.5) with an annual community report written by Eclipse staff, and the various changes to the Eclipse councils previously described in sections 9.5, 9.6 and 9.7.

In summary, all of the major Eclipse Foundation governance design decisions identified by O'Mahony et al. (2005b) – tiered membership, one vote for each member company regardless of the number of contributing developers, a board of directors with representation of all membership classes, and an Eclipse Management Organization with staff and councils led by an Executive Director – remain in place in 2011. Within that framework, however, much is different. All three of the Eclipse councils perform different functions than those envisioned in 2003. The annual planning function has transformed from *driving* project activity to accurately reporting on what the projects are doing. The membership classes and the development process also both changed.

Chapter 8 previously examined the rules-in-use within Eclipse projects. Of the four categories of project operating rules identified in chapter 8, EDP 1.0 (2003) codified only two: mandatory requirements across all projects and local conventions ("[The EDP] does not describe the internal operation of those teams, which is governed by the Project Charter", p. 1). EDP 2.31 (2007) added recognition for unwritten community norms and guidelines that are recommended but not enforced. (The opt-in programs, first launched in late 2005 for the Callisto release train in 2006, are managed outside of the EDP). Over time, new rules were introduced, such as the rules for release numbering and release management introduced in EDP 2.4 (2008). Existing rules were sharpened and improved, such as clarification of the distinct roles of committers, product leads, and PMC leads in EDP 2.5 (2009). Tacit practices were made explicit and codified, such as the use of perpetual incubator projects in EDP 2.5 (2009). Ineffective rules were removed, such as the requirement for an architecture plan covering all Eclipse projects (removed from EDP 2.31 in 2007) and the expectation for synchronous meetings or conference calls for each project review (replaced with "review periods" and discussion managed asynchronously through IT communication channels in EDP 2.5 in 2010).

The key insights from this temporal perspective are the following. (1) The initial Eclipse Foundation governance model was *designed* – mainly by senior managers at IBM, in consultation with senior managers at other companies involved in the Eclipse Consortium. (2) The big design decisions (O'Mahony et al. 2005b) have remained intact: tiered membership, one vote for each member company regardless of the number of contributing developers, a board of directors with representation of all membership classes, and an Eclipse Management Organization with staff and councils led by an Executive Director. (3) The lower-level operating rules have changed over time, with new rules added, existing rules sharpened and improved, tacit practices made explicit and codified, and ineffective rules removed. The over-all pattern is a mix of deliberate and emergent strategy (Mintzberg, 1987), with an early and pronounced shift in emphasis from the former to the latter. Chapter 10 and chapter 11 revisit these observations from the perspective of managing tensions between the community and ecosystem.

9.10 Synthesis and theory development

Previous sections of this chapter have specified the operating rules of the various polycentric governance groups that comprise the Eclipse Foundation. This section re-interprets those results from the perspective of the Eclipse developer community and the Eclipse business ecosystem. The objective is an understanding of how Eclipse governance structures interact with the institutions that structure individual and company participation in the Eclipse field setting.

Figure 16 illustrates the ways in which the polycentric governance groups of the Eclipse Foundation – the membership at-large, the board of directors, the Executive Director, the Eclipse Foundation staff, and the Eclipse governance councils – relate to one another through formal authority. Table 52 is a summary of information presented throughout this chapter on interactions between governance groups and the developer community and ecosystem. Collectively, Figure 16 and Table 52 explore the internal structure of the "Eclipse Foundation" construct in Figure 10 of chapter 7.

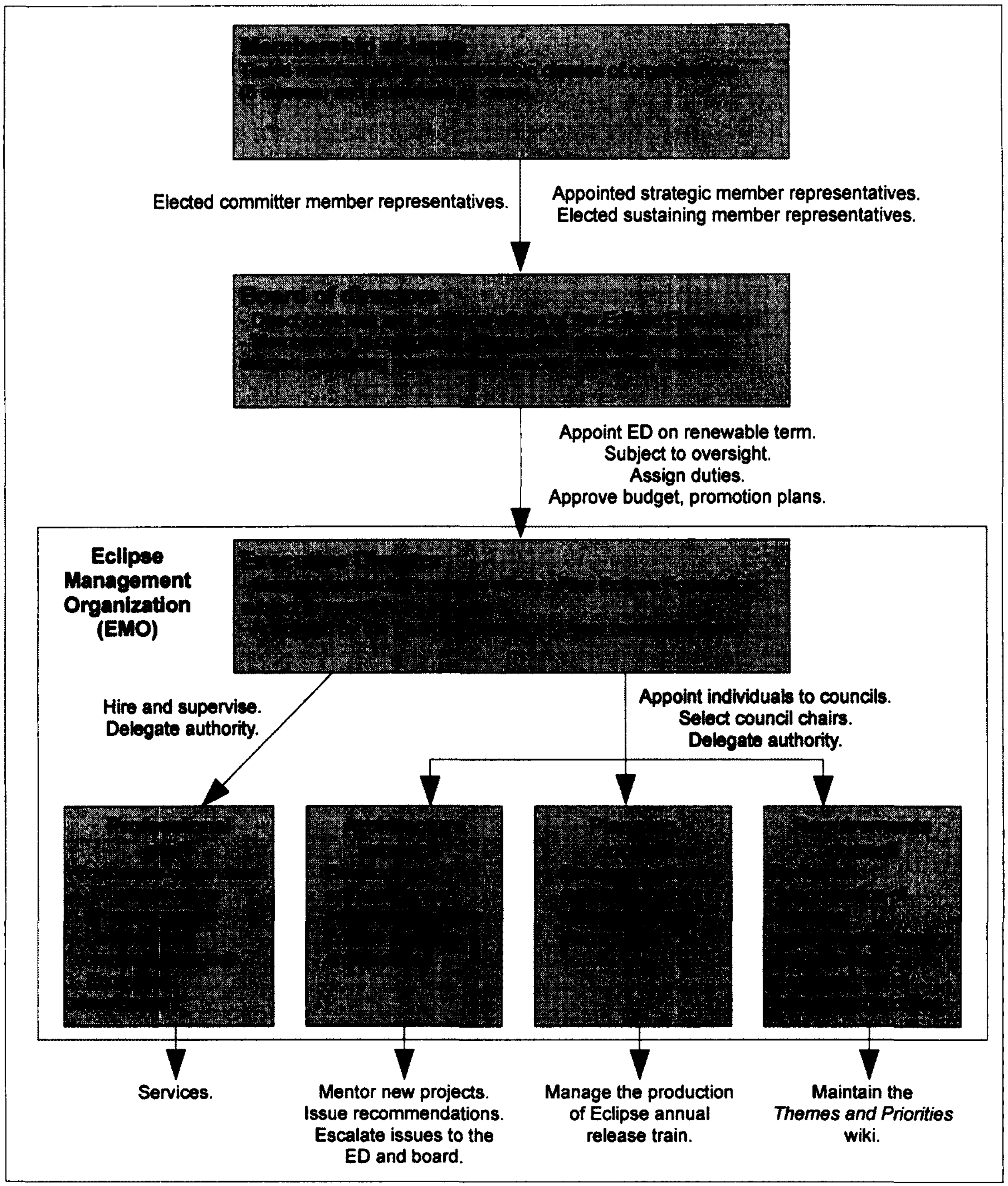

Elected committer member representatives.
Appointed strategic member representatives.
Elected sustaining member representatives.
Appoint ED on renewable term.
Subject to oversight.
Assign duties.
Approve budget, promotion plans.
Eclipse Management Organization (EMO)
Hire and supervise.
Delegate authority.
Appoint individuals to councils.
Select council chairs.
Delegate authority.
Services.
Mentor new projects.
Issue recommendations.
Escalate issues to the ED and board.
Manage the production of Eclipse annual release train.
Maintain the *Themes and Priorities* wiki.

Figure 16: Relationships between governance structures (March 2011)

Table 52: Influence of governance structures on community and ecosystem

	The Eclipse Foundation as an open source software foundation	The Eclipse Foundation as the keystone organization of a business ecosystem
Governance structure	Impact of governance on the Eclipse developer community	Impact of governance on the Eclipse ecosystem
Membership at-large	• Vote on membership matters impacting the community.	• Vote on membership matters impacting the ecosystem.
Board of directors	• Confirm the creation of new top-level projects and PMCs. • Confirm nominations of PMC leads. • Approve the annual release roadmap. • Approve *IP Policy* exceptions, including use of a distribution license other than the EPL. • Approve changes to the EDP.	• Approve *IP Policy* exceptions, including use of a contribution license other than the EPL. • Company enforcement mechanisms: may terminate a member's agreements. • Approve changes to the *Membership Agreement*.
Executive Director (authority may be delegated to others)	• Decide the outcomes of project reviews, including creation reviews for new projects and release reviews to release code. • Nominate PMC leads (confirmed by board). • Nominate new PMCs for top-level projects (confirmed by board). • Approve new project leads (elected by project committers) • Enforce the EDP requirements and board directives (when necessary). Project enforcement mechanisms: (1) terminate a project, (2) remove leadership, (3) unfavourable IP due diligence, (4) unfavourable project reviews.	• Lead Eclipse platform marketing, evangelism, promotion, public relations and industry events. • Decide the outcomes of project reviews, including creation reviews for new projects that companies bring to Eclipse. • Issue Executive Director reports to the Eclipse membership at-large and chair meetings of the Eclipse membership at-large.
Professional staff	• Establish and maintain legal documents including the *Committer Agreement* (required for new committers). • Make decisions in community matters, as delegated by the Executive Director. • Manage project resources, community resources, and other IT infrastructure. • Organize project reviews and logistics.	• Operate "opt-in member programs" that provide value to member companies. • Implement Eclipse marketing programs that drive traffic to member websites. • Organize face-to-face regional events. • Through the IP due diligence process, provide confidence to ecosystem companies that Eclipse software can be safely distributed in commercial products.
Architecture council	• Mentor new projects. • Issue and maintain guidelines and best practice recommendations. • Act on community requests for an opinion, enhancement, or escalation of an issue.	• Mentor company-driven projects to build community and contributor diversity. • Provide a channel for employers to the technical experts and technical decision-makers of the Eclipse community.
Planning council	• Manage the production of the Eclipse annual release train. • Produce informational resources. • Approve or deny exception requests.	• Through the Eclipse annual release train, provide stable and tested software releases on a predictable timeline.
Requirements council	• Maintain the *Themes and Priorities* wiki.	• Maintain the *Themes and Priorities* wiki.

The Eclipse Foundation polycentric governance structures are involved in the operating rules of the Eclipse community and Eclipse ecosystem as well as the policy rules (Ostrom, 1990) by which the operating are created and re-shaped, and the constitutional rules (Kiser & Ostrom, 1982) of who can, who cannot, who must, and who must not participate in policy rules.

At the *operating level of the Eclipse developer community*, the planning council manages the production of the Eclipse annual release train, the architecture council provides mentorship and intervention upon request, the professional staff manage various resources, the Executive Director (either directly or through delegates) decides the outcomes of project reviews and enforces EDP requirements and board directives, and the board of directors approves the annual release roadmap.

At the *policy level of the Eclipse developer community*, each governance group is involved. The planning council establishes the (opt-in) rules by which the annual release train is produced and approve or deny exception requests from project committers. The architecture council issues and maintains guidelines and best practice recommendations and provides opinions on architectural matters upon request; these activities impact the guidelines and also the community norms of project rules-in-use through the high social capital of council members. The Executive Director and the professional staff manage the activities involved in revising governance documents. The board of directors approves IP policy exceptions and changes to governance documents that specify the rules-as-written. The membership at-large approves membership matters impacting the community, most notably amendments to the *Bylaws*. Project committers, project leads, PMC members, and PMC leads establish local project conventions.

At the *constitutional level of the Eclipse developer community*, committer members elect committer member representatives to the board of directors, which collectively hold approximately fifteen percent of board votes. The board of directors appoints the Executive Director, and confirms nominations of PMC leads and new PMCs. The Executive Director nominates new PMCs, nominates individuals from the community as PMC leads and members of new PMCs, approves new project leads, appoints individuals to the Eclipse councils, selects the council chairs, hires and manages the professional staff, and may delegate authority to other individuals. The architecture council, through a self-managed nomination and election process, recommends new council members to the Executive Director who normally appoints those recommended individuals to the council. Project committers select and appoint new committers through a structured meritocratic nomination and election process (subsection 8.2.1) and elect new project leads. PMC members elect new PMC members, approve the election of project leads by the committers of subprojects, appoint PMC representatives to the Eclipse councils, and may remove a project lead that has become inactive or disruptive.

At the *operating level of the Eclipse ecosystem*, the planning council manages the production of stable and tested software releases on a predictable timeline, the architecture council provides a channel for information and influence, the professional staff operate membership programs and organize events, the Executive Director (or ED delegate) decides the outcome of project creation reviews and issues ED reports on membership matters, and the board of directors enforces membership obligations..

At the *policy level of the Eclipse ecosystem*, the Executive Director and professional staff maintain the Eclipse legal policies, establish the operating rules of the opt-in members programs, and manage the activities involved in revising the *Membership Agreement*. The board of directors approves IP policy exceptions and changes to governance documents that specify the rules-as-written – most notably the Eclipse *Membership Agreement.*

At the *constitutional level of the Eclipse ecosystem*, organizational members appoint strategic representatives to the board of directors and elect sustaining member representatives, collectively holding approximately seventy percent and fifteen percent of board votes, respectively. Strategic developer members are entitled to appoint representatives to the Eclipse councils.

In summary, Eclipse governance structures interact with the institutions that structure individual and company participation at both the operational level and the policy level in multiple ways. At the constitutional level, the Eclipse board of directors and councils are comprised of elected and appointed representatives from the Eclipse developer community and the Eclipse ecosystem; strategic member companies collectively hold the majority of formal voting power, but all membership classes have a voice on the board and receive time and attention on each meeting agenda. Governance is said to be *vendor-neutral* in the sense that all members of a particular membership receive the same the membership powers and member obligations.

This chapter examined governance-level activity within Eclipse governance structures. The next chapter is a similar study of company participation in Eclipse.

10 Company participation

This chapter examines company activity in the Eclipse ecosystem in order to discover and articulate the institution that structures and motivates company participation. Previous chapters identified the formal rules structuring company participation: (1) the property rights and legal frameworks governing use and redistribution of Eclipse platform assets (chapter 7), (2) the Eclipse *Membership Agreement* that specifies member powers and obligations (chapters 7 and 9), and (3) opt-in programs for Eclipse Foundation member companies operated by Eclipse Foundation staff (chapter 9). This chapter begins by examining the activities of eight participating companies through direct observation, archival records, and interviews with informants. For each case company, it identifies a business model and ways in which the company benefits from its participation in Eclipse. From these observations, and from previous observations about the rules governing company interaction, it develops an IAD representation of the Eclipse ecosystem as an institution, and explores tensions between the developer community and business ecosystem.

10.1 Company case selection

Of the approximately 170 member organizations of the Eclipse Foundation (*Executive Director Report*, March 2011), this research closely examines eight case companies. Each employs active committers to one or more of the case projects examined in chapter 8. Collectively, the set of case companies includes the most active contributors to each of the five case projects[155] as well as companies with lower levels of

[155] Table 37 later in this chapter provides the supporting evidence of this assertion.

contribution selected for diversity and potential for insights. The eight case companies and their involvement in the case projects of chapter 8 are as follows:

- *Actuate Corporation* is a publicly-traded provider of software products and services for developing enterprise business intelligence (BI) applications. Actuate is a strategic developer member of the Eclipse Foundation; it founded and co-leads the Eclipse BIRT project, currently employing all active BIRT committers.
- *EclipseSource* is a privately-held provider of Eclipse-based services and software products. It is a strategic developer member of the Eclipse Foundation. EclipseSource staff lead the Eclipse RT (runtime) top-level project, and lead or contribute to many Eclipse runtime and platform projects, including Equinox.
- *IBM* is a multinational technology and consulting firm, and the world's largest software company. IBM developed the original Eclipse 1.0 SDK, orchestrated the formation of the Eclipse Consortium and its transition to the Eclipse Foundation, and employs more Eclipse committers than any other single company. IBM staff contribute to more than sixty Eclipse projects, including JDT, CDT, and Equinox. IBM is a strategic developer member of the Eclipse Foundation.
- *Innovent Solutions* is a privately-held provider of information management consulting services, product sales, and support. Innovent is a solutions member of the Eclipse Foundation, and contributes the BIRT project as a PMC member.
- *Intel Corporation* is the world's largest semiconductor manufacturer. *Wind River Systems*, a provider of operating systems and middleware software for embedded systems, is a wholly-owned subsidiary of Intel, acquired in 2009. Both Intel and Wind River were founding members of the Eclipse Foundation; Intel is currently a solutions member. Intel and Wind River staff lead the CDT project.
- *Nokia Corporation* is the world's largest manufacturer of mobile phones. It is a strategic developer member of the Eclipse Foundation. Nokia staff contribute to CDT and other projects related to software development on mobile phones. It developed and contributed the code for the Eclipse Debugger for C/C++ (EDC).
- *Sonatype* is a venture capital-backed provider of software products and support services for enterprise software development, with particular focus on the software development build process. It founded the Apache Maven project at the Apache Software Foundation, and is a strategic developer member of Eclipse.
- *Tasktop Technologies* is a privately-held provider of software products and services for application lifecycle management. It leads the Mylyn project.

Figure 17 illustrates the relationships between the project cases of chapter 8 and the company cases of this chapter as a simplified network diagram (section 3.5).

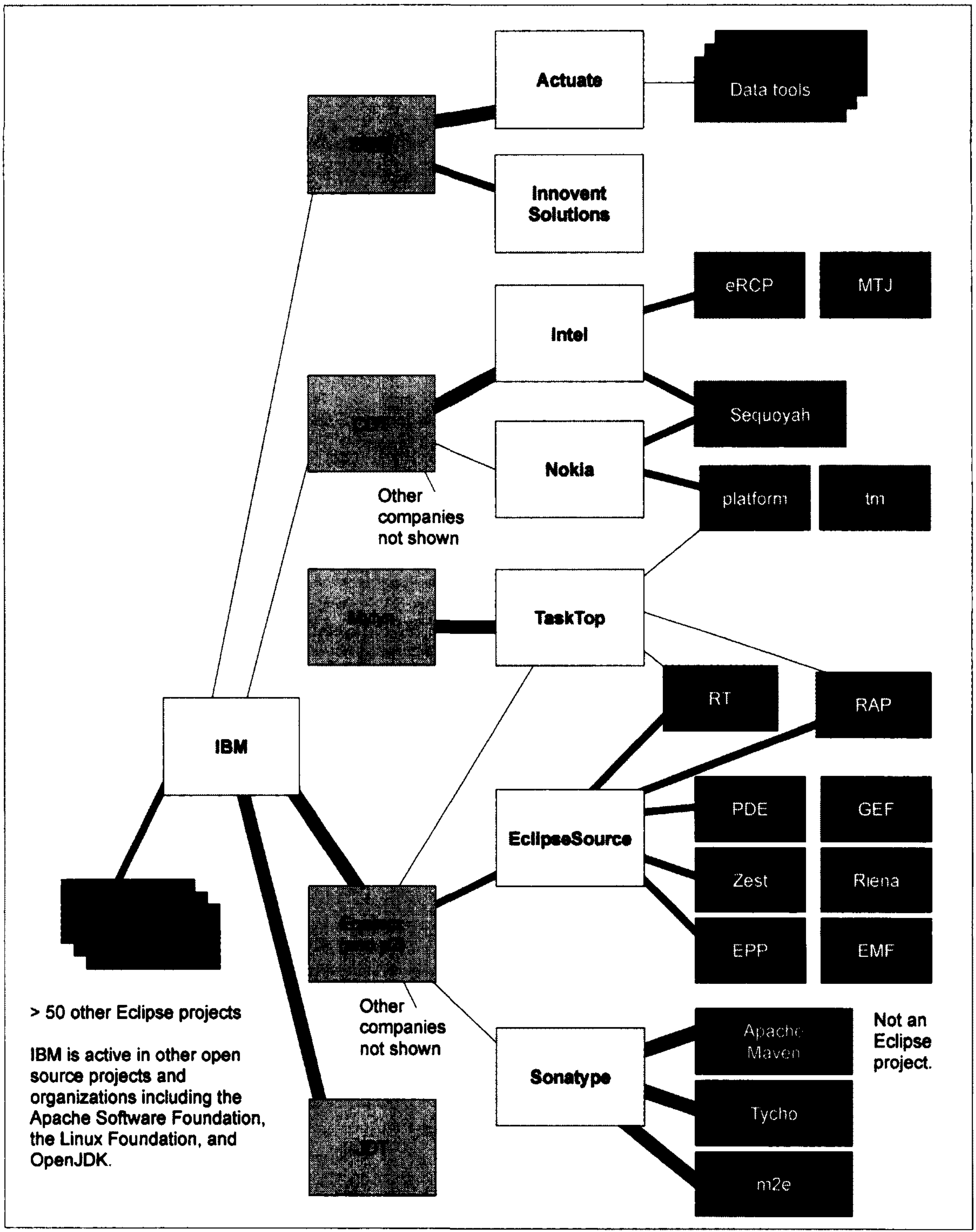

Actuate
Data tools
Innovent Solutions
eRCP
MTJ
Intel
CDT
Sequoyah
Nokia
platform
tm
Other companies not shown
TaskTop
RT
RAP
IBM
PDE
GEF
EclipseSource
Zest
Riena
EPP
EMF
> 50 other Eclipse projects
IBM is active in other open source projects and organizations including the Apache Software Foundation, the Linux Foundation, and OpenJDK.
Other companies not shown
Apache Maven
Not an Eclipse project.
Sonatype
Tycho
m2e
JDT

Figure 17: Relationships between case projects and case companies (March 2011)

The nodes of the network diagram are case companies (light grey shading, dark bold font), case projects (medium grey shading, dark font), and other Eclipse projects (dark grey shading, light font). Ties indicate company participation in a project, with the thickness of the tie indicating the extent of participation. Factors considered include employing committers, employing project leads and PMC members, and code donations.

10.2 Cross-case analysis

Table 53, Table 54, Table 55, Table 56 and Figure 18 compare the eight case companies along various dimensions of interest. Table 53 provides a descriptive summary of aggregate characteristics, including primary industry, major events, ownership, and two measures of size (number of employees and revenue). Table 54 summarizes each company's participation in Eclipse along ten dimensions. Table 55 and Table 56 briefly summarize salient aspects of each company's business model using the modified Johnson et al. (2008; Johnson, 2010) framework developed in section 3.2, and the benefits observed for each company from that company's participation in Eclipse[156]. Figure 18 is a visual timeline of events and Eclipse Foundation membership information for each case company.

The information in these tables is compiled from publicly available sources, triangulated and augmented by observations of activities by company employees in case projects and governance cases, and additional insights from practitioner interviews[157].

[156] The summary information provided in these tables is not intended as a complete business model specification but rather those aspects of the in the Johnson et al. (2008) framework salient to this study.

[157] Table 53, Table 54, Table 55, Table 56 and Figure 18 are compilations of sources and chains of evidence documented in the company case notes documents according to the method outlined in subsection 6.2.5. Sources include company websites, the Eclipse.org website and Eclipse Dash project, meeting minutes of Eclipse Members Meetings. *Abridged minutes of the Eclipse Foundation board of directors*, annual and quarterly corporate reports, independent analyst reports, company white papers, presentations and blogs by company employees, press releases, articles in the trade press, EclipseCon and Eclipse Summit Europe events, observation of the activity of company employees within Eclipse projects and Eclipse governance groups, and practitioner interviews. Unless otherwise stated, information was correct on March 1 2011. In Table 54, the "Committers" row indicates the number of "Total Committers" reported by the Eclipse Dash Project (http://dash.eclipse.org), accessed April 1 2011; this query returns the sum of active committers and inactive committers over the entire lifetime of the database (since 2001). "Rank" is the rank ordering of companies by "Total Committers." Due to known issues with the Eclipse commits database, rankings are approximate only; only ranks in the top twelve are shown in the table. Ranking companies by different metrics (such as currently active committers) provides different results.

Table 53: Comparison of company cases (part 1 of 4): descriptive summary

	Actuate Corporation	EclipseSource	IBM	Innovent Solutions	Intel Corporation/ Wind River Systems	Nokia Corporation	Sonatype	Tasktop Technologies
Primary industries	Business intelligence software.	Enterprise software and services.	Global services; enterprise middleware software; mainframe computers.	Software consulting services.	Semiconductor design and manufacturing.	Mobile devices and services.	Enterprise software.	Enterprise software.
Founding events	• Founded 1993. • IPO in 1998.	• Innoopract founded 1992. • Code 9 founded 2008. • EclipseSource founded 2008 combining Innoopract and Code 9 under one brand.	• Founded 1911 from merging firms founded in 1886, 1889, 1900 and 1901. • Renamed IBM in 1924.	• Founded 1996.	• Intel founded 1968. • Wind River founded 1981. • Intel acquired Wind River in 2009 as a wholly-owned subsidiary.	• Founded 1865. • Incorporated as Nokia in 1871.	• Founded 2007. • VC financed in 2007 and 2010.	• Founded 2007.
Ownership	Public: BIRT (NASDAQ)	Private.	Public: IBM (NYSE).	Private.	Public: INTC (NASDAQ, NYSE)	Public: NOK (NYSE)	Private. VC financed.	Private.
Employees	573	~30	~400 000	~15 (staff and consultants)	82 500 (I) 1 600 (W)	132 430	~50	~25
Revenue (fiscal year ending prior to March 2011)	US$135M	Not provided.	US$95.8B	Not provided.	US$43.5B (I) US$360M (W)	€42.4B ~US$56B	Not provided.	Not provided.

Table 54: Comparison of company cases (part 2 of 4): participation in Eclipse

	Actuate	EclipseSource	IBM	Innovent	Intel / Wind River	Nokia	Sonatype	Tasktop
Membership class	Strategic developer	Strategic developer	Strategic developer	Solutions member	Solutions member	Strategic developer	Strategic developer	Solutions member
Committers	56	13	323	2	22	12	10	6
Rank	2	10	1	n/a	5	11	n/a	n/a
Eclipse Projects	BIRT (lead).	Equinox (lead).	CDT, Equinox, JDT (lead).	BIRT (PMC).	CDT (lead).	CDT.	Equinox/p2 (lead)	Mylyn (lead), Equinox/p2.
(case projects listed first, then other projects)	DTP subprojects (frameworks and tools for data management).	Leadership of many RT and platform projects.	>60 projects including the Eclipse Platform.		TM, TCF (projects for embedded and device software)	TM; TCF (projects for mobile software development)	m2e, Tycho (projects related to Maven build technology)	Marketplace client (MPC); RAP; Virgo.
PMCs	2 BIRT (lead), DTP.	3 Eclipse, RT (lead), Tools.	9 All except Mylyn and SOA.	1 BIRT.	2 Eclipse, Tools.	0	0	1 Mylyn (lead).
Board of directors	Yes (appointed)	Yes (appointed)	Yes (1 appointed and 1 elected)	No	No	Yes (appointed)	Yes (appointed)	Yes (elected)
Architecture council	Yes (3 members)	Yes (3 members)	Yes (11 members)	No	Yes (chair and 2 members)	No	Yes (1 member)	Yes (2 members)
Planning council	Yes (1 member)	Yes (1 member)	Yes (chair and 3 members)	No	Yes (1 member)	Yes (1 member)	Yes (1 member)	Yes (1 member)
Requirements council	Yes (1 member)	No	Yes (2 members)	No	Yes (2 members)	Yes (1 member)	Yes (1 member)	Yes (1 member)
Other	Event sponsor.	Leads long-term support initiative. Training series. Event sponsor. Plug-in central.	Event sponsor. Donor of IT Infrastructure.	Training series. EclipseCon chair.	Event sponsor.		Event sponsor. Donor of build infrastructure.	Eclipse Marketplace Client.

Table 55: Comparison of company cases (part 3 of 4): business model and benefits

	Actuate	EclipseSource	IBM	Innovent
Target customers	Enterprises requiring Business Intelligence (BI) capability, especially companies moving from in-house to managed solutions.	Companies developing enterprise software with Eclipse technology.	Global enterprises requiring integrated solutions: global services (IT and management), software, and electronic hardware.	Enterprises requiring information management solutions for business intelligence, information access and search, and data management.
Value proposition	Improve corporate performance by providing information that can be easily accessed and understood to make better decisions.	Help companies efficiently deploy, manage, and extend Eclipse-based application development platforms and develop on Eclipse frameworks.	IBM Rational: improve speed, quality and efficiency of enterprise software development projects.	Build systems that enable clients to find information, understand its context and meaning, and trust it to be accurate and timely.
Revenue generation	Software products; some services. • **ActuateOne** suite of products for developing and deploying custom business intelligence applications. • Suite of complementary products. • Training and support. • BIRT consulting services.	Services and software products. • Eclipse support. • Sponsored Eclipse development. • Eclipse maintenance. • Eclipse training. • **Yoxos**, a commercially-supported Eclipse distribution.	• Services (42%). • Software (42%), including licenses and support for IBM enterprise middleware products from the **Rational**, **Tivoli**, **Lotus** and **WebSphere** product lines. • Financing (9%). • Hardware (7%).	Consulting services, product sales, and support. • Product-based consulting services (BIRT, other BI products, search). • Reselling of partner products • Support
Required capabilities	• Eclipse BIRT is a key component of the Actuate product line. • Leadership of the BIRT project. • Domain-specific BI expertise. • An ecosystem of partners. • Large and active communities of Eclipse BIRT users and adopters.	• Yoxos is built on the Eclipse IDE and various Eclipse projects. • Leadership of Eclipse runtime and rich client platform projects. • Breadth and depth of technical expertise in Eclipse platform architecture and runtime projects. • Commit privileges to Eclipse code.	• Many IBM software products are built on Eclipse frameworks. • Breadth of product portfolio and global presence. • System integration. • Products that are cross-platform and interoperable with products from other providers.	• Network of partners providing complementary products and services. • Breadth and depth of information management expertise. • Brand and reputation. • Channel to potential customers.
Benefits from Eclipse participation and contribution	• Launch a market discontinuity. • Unique branding and positioning around Eclipse BIRT. • Benefits of open source R&D. • Grow the market of potential customers.	• Enable a business model anchored around Eclipse. • Maintain deep technical expertise. • Obtain unique branding and competitive positioning.	• Benefits of open source R&D practices. • IBM contributors develop deep technical expertise.	• Influence on project plans and roadmaps. • Access to a network of complementors. • Marketing through Eclipse Foundation programs.

Table 56: Comparison of company cases (part 4 of 4): business model and benefits

	Intel (Wind River)	Nokia	Sonatype	Tasktop
Target customers	Providers of electronic systems (Intel), especially mobile and embedded systems (Wind River).	Mobile service providers; end-users of mobile devices; mobile application developers.	Companies developing enterprise software.	Companies developing enterprise software.
Value proposition	Develop, run, and manage device software faster, better, at lower cost, and more reliably.	Connect people to what matters to them by combining mobile technology and services.	Increase software development productivity, improve quality and efficiency, and reduce cost and risk.	Increase software developer productivity with integrated ALM tools and a task-focused interface.
Revenue generation	Intel: • Sales of microprocessors (72%) • Sales of other semiconductors • Other products and services Wind River: • Product licenses (37%) (including **Wind River Workbench** for embedded and device-level software development) • Subscriptions (34%) • Services (29%)	Mobile products and services. • Smart phone sales to end-users and mobile service providers. • Mobile phone sales to end-users and mobile service providers. • Mobile services, applications, and content that enrich the user experience.	Software products and services. • **Nexus Professional** repository manager. • **Sonatype Professional** Java development infrastructure. • **Training** in Apache Maven. • **Support** for Maven-based enterprise build infrastructures • **Consulting services** for Maven plug-in development and build environment customization.	Software products and services. • **Tasktop Pro** application lifecycle management (ALM) product (for Eclipse or stand-alone). • **ALM connector plug-ins** for Tasktop and Eclipse Mylyn. • Eclipse **development** services. • Development **consulting** services to integrate other ALM software systems with Mylyn and Tasktop. • **Training** in Mylyn and Tasktop.
Required capabilities	• Wind River Workbench is built on the Eclipse IDE and CDT. • Technologically advanced software components for developing software on current and future semiconductor devices.	• Nokia tools, including **Carbide** and **Web Tools for Series 40 phones**, are built on Eclipse. • Cross-platform software technologies for Nokia phones. • A competitive ecosystem of mobile application developers.	• Products are built on open source software from Maven, Eclipse, Hudson, and projects. • Domain-specific build expertise. • Influence over the development of critical software components. • Reputation within communities.	• Tasktop is built on Eclipse Mylyn. • Expertise in software ALM and developer productivity. • Deep technical expertise in Eclipse architecture and projects. • Leadership of Mylyn project. • Commit privileges to Eclipse code.
Benefits from Eclipse participation and contribution	• Project features, extensions and capability that would not otherwise exist. • Accelerated development of company products.	• Project features, extensions or capability that would not otherwise exist. • Influence over project direction. • Benefits of pooled R&D.	• Strengthen a business model anchored in open source software. • Benefits of open source R&D. • Awareness and goodwill within developer and user community.	• Enable a business model anchored around Mylyn. • Benefits of open source R&D. • Unique branding, competitive positioning. • Visibility within Eclipse community.

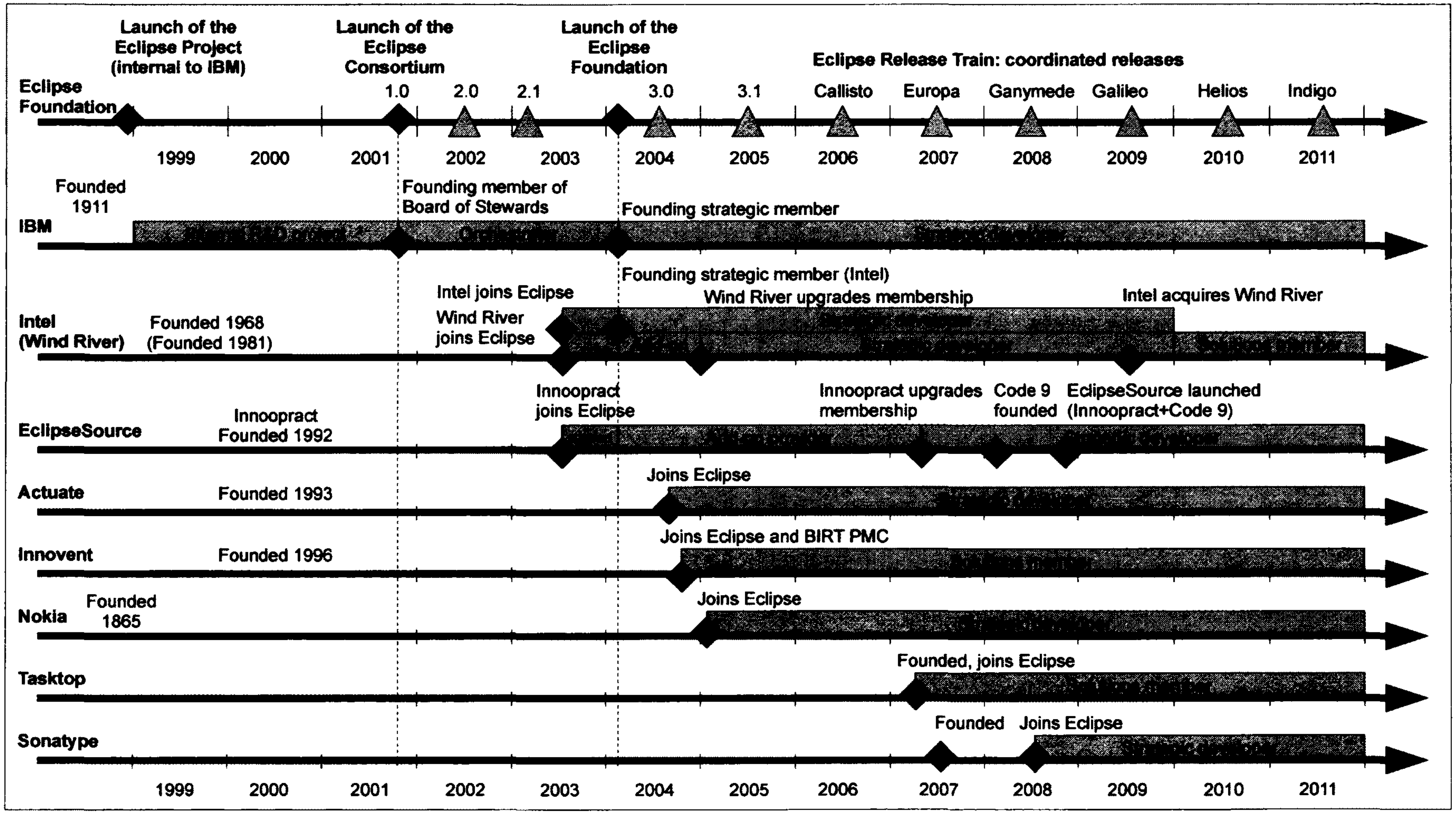


Figure 18: Timeline of company cases

The following subsections elaborate on information presented in the preceding tables and figures – specifically the various ways in which companies participated in the Eclipse field setting, the ways in which companies contributed to the Eclipse platform, the ways in which companies joined projects in which they were not already active, and the benefits that companies enjoyed from their participation in Eclipse.

10.2.1 Participation in the Eclipse field setting

The eight case companies participated in Eclipse in various ways. All were users of Eclipse technology. A typical practitioner interview response is the following:

> *The most important benefit of being involved in Eclipse is access to high-quality Eclipse software.*

Most case companies also consumed Eclipse technology within a product offering built-on Eclipse tools or frameworks. In some companies – most notably Actuate, Wind River (a subsidiary of Intel)[158], Sonatype, and Tasktop – sales of products built on Eclipse technology are a main revenue trigger for the business unit. For Nokia and the Intel semiconductor business, Eclipse-based products and extensions are a complement to the core business, enabling more sales of phones and semiconductor chips, respectively, and increasing the perceived value of each company's core offerings to its target customers; neither company charges for their Eclipse-based products, instead distributing them at no cost. For EclipseSource, a commercially supported Eclipse-based product offering is one revenue stream within a diverse Eclipse-based service and product business.

Most of the case companies also consumed Eclipse technology by offering

[158] In this discussion, it is important to distinguish between the Wind River business of products and services for developing embedded systems software, and the much larger Intel semiconductor business. Both are users and consumers of the same Eclipse technology, but in very different ways. Formerly separate companies, Intel acquired Wind River in 2009 and operates it as a wholly-owned subsidiary.

Eclipse-based services. For EclipseSource and Innovent, Eclipse-based service offers are a main revenue trigger for the business. Each company focuses on a particular domain – runtime and rich client applications, and BIRT-based information management systems, respectively. Sonatype and Tasktop each entered their respective markets through service offerings (anchored around Apache Maven build technology and integration of third-party software systems with Eclipse Mylyn, respectively), then subsequently launched Eclipse-based product offerings as a primary growth strategy for future business.

The case companies all contributed to the Eclipse platform by assigning staff to the Eclipse community – as contributors, committers, project leads, PMC members, and PMC leads. Some companies focused their contributions exclusively on one specific project or within a niche of closely related projects, and others distributed their contributions more widely. Some companies led community-driven projects (e.g., Intel on CDT, or Sonatype on Equinox), and some companies led projects and employed the majority of developers of those projects (e.g., Actuate on BIRT, or IBM on JDT). One case company (Innovent) was active within a PMC without actively committing code.

The case companies were all members of the Eclipse Foundation at either the strategic or solutions level, contributing membership dues and a public signal of endorsement of the Eclipse brand. Some were active in Eclipse governance – the board of directors and the three governing councils examined in chapter 9. Strategic members (Actuate, EclipseSource, IBM, Nokia, and Sonatype) are entitled to directly appoint representatives through their membership powers. Of the solutions members, the Tasktop CEO sat on the Eclipse board of directors as an elected representative, and both

Intel and Tasktop had representatives on all three governance councils.

Other observed forms of participation and contribution included sponsorship of Eclipse community events (both large shared-sponsorship events such as EclipseCon and exclusive sponsorship of smaller events such as demo camps), participation in Eclipse Foundation programs (such as the Eclipse Training Series or the Eclipse Members Distros program), and leadership of strategic initiatives within the Board of Directors and the community (such as an initiative to provide multi-vendor long-term support for legacy Eclipse releases led by EclipseSource and SAP; Krause & Schmidt, 2010). IBM is major donor of IT infrastructure, and Sonatype donates software infrastructure for the Eclipse build process. EclipseSource and Tasktop have both developed software assets for the foundation. A co-owner of Innovent Solutions recently served as the program chair of EclipseCon 2009 – a major commitment of time and effort and a visible and valued contribution to the community.

10.2.2 Contribution to the Eclipse platform

Within the case data, companies active within Eclipse projects contributed code to those projects in one or more of at least four different ways in addition to assigning developers to work on a project. First, some companies took a leadership role on a particular feature or desired capability. Committers employed by the company developed code within the project according to the Eclipse Development Process, community norms, and local project conventions, as described in chapter 8. By actively contributing to that area of the code, those committers became recognized as technical experts of that code, called on to vet bug reports and community contributions. For example, Wind

River identifies leadership in multicore software development as one of three key initiatives, and committers employed by Wind River are active within the Multicore Debug Working Group to extend multicore capability within the Eclipse CDT project[159].

Second, some companies developed code outside of the Eclipse infrastructure, but developed it in the open with the stated intent to contribute the code in the future. An example is Nokia's development of EDC – the Eclipse debugger for C/C++. In early 2009, Nokia began development of a new debugger for its Carbide.c++ tools for Symbian OS development, a software application built on software from the Eclipse CDT project. Nokia assigned nine developers to work on the internal project, two of whom were committers to CDT. The Nokia project lead posted updates to the CDT mailing list and blogged about ongoing development. In December 2009, after using the software internally, Nokia committers contributed the EDC source code to the CDT project, and began moving bug reports, developer discussions, and other activity from the Nokia infrastructure to the CDT project infrastructure to continue development there. The Nokia project lead explains the team's motivation (Ryall blog, December 2 2009):

> *While we wanted to develop EDC in the open we also wanted to get enough of the basics working so that when we contributed the code people would be able to easily try it out. So while we worked on all of the basic things a debugger should do we tried to keep the CDT community informed about our progress. Now that the starter code is in we'll be making our continuing work more visible, moving discussions into the Eclipse Bugzilla from our internal one. There is still a lot of work to do but now we're in a position to do it in collaboration with the community.*

By January 2010, committers employed by other companies had began developing

[159] Key initiatives are from the Wind River corporate profile (Wind River website, http://www.windriver.com, accessed March 2011). The CDT Multicore Debug Working Group meets by conference call every two weeks with meeting minutes posted to the Eclipse wiki (http://wiki.eclipse.org/CDT/MultiCoreDebugWorkingGroup).

extensions to the EDC code. Between February and April, four other Nokia developers from the EDC project had been nominated and elected as CDT committers, bringing the number of Nokia-employed committers to six.

Third, some companies contributed code that had been developed internally for the company's business needs and its own use and consumption. An example is code donated by Ericsson that became the core of the Mylyn Reviews project – a subproject of the Eclipse Mylyn project launched in February 2010 during the study interval. Ericsson had originally developed this code for consumption within its own internal software R&D process. By contributing the code to Eclipse, the Ericsson R&D process could also benefit from community bug reports and extensions developed by others, including employees of Tasktop and SAP who were also committers on the Mylyn Reviews project.

Fourth, some companies contributed code developed for a customer. Both Tasktop and EclipseSource develop Eclipse extensions for customers as a component of their product and service offers. Tasktop "Eclipse development services" are typically Mylyn-related application lifecycle management extensions, and EclipseSource "sponsored Eclipse development" services are typically extensions to Eclipse runtime projects or the Rich Client Platform. Both companies are well-positioned to commit appropriate extensions to the Eclipse platform source code, and encourage their customers to allow them to do so. Employing developers with Eclipse commit privileges is part of each company's value proposition and its differentiation from competitors.

In summary, there are various ways in which participating companies develop and contribute code to Eclipse projects.

10.2.3 Entry strategies

The companies examined here employed one or more of five different entry strategies for joining Eclipse projects[160]. Each entry strategy is described briefly below.

The first entry strategy is *launch and lead a new project.* Actuate founded the BIRT project in 2004 and contributed the initial code; it shares PMC leadership with Innovent and IBM, but continues to lead technical development and community-building by employing most of the active committers and sponsoring the *BIRT Exchange* community. Sonatype launched the m2eclipse and Tycho projects in 2008 and 2010, respectively; both are built on code contributions developed initially by Sonatype staff outside. Mik Kersten launched the Mylyn project – then called Mylar – in 2005 to implement the task-focused interface that was the subject of Ph.D. thesis at the University of British Columbia (Kersten, 2007), and then founded Tasktop Technologies in 2007 to provide products and services built on Mylyn.

The second entry strategy is a *supporting role* in the launch of a new project. Actuate launched the BIRT project in 2004, developed the initial code, and continued to perform the core development. Scott Rosenbaum, VP of Engineering at Innovent Solutions, was invited to join the PMC later that year. Innovent was primarily an adopter rather than developer of BIRT technology, and as such, provided diversity of perspectives on the PMC. There are strong parallels here with notions in the innovation literature of "bringing customers into the board room" (McGovern et al. 2004) and involving lead users in product design (von Hippel, 1988).

[160] These examples draw on observation of the case companies from this chapter and on company participation in the case projects from chapter 8.

The third entry strategy is *work up through the meritocracy* – assign staff to contribute to a project, earn the trust of the community over time through high-quality contributions, and eventually become committers on the projects. In early 2010, a group of developers at SAP approached the Equinox p2 committers about SAP requirements for a provisioning system with tight space constraints – requirements that differed from those of current p2 users and adopters, but which might be appealing to a different audience. Several SAP developers became committers in the Equinox incubator, separate from the main code base, for possible graduation in the future. After contributing a series of related refactoring patches to p2, an SAP developer was elected to become a full p2 committer in March 2011 to continue refactoring work on the main code base.

The fourth entry strategy is *hire existing committers*. In February 2010, Sonatype hired the project lead of the Equinox p2 project who was previously employed by IBM. Roles within the Eclipse developer community meritocracy attach to the individual rather than the employer, placing Sonatype immediately in an influential position on the project.

The fifth entry strategy is *code contribution*. In March 2009, the Mylyn Connector Discovery code moved from the Mylyn project to the Equinox p2 project through a move review. The developers of that code, both committers on the Mylyn project and employees of Tasktop, were nominated as committers to p2 and the Equinox incubator projects so that they could continue to maintain and extend their code.

Project entry strategies are analogous to the market entry strategies of the management literature (e.g., Hultink et al. 1997). Companies that sought to join an Eclipse project selected a mode of entry that best matched their business needs.

10.2.4 Benefits of participation

The case companies enjoyed a wide variety of benefits from their active participation in Eclipse. Although anyone can use and consume Eclipse platform assets within the rules of the *Eclipse Public License*, these case companies variously benefited in additional ways (Table 55 and Table 56). There are three common themes.

First, *each case company benefited in ways that strengthened its business model.* More specifically, each company's participation in Eclipse enabled access to capabilities required to deliver on its value proposition and earn attractive profits. The specific capabilities that a particular company required depended on its context and other aspects of its business model, including its value propositions to stakeholders, its means of generating revenue, its profit expectations, and its cost structure[161]. Some examples include deep technical expertise, unique branding and competitive positioning, visibility within the Eclipse community, participation in optional members programs, and access to a network of complementors. In general, companies can and do acquire access to required capabilities in many different ways, including internal development, acquisition, partnerships, service contracts, and so forth (Powell, 1990; Culpan, 2009). The traditional strategy literature emphasizes ownership and control, but these case companies instead acquired some capabilities through what Chesbrough & Appleyard (2007) call *open strategy*, of which participation in Eclipse was one component.

Second, companies with active committers on projects, who consumed those same projects as inbound open innovation on which to build the company's products, services,

[161] For clarity and consistency, discussion of company business models continues to refer exclusively to the modified Johnson et al. (2008; Johnson, 2010) business model framework developed in chapter 4.

and technologies, benefited in the myriad of ways previously documented in the open source software and community innovation literatures (section 3.6; see also sections 3.1 and 3.2). Companies contributing to community-driven projects reduced development costs through pooled R&D (West & Gallagher, 2006) while benefiting from the contributions and cumulative innovations (Allen, 1983; Scotchmer, 2006; Fiona & O'Mahony, 2007) of others. On both community-driven and company-driven projects, companies introduced the upside potential of user innovation (von Hippel, 1988, 2005) through patches, bug reports, and other contributions. Several practitioners stated strong beliefs that open source development practices can produce higher quality software than closed practices. These well-known benefits of open source R&D and community innovation are in no way unique to Eclipse, but they are magnified by the depth and breadth of the Eclipse user, adopter, and developer communities. Likewise, moving an externally-developed open source project to Eclipse is a possible legitimation strategy to shed the appearance of control by a sponsoring company or to distinguish a project from its rival projects and make it more attractive for company adoption and contribution.

Third, *participation can enable new business models that were not previously viable.* This is a stronger claim than the first theme. Some case companies did more than strengthen existing business models – they transformed their business models or invented new business models enabled by Eclipse participation. The most scrutinized example is IBM and its launch, orchestration, and continuing participation in the Eclipse Consortium and Eclipse Foundation (examined in section 7.3) – actions entirely consistent with IBM's larger reinvention around interoperability, integration and global professional services (examined in section 7.2). For IBM, Eclipse was at once a competitive weapon (attacking the revenue and differentiation of competitors through commoditization), a means to drive user adoption of favourable standards, an order-of-magnitude cost restructuring from proprietary closed innovation, and a strategic shift in revenue trigger and competitive advantage from a one architectural layer to another (West, 2003; Capek et al. 2005; O'Mahony et al. 2005a, 2005b; Samuelson, 2006). Actuate's launch of the BIRT project appears similarly motivated. In 2004, Actuate was a provider of proprietary enterprise software and services in a crowded business intelligence industry segment with many competitors. Launching the open source BIRT project within the Eclipse Foundation (rather than elsewhere, or independently), alongside IBM and others, provided legitimacy and a template from which to follow for driving adoption and building user and adopter communities. Actuate could then focus its business on moving companies from in-house solutions built on Eclipse BIRT to managed integrated solutions provided by Actuate – another example of providing the "whole-product" (Moore, 1991) along with differentiating features and services built on a freely available platform (a practitioner explanation of Eclipse examined in section 7.7). Tasktop's business model is anchored around leadership of the Eclipse Mylyn project, and the resulting advantages at providing managed Mylyn solutions, extensions, and services relative to other vendors not deeply involved in the Mylyn project. EclipseSource anchors its business around providing Eclipse-based services and products, enabled by deep involvement in Eclipse projects and Eclipse governance, dating back (for at least one founder) to IBM and OTI in the late 1990s. Jeff McAffer, Chief Technical Officer of

EclipseSource, explains his motivation for founding Code 9 and for joining Code 9 with Innoopract to create EclipseSource as a way for companies to get code into Eclipse platform and runtime projects:

> *Most of the significant players in the runtime world are very big companies: Oracle, IBM, VMWare, and so forth. Most of the people with influence and impact in the Eclipse runtime community work for these big companies. If you're a small firm who wants to adopt Eclipse, you can't pay IBM to do things for you – not necessarily because you don't have any money, although that might be the case anyway – but because IBM is not structured to take your money. The people in IBM that have Eclipse expertise are not connected with the consulting part of IBM, and the people at the IBM consulting group who could take your money don't have the Eclipse expertise to help. That connection is missing, so there is no one there who you can pay to help.*
>
> *That's one of the things that I thought I could enable. I could help people with a latent need to impact Eclipse by giving them a way into the Eclipse community. That's similar to what Innoopract was doing and that's why we decided to merge. We had compatible business models and goals, and complementary domain knowledge and expertise. The projects that we worked on were complementary but not overlapping.*

IBM, Actuate, Tasktop, and EclipseSource are illustrative examples of companies – large and small, incumbent multinational enterprises and nascent new ventures – inventing new business models made viable by participation in the Eclipse field setting.

10.3 Synthesis and theory development

The analysis and discussion of the company cases provides the information required to complete the IAD table shell for company participation in the Eclipse ecosystem (Table 57).

Table 57: The institution of company participation (March 2011)

IAD Element	Description
Underlying factors	
Rules-in-use	Activity within the Eclipse ecosystem is structured by (1) property rights and legal frameworks governing Eclipse platform assets, (2) the *Eclipse Membership Agreement*, and (3) opt-in programs for Eclipse Foundation members. Property rights are summarized in Table 29. Salient attributes of the Eclipse Public License (EPL) include weak copyleft and patent licensing terms to protect users. Opt-in programs include industry working groups, the member distros program, the Eclipse Training Series, long-term support, many others (subsection 9.4.4). Enforcement mechanisms include legal enforcement of property rights (litigation or threat of litigation) and termination of membership.
Constituent attributes	Companies conducting business within the converged information and communication technology industries. Technology-intensive. Rapid pace of technology and market change.
Resource attributes	*Artifacts*: open source software tools, frameworks, and runtimes on a modular plug-in architecture, distributed as source and packages; complementary assets. • Released software is vetted through the Eclipse IP due diligence process. *Facilities*: an Eclipse project IT infrastructure including source code repository, developer mailing lists, user forums, Bugzilla issue tracker, and other assets. Membership services provided by the Eclipse Foundation: e.g., public relations and promotion of the Eclipse brand, the Eclipse Marketplace Client (MPC), Eclipse events, support for press releases, members meetings, research reports, etc.
Action arena	
Actors	Profit-motivated economic actors: for-profit companies (and other organizations and independent individuals) conducting business transactions with products, services, and technologies anchored in the Eclipse platform.
Action situation	Participation in the Eclipse business ecosystem. One or more participation modes: • Use Eclipse tools (user). • Produce market offers that consume Eclipse tools and frameworks (adopter). • Join the Eclipse Foundation (member; four tiered membership classes). • Contribute to the platform, indirectly by allocating resources to the Eclipse development community, or directly by code donation. • Participate in governance (elections; appointments to councils).
Interaction patterns	Coopetition: simultaneous competition and cooperation.
Outcomes	Ecosystem health: • *Productivity* of innovative output. • *Robustness* to external shocks. • *Niche creation* of new initiatives and industry verticals.
Evaluative criteria	High health metrics.

The contents of this table are compiled from sources cited throughout chapter 10 and chains of evidence documented in company case notes documents.

The following discussion begins at the right of the IAD framework (chapter 5, Figure 2) in the same way and for the same reasons as the previous discussion of the developer community in chapter 8. It begins at *outcomes* and works systematically back through the *action arena* to the *rules-in-use* – from bottom-to-top of Table 57.

The action situation of interest is *participation in the Eclipse ecosystem*: that is, the various escalating decisions taken by companies and other economic actors to voluntarily enjoin the outcomes of their own business activities with the outcomes of the Eclipse ecosystem – producing market offers that consume Eclipse tools and frameworks, joining the Eclipse Foundation, assigning development resources to Eclipse projects and the Eclipse developer community, contributing assets to the Eclipse platform, and participating in Eclipse Foundation governance. The principal interest in this present analysis is *deep* participation in the Eclipse ecosystem, in ways that advance the Eclipse platform as an open source software commons freely available to all (section 7.8.4).

The outcomes and evaluative criteria of interest are those employed by the Eclipse Foundation to assess ecosystem health (Smith & Milinkovich, 2007; Smith blog, October 2 2006, May 2 2007; Milinkovich, 2008a): (1) *productivity* (how much value is being created by the ecosystem), (2) *robustness* (how durable is the ecosystem and how adaptable is it to external events), and (3) *niche creation* (the ability to expand the ecosystem with meaningful diversity) (adapted from Iansiti & Levien, 2004a, 2004b).

Actors can be companies or other economic actors. Consistent with the case data, actors are said to be *profit-motivated*, although not necessarily profit-maximizing in the technical sense of neoclassical economics. The pattern of interactions between economic actors is *coopetition* (Brandenburg & Nalebuff, 1996) – simultaneous competition and cooperation. For example, three of the most active companies in Eclipse are IBM, Oracle, and SAP – fierce competitors in their product markets, but active together on Eclipse governance and collaborators on some Eclipse projects. Committers from many of the case companies examined here participate in Eclipse projects along with committers employed by direct competitors, such as Intel in the semiconductor space along with Freescale Semiconductor, ARM, STMicroelectronics and Texas Instruments, and Nokia in the mobile devices space along with RIM, Motorola, and Google. Competing Eclipse enterprise distributions by Tasktop, Sonatype, and EclipseSource have overlapping value propositions and target markets.

Participation in the Eclipse ecosystem is tiered: companies decide not only whether or not to participate, but also how much and in what ways. At the lowest levels of participation are the ecosystem participants who use and adopt Eclipse platform assets. More active ecosystem participants may become members of the Eclipse Foundation, participate in opt-in membership programs, contribute to the platform directly and indirectly, and become active in foundation governance.

The rules-in-use structuring company participation can be usefully organized into three categories. The first category is property rights and the legal framework governing Eclipse platform assets: the *Eclipse Public License (EPL)*, *Guidelines for Eclipse Logos & Trademarks* (2009), and other rules outlined in chapter 7 (especially subsection 7.8.4 and Table 29). Platform property rights are particularly salient to ecosystem companies, because they determine, in part, what companies can and cannot do with platform assets and the products and services built on those assets. As argued in chapter 7, the Eclipse property rights are deliberately crafted with intent to promote adoption and contribution.

The second category, the *Membership Agreement*, is a bundle of rules applicable to Eclipse Foundation members. It specifies the rights and obligations of membership, including the membership powers previously summarized in Table 22 (section 7.1) and revisited in section 9.8. The membership agreement structures membership into tiers; higher levels of membership provide greater benefits at higher costs. Different membership classes are attractive to different participants with varying business needs.

The third category is *opt-in programs* for Eclipse Foundation members. These programs, operated by Eclipse Foundation staff and members of the community, include industry working groups, the Member Distros program, the Eclipse Training Series, and most recently, the Eclipse long-term support program (Krause & Schmidt, 2010). Like the opt-in project programs of Table 44 for Eclipse projects, the opt-in member programs are voluntary bundles of rules-in-use that provide potential benefits to participants. The operation of members programs by Eclipse Foundation staff was described in chapter 9.

The commons resource of the ecosystem is the Eclipse platform (specified in subsection 7.8.4 and Table 29), and two aspects are particularly salient to company participation. The first aspect is the Eclipse IP due diligence process, providing assurance to ecosystem participants that released platform software has been contributed according to the *Eclipse.org Terms of Use* and *Committer Agreement*, and vetted by the Eclipse IP team. The second aspect is the membership services provided by the Eclipse Foundation. Case evidence suggests that both are valued by some ecosystem companies.

10.4 Tensions and tension management

Section 7.8 previously argued that the Eclipse community and Eclipse ecosystem each depend on the other for critical resources, both directly and mediated through the platform. The aspired outcomes of a "thriving, diverse and active community of developers" (EDP 2.5, section 2.3) and a vibrant, growing, healthy ecosystem (Smith & Milinkovich, 2007; Smith blog, May 2 2007; Milinkovich, 2008a) are co-dependent. As an illustrative example, Ian Skerrett, Eclipse Director of Marketing, explains:

> *In many conversations, I can use the words "community" or "ecosystem" interchangeably because either word will often work just as well for the point I'm making. The community and the commercial ecosystem have much in common, and a lot of what the Eclipse Foundation does – marketing, IT, IP, and so forth – benefits both.*

Nonetheless, the conceptual framework of section 7.8 and subsequent refinements in chapters 8, 9 and 10 also show that the community and ecosystem do differ in some important ways, including the nature of the actors, actor motivations and actor decision processes, emphasis of creation or commercialization aspects of innovation, the ways in which they interact with the Eclipse platform, and the strength and mechanisms of influence on the Eclipse Foundation. Preferences regarding the assignment of costs and division of benefits between the community and ecosystem may differ. Indeed, given such deep differences between constituents and interaction patterns, it would be remarkable if there were no tensions between the community and ecosystem.

Within the case data of this research, some observed areas of ongoing tension between the Eclipse community and ecosystem are (1) the "right" amount of process (including management of intellectual property), (2) how to resolve discrepancies between rules-in-use and rules-as-written and enforcement of obligations, and (3) reconciling transparent and open governance with effective governance by a "traditional" board of directors. In each case, the Eclipse Foundation manages these tensions through specific mechanisms.

A first area of tension is the "right" amount of process. Regarding intellectual property, Executive Director Mike Milinkovich (2010, p. 30) writes:

> *There is no doubt that the focus on commercialization places added burdens on Eclipse projects. Our development and intellectual property processes require real work to comply with. But there is value in that labour, and the value is in the added use, adoption, commercialization and plain old respect that that the Eclipse brand brings to a project.*

Project committers undertake much of the work required by the IP policy, and the immediate direct beneficiaries are the ecosystem companies that commercialize Eclipse software, some of whom contribute resources to the community and some of whom do not. The IP Policy is one area the rules that is strictly enforced: releases cannot contain unapproved code, projects cannot release code without a release review, and a release review will not pass without full compliance with the IP policy (or with exceptions specifically approved for that project by the board of directors). IP activities consume community time, effort, and attention from other tasks, such as community-building or advancing and maintaining platform, that may be more appealing to committers. Practitioner discourse during community events implies that at least some developers consider IP due diligence requirements as an economic cost when assessing and selecting design alternatives. Community discussion around the right level of process is particularly evident around creation, provisioning, and incubation of new Eclipse projects originally started elsewhere. Two examples are Oracle's recent donation to Eclipse of the Hudson continuous integration server, and the move of the EGit and JGit projects to Eclipse to provide tight integration for the Git distributed version control system.

The Eclipse Foundation appears to manage this tension through at least two mechanisms: the first mechanism is ongoing communication and active dialogue about the policy and its rationale (e.g., Campbell 2007, 2009; *Committer Guidelines*, 2008; *New committer handbook*, 2011; *Words of wisdom and bits of advice*, 2011), and the second is initiatives to improve the efficiency of the process and to lessen the time and effort required for compliance. An early initiative was the Parallel IP Process for incubating projects, first introduced in 2007, to allow conforming projects to check-in code to an Eclipse repository then complete IP due diligence in parallel, rather than requiring due diligence to be completed before checking in code (Milinkovich blog, January 22 2007; *Parallel IP process*, 2010; see also subsection 8.2.2). A recent initiative is Eclipse Labs, introduced in 2010 as a hosted infrastructure for Eclipse-related projects developed outside the rules of the Eclipse Foundation (Milinkovich blog, May 13 2010; see also subsection 9.4.1). Projects at Eclipse Labs are not official Eclipse Foundation projects, and have neither the benefits nor the obligations of the Eclipse Development Process. This has been a popular initiative with the community.

A second area of tension is how to best resolve discrepancies between rules-in-use and rules-as-written. This tension also relates to the enforcement of rules within the community and ecosystem. The underlying question is what should happen when actual behaviour differs from the rules-as-written. Should the rules be revised to more accurately reflect practice? Should the rules-as-written be enforced through whatever enforcement mechanisms are available? Or should the rules be better understood as guidelines, so that discrepancies are both normal and expected? This becomes an area of tension between community and ecosystem when preferences differ or if enforcement is perceived as uneven. The board of directors periodically discusses enforcement of board resolutions concerning projects (e.g., *Abridged minutes of the Eclipse Foundation board of directors*, 2008Q2, June 18-19) and enforcement of membership obligations for strategic developers (e.g., October 15 2008), and it is a common theme in community discourse. Section 9.9 previously noted the diversity with regards to community comments on the first revisions to the Eclipse development process in 2007. The IP policy discussed above is an example of a bundle of rules that are tightly enforced. In other matters, the Eclipse Foundation has instead revised rules to bring them into better alignment with rules-in-use that are working effectively. "Synchronize the process as-written and as-practiced" was an explicit objective to the revisions to the Eclipse Development Process in 2007 (*Abridged minutes of the Eclipse Foundation board of directors*, Q42006, December 13-14, pp. 13-15) and in 2008 and 2010 (see also section 9.10). When neither the rules-in-use nor the rules-as-written have been deemed effective, the Eclipse Foundation has experimented with introducing new rules developed in collaboration with the community and ecosystem. An example is the re-purposing of the architecture council in 2007 from a planning to a mentoring function (section 9.5).

In summary, the Eclipse Foundation has managed this tension through three mechanisms. First, where the rules-in-use are deemed effective, Eclipse governance groups revise the rules-as-written to better reflect the rules-in-use. Second, where the rules-in-use are not deemed effective, Eclipse governance groups introduce new processes developed in collaboration with the community and ecosystem. Often these are staged initially as controlled experiments, then rolled out more broadly. Third, the foundation fosters active public conversations, on Eclipse community channels, and other communication channels outside of Eclipse.

A third area of tension is between effective governance of the Eclipse Foundation by the board of directors, and transparency and accessibility to the Eclipse community. Both are legal obligations – the first embedded in corporate law and in the Eclipse Foundation bylaws, and the second in the bylaws, EDP, and culture of Eclipse. Executive Director Mike Milinkovich writes (Milinkovich blog, November 30 2009):

> *I struggle walking the line between transparency, and giving the Board the opportunity to properly review and approve the plans coming from the staff. I do get it wrong upon occasion."*

Board of directors meetings are one of the few areas of Eclipse that are not open to direct observation (section 9.2). Abridged minutes for quarterly face-to-face meetings are published after approval at the following quarterly meeting, without disclosing detailed voting information, the sources of comments, or detailed financial information. There are normally no public minutes for monthly conference calls, unless important resolutions were passed on the call. Voting power rests, collectively, with the strategic member organizations which together hold approximately two-thirds of the votes.

The Eclipse Foundation appears to manage this tension through four mechanisms. First, the formal rules of the Eclipse *Bylaws* (examined in section 9.2) ensure diversity of representation on the board, with seats allotted to elected committers and all representatives from all membership classes. Second the formal rules limit the influence of any particular organization through several mechanisms, so that no single company can dominate. Third, the abridged board minutes are quite detailed, appearing to contain as much information as possible without revealing information deemed to be sensitive or problematic for open discussion according the rationale articulated in section 9.2. Fourth, the Executive Director brings matters forward to the community promptly. For example, discussions surrounding the proposed 2011 amendments to the Eclipse *Bylaws* (discussed in chapter 9) began at EclipseCon 2011 with the governing councils (a meeting on Sunday afternoon), the board of directors (the Q1 meeting on Monday morning), and the membership-at-large (the annual general membership meeting on Monday afternoon). The Executive Director writes (quoted from Simon Phipps blog, February 16 2011):

> *Although some may question the notion that there is a meritocracy involved where the Board has many corporate members who are there by virtue of their financial and resource commitment to the community, in practice this works extremely well. What we have ended up with is a mix of large and small companies who are strategically committed to the success of the community. This commitment is tangible and re-evaluated annually.*

Reflecting on these observations, it may be some tension between community and ecosystem is an intrinsic system property of deliberately cultivating and governing both collectives within a single foundation. However, the Eclipse Foundation appears to successfully manage and harness those tensions through the combination of (1) transparency of decisions and decision-making, (2) frequent and ongoing conversation, and (3) deliberately attending to the interests of both the community and ecosystem without favouring one at the expense of the other, as well as actions specific to each tension, such as initiatives to automate and otherwise improve processes, revision to re-align the rules-as-written with effective rules-in-use, and enforcing formal rules to limit the extent of influence of any single company. Some participants noted that tensions also occur elsewhere throughout the field setting. Don Smith, Eclipse Foundation Director of Ecosystem Development, explains:

> *There can also be plenty of tension within the community and within the ecosystem. Developers don't all think alike, and companies don't all want the same things. Also, disagreements that appear to be between people in the community may actually reflect the different positions and interests of their organizations.*

This chapter examined the activities of companies within the Eclipse ecosystem in order to specify the IAD characteristics of the institution of company participation, and explore the benefits, incentives, and motives of company participation; it completes the four-chapter presentation of the research results at each of the four levels of case analysis. The next chapter discusses these results and their contribution to theory and practice.

11 Discussion

This chapter addresses the research questions, articulates the theoretical contribution of this work, discusses alternative formulations and constructs, compares and contrasts these results with the findings of the commons literature, and discusses opportunities for future research. Collectively, the sections comprising this chapter interpret and assess the research results, and position those results with respect to what was previously known.

Previous scholarly studies of business ecosystems have typically emphasized one or more of four different conceptual lenses. An *industry structure and technology platforms* stream relates the properties of the business ecosystem to the properties of the underlying technology platform and the strategic choices of "platform leader" firms who are advantageously positioned to direct the development of the platform and appropriate value (Baldwin & Clark, 2000; Messerschmitt & Szyperski, 2003; Gawer, 2009a; Baldwin & Woodard, 2009; Cusumano, 2010; see also literature reviewed in sections 3.3 and 3.4). An *open innovation* stream examines the motivations, strategic choices, and business models of companies who open up their R&D process to purposeful inflows and outflows of ideas and technology from a "value network" of suppliers, customers, complementors, and competitors (Chesbrough & Appleyard, 2007; West & Wood, 2008; see also section 3.2). An *innovation communities* stream expands and generalizes the community construct to admit participation of organizational members and to interpret the cooperative and competitive actions of ecosystem companies within the community context (Lynn et al. 1996; von Burg & Kenny, 2003; Miles et al. 2005; von Hippel, 2005; West & Lakhani, 2008; Snow et al. 2009; see also section 3.6). An *innovation networks* stream focuses on the network properties of the relationships between companies and technologies (Iansiti & Levien, 2004a; Iyer et al. 2006; Weiss & Gangadharan, 2010; see also section 3.5). These perspectives are complementary; each brings into sharp focus some aspects of the business ecosystem phenomenon that are unaddressed or under-addressed by the other theoretical perspectives. By emphasizing the rules-in-use that structure action (North, 1990; Ostrom, 2005), the institutional perspective of this research attends closely to the differing institutions of individual participation and of company participation, the relationships between the open source developer community of individual contributors and the business ecosystem of economic actors, the dual role of the governing foundation as both an open source foundation (O'Mahony, 2005, 2007; O'Mahony & Ferraro, 2007; West & O'Mahony, 2008) and a business ecosystem keystone (Iansiti & Levien, 2004a, 2004b) and the tensions inherent in bringing together those roles – aspects of the business ecosystem that enrich and complement those of the other perspectives. The literature on commons governance (Ostrom, 1990; Hess & Ostrom, 2007) provided a vocabulary and framework for data collection, analysis, and comparison of Eclipse institutions with other systems, and a view of the Eclipse platform as a commons resource vulnerable to commons dilemmas. The resource dependence literature (Pfeffer & Salancik, 1982), enfolded after research was underway, provided a vocabulary for specifying and explaining co-dependent relationships between parts. The practitioner and nascent scholarly literature on architectures of participation (O'Reilly, 2005a; Baldwin & Clark, 2006; West & O'Mahony, 2008) provided conceptual building blocks for theorizing about systems of interconnected institutions of participation.

Likewise, previous empirical studies of distributed innovation have typically examined either a value-creating community or a value-appropriating ecosystem of companies, but not both systems together. By addressing a broader range of phenomena within the scope of one field study – a field setting including a developer community, a commercial ecosystem, a platform commons, and governance structures – this research has examined interactions *between* structures that were outside the boundaries or at the boundaries of previous studies.

Collectively, the outcomes from the use of an alternative conceptual lens and a broader range of phenomena include a deeper understanding of the Eclipse ecosystem, and contributions towards a theory of participation in community-developed platforms. These outcomes have practical implications both for management practice and for management research.

The next section begins the discussion by presenting answers to the first of two guiding research question that motivated this research.

11.1 What is the Eclipse ecosystem?

The first guiding research question is answered in four parts, each corresponding to one of the top-level case study questions identified in section 6.2. Each answer provides cross-references back to previous sections of this document where the salient case evidence was presented and the theoretical arguments first developed, often in greater detail than the summary explanations provided in this section.

Question 1a: What does "Eclipse ecosystem" mean to Eclipse practitioners?

A thick scholarly literature in the social sciences warns of the imprecision and malleability of natural language: two interlocutors who employ the same term but attach different meanings to that term, either knowingly or unknowingly, may not be conversing about the same thing. A first step to describing and explaining the Eclipse ecosystem is examining what that term means to Eclipse practitioners. From triangulation of case evidence and interview responses, section 7.6 developed the following definition:

> *The Eclipse ecosystem is the intentional collective of economic actors whose individual business activities, anchored around the Eclipse platform, share in some large measure the outcomes of whole collective.*

This is an adaptation of the business ecosystem definition proposed by Moore (2006). For Eclipse practitioners, shared meaning emerged over time, converging from a broad and loosely-bounded concept in stages that enfolded new ideas as they became available.

Chapter 2 previously developed two semantic frameworks in which to situate the Eclipse ecosystem construct (Table 1 and Table 4). In the categories of Table 1 (p. 22), the Eclipse ecosystem is at once a label, a metaphor, and an analogy. As a *label*, it refers to a *business ecosystem* in the technical sense of Moore (2006; section 2.4), comprised of economic agents – including (but not exclusively) for-profit business enterprises which create commercial offers and technologies that use, consume, or complement the Eclipse platform. As a *metaphor*, it frames cognition, conversation, sense-making, and decision-making by focusing attention on ecological images including cooperative as well competitive interactions, interconnectedness, shared outcomes, co-evolution, diversity, networks of relationships, and leadership in specialized niches (section 2.3). These notions are all evident in Eclipse practitioner community discourse. As an *analogy* (section 2.5), it implies vocabulary, constructs, and prescriptions from the analogous source domain of natural ecology – particularly measures of ecosystem health and a posited relationship between diversity and resilience stability. This rich and multifaceted texture of meanings resulted, in part, from enfolding over time diverse ideas from the scholarly and practitioner literatures (identified and described in more detail below). In the four dimensions of Table 4 (p. 54; conceptual location, elements, anchor point, and boundaries), the Eclipse ecosystem is an economic mode of production (conceptual location), comprised of economic actors which may be either organizations or independent individuals (elements), anchored around the Eclipse platform (anchor point), and bounded by intentionality and shared outcomes (boundaries).

The expression "Eclipse ecosystem" (often with the variant spelling of "eco-system") was in use at IBM and OTI (the group within IBM that developed Eclipse 1.0 as an internal project within IBM; section 7.2) prior to the launch of the Eclipse Consortium in 2001 (examples in Table 25, p. 196). This early usage at IBM and OTI preceded the publication of nearly all of the business ecosystems literature (reviewed in chapter 2), and the entry of "ecosystem" into the popular management lexicon (section 2.1). In the late 1990s, perceptions of open source software by business leaders, academics, and the general public were different than today. In the literature of the day[162], Stallman (1996a;

[162] Raymond (2001) and Gay (2002) are edited compilations of previously published essays, by Eric S. Raymond and Richard M. Stallman, respectively, collected together in book form. Stallman's essays were first circulated in the 1990s, and earlier versions of Raymond's titular essay had been previously published in *First Monday* (Raymond, 1998), an open access online journal, and presented at the Linux Kongress (Raymond, 1997) and other speaking engagements. Collectively, Stallman (1996a; Gay, 2002) and Raymond (2001) provided the most clearly articulated and widely-disseminated explanations of free and open source software at the time, and both views remain important and influential today.

Gay 2002) argued that free software was a moral imperative about freedom and behaving ethically, and Raymond (2001) portrayed open source software developers as volunteer hackers with a distinct gift culture and hacker ethos. IBM, among the earliest of the major computer companies to embrace open-source software, had only recently announced its commercial support for open source Apache and Linux software in 1998 and 1999 (Capek et al. 2005). Treatments of open source software development as a rational economic paradigm – such as Lerner & Tirole (2002), West (2003), von Hippel and Krogh (2003), Fink (2003), Weber (2004), Perens (2005), and Fitzgerald (2006) – were not yet published, and even years later some management scholars continued to speak of open source as non-commercial. For example, Chesbrough et al. (2006, p. 2) characterized open source as open innovation without a business model to capture value. In this context, "ecosystem" appears to have been a broadly inclusive term encompassing all aspects of company adoption and participation. Its function, in part, was to legitimate participation in Eclipse by profit-motivated companies as a rational business strategy, at a time when many business people perceived of open source software as "non-commercial" – thus Eclipse was explicitly explained as *both* open source (a community) and commercial (an ecosystem). Early Eclipse Consortium press releases variously portrayed the Eclipse ecosystem as comprised of technology ("an open source eco-system of royalty-free technology", December 16 2002), organizations ("a worldwide 'eco-system' of tools providers and consumers", April 15 2003), or individual people ("developers, software architects, technical managers, systems integrators, thought leaders, and other software development tools producers and consumers using or interested in learning about Eclipse technology", November 25 2003) (see also Table 25 in chapter 7). The semantics of "ecosystem" were fluid, partly-formed, and still developing. With its precise meaning left undefined, "ecosystem" became an inclusive and unifying concept.

After the launch of the Eclipse Foundation in 2004, practitioner discourse on the Eclipse ecosystem successively incorporated insights from the early management writing on *platforms* (section 3.3; e.g., Gawer & Cusumano, 2002; Cusumano & Gawer, 2002) to conceive of Eclipse as a participative platform, then Iansiti & Levien's (2004a, 2004b) *ecological analogy* of co-evolving networks and the role of the ecosystem keystone (sections 2.3 and 2.5), then the *open innovation* literature (Chesbrough, 2003a, 2006; Chesbrough et al. 2006) about open business models that both create and capture value (section 3.2), and Moore's (2006) formulation of the business ecosystem as a form of economic organization that coordinates continuous innovation across organizations and markets (section 2.4). Section 7.6 argued that each stage introduced a sharper vocabulary and new conceptual frameworks to more clearly explain and understand what, for the most part, Eclipse was already doing.

In summary, the "Eclipse ecosystem" concept has evolved over time, converging towards the business ecosystem concept articulated by Moore (2006), without shedding earlier metaphorical and analogous implications. Initially a legitimating concept for commercial interest in open source software, linguistic ambiguity (Astley & Zammuto, 1992; discussed in section 7.6) successively accommodated and consumed new ideas to remain a relevant central feature of the Eclipse *identity* (Whetten & Godfrey, 1999).

Question 1b: How does the Eclipse ecosystem relate to the Eclipse developer community?

The Eclipse ecosystem is tightly coupled to the Eclipse platform, the Eclipse community, and the Eclipse Foundation. By triangulating case evidence and interview responses, and drawing on salient concepts from the management literature, section 7.8 induced operational definitions for each component and operationalized their interdependencies as *resource exchange relationships* (Pfeffer & Salancik, 1978), resulting in the systems model depicted in Figure 10 (p. 206).

Section 7.4 and subsection 7.8.4 examined the *Eclipse platform*, defined here as a *platform* in the technical sense of the technology management literature (Gawer & Cusumano, 2002, 2008; West, 2003; Gawer, 2009a, 2009b):

> *The Eclipse platform is a set of building blocks on which the organizations and individuals comprising the Eclipse ecosystem can develop complementary products, technology and services.*

As defined here, scope of the Eclipse platform is broadly inclusive of Eclipse *open source software assets*, the *complementary assets* (Teece, 1986; section 3.1) that enhance the value of Eclipse software, and the *infrastructure* to maintain and distribute platform assets[163]. It is a *commons resource* available to all (Hess & Ostrom, 2007; section 5.3): software assets are distributed under open source licenses – mainly the *Eclipse Public License* (EPL) (section 7.4) – that permit anyone to obtain, use, modify, and redistribute

[163]This broadly inclusive formulation of the Eclipse platform is consistent a definition proposed by the Eclipse Foundation Executive Director (Milinkovich, 2008a, p. 54) – the "combined base of technologies, architectures, designs and assets used to build market offers, components, products and services, legal and licensing frameworks, and processes which anchor economic community" – and platform definitions in the management literature (e.g., Gawer & Cusumano, 2002, 2008; West, 2003; Gawer, 2009a). See also subsection 11.1.1 for further discussion on the selection of labels and language within this explanation.

the software, distributed through infrastructure that is accessible through the public Internet (section 7.4 and subsection 8.2.2). Eclipse practitioners distinguish between several categories of software assets within the Eclipse platform, including Eclipse *tools* for performing specific user tasks, Eclipse *frameworks* for building software applications, and Eclipse *runtimes* that perform server-side tasks.

Section 7.5 and chapter 8 examined the *Eclipse developer community*, defined here as an *open source community* in the technical sense of the scholarly literature:

> *The Eclipse developer community is the meritocracy of individual contributors and committers who maintain and extend the Eclipse platform.*

According to West & O'Mahony (2008), Eclipse is among the most transparent and accessible software communities, with community participation comparable to the earliest and most successful open source communities. By all accounts, most (but not all) Eclipse committers are paid to contribute to Eclipse – either as part of their job function with an employer, as a contractor performing work for customers, or some other arrangement (Milinkovich, 2008b; Skerrett, 2008b; Wagstrom, 2009, p. 50).

Section 7.1 and chapter 9 examined the *Eclipse Foundation:*

> *The Eclipse Foundation is the membership-based not-for-profit organization created to advance the Eclipse platform and to cultivate both an Eclipse community and an Eclipse ecosystem.*

Providing governance to both the Eclipse community and the Eclipse ecosystem, the Eclipse Foundation is both the *keystone* organization (Iansiti & Levien, 2004a, 2004b) of the Eclipse ecosystem and the *open source software foundation* (O'Mahony, 2005; West & O'Mahony, 2008; Xie, 2008) providing governance, stewardship, and a legal identity to the Eclipse community. According to Mike Milinkovich, the Executive Director of the

Eclipse Foundation: "This duality is built into our bylaws, our organization and, I would assert, our DNA" (Milinkovich, 2010, p. 29).

The components of the Eclipse field setting are connected by resource flows (Pfeffer & Salancik, 1978). The primary cycle of resource flows driving platform innovation is a *resource cycle* from the Eclipse ecosystem, to the Eclipse developer community, to the Eclipse platform, and back to the Eclipse ecosystem (Figure 11, p. 213). The economic actors of the Eclipse ecosystem provide the Eclipse developer community with development resources, the most important of which are the effort and attention of individuals who are paid to contribute to Eclipse projects (chapter 10). Individuals within the Eclipse developer community maintain and extend the Eclipse platform by writing and testing software, creating documentation and other resources, and other project activities (chapter 8). The economic actors of the business ecosystem use, extend, and commercialize the assets of the Eclipse platform to create and capture economic value (chapter 10) – and in doing so, entwine their own business outcomes with the outcomes of the ecosystem. The business ecosystem resource cycle resembles the energy and material flow cycles of biological ecosystems (chapter 2; section 7.8; Muegge, 2011), and the self-reinforcing causal feedback loops of systems thinking (Kim & Senge, 1994; Senge, 2006). As a interconnected system, its ongoing operation and sustainability require operation of each node and each connection between nodes.

A second set of resource flows connects the Eclipse Foundation to the community, platform and ecosystem (Figure 12, p. 214). The Eclipse Foundation provides governance and services to both the community and ecosystem. It provides stewardship for the platform and is a gatekeeper of Eclipse quality through the project review process; for example, Eclipse projects must pass a *release review* in order to declare a software release for public consumption, and it is the Executive Director or ED delegates who approve or fail reviews based on public comments, the scope of the project, and the purposes of the foundation (subsection 8.2.7). From the member organizations of the Eclipse ecosystem, the Eclipse Foundation obtains the financial resources for operation. From the developer community, it obtains the effort and attention of individuals who contribute to governance activities. Individuals in the community and companies in the ecosystem can become Eclipse Foundation members and participate in Foundation governance (discussed in section 7.1).

Chapter 7 presented the case evidence and inductive conceptual development for this explanation. Chapters 8, 9, and 10 presented case evidence of projects, governance, and companies, respectively, to support and further sharpen this explanation. Section 11.3 articulates the theoretical contribution of this explanation, and section 11.4 discusses alternative formulations.

Question 1c: Can the Eclipse ecosystem be described and explained using the constructs and analytic apparatus of institutional theory and the IAD framework?

According to the explanation developed here, action by organizations within the Eclipse ecosystem and individuals within Eclipse developer community is structured by *institutions* – the formal constraints, informal constraints, and enforcement characteristics that structure human and organizational interaction in ways “perfectly analogous to the rules of the game in a competitive team sport” (North, 1990, p. 4). The co-dependent institutions of the Eclipse ecosystem and Eclipse developer community respectively shape the participation decisions of companies and individuals.

Section 7.8 presented the conceptual development of this institutional description. Interpreted through the constructs and analytic apparatus of the *Institutional Analysis and Design* (IAD) framework (Ostrom, 1990, 2005; Hess & Ostrom, 2007; Poteete et al. 2010; surveyed in chapter 5), the Eclipse platform is a *commons resource* (subsection 7.8.4) and the Eclipse Foundation is a collection of polycentric governance structures (chapter 9); both are shared by the Eclipse developer community (chapter 8) and the Eclipse ecosystem (chapter 10) – as defined in the previous answer to question 1b.

Subsection 7.8.4 specified the IAD characteristics of the Eclipse platform as a commons resource shared by the community and ecosystem. Salient IAD attributes (Table 29, p. 215) are *property rights* (including the *Eclipse Public License (EPL)* for contribution and distribution, the *Eclipse.org Terms of Use* with contributor retention of copyright, the Eclipse trademark policy and other legally-enforceable rules), and the *resource attributes* of the platform components: *Eclipse software artifacts* (public goods with high initial fixed costs, near-zero marginal cost, large network effects in use, and high perceived value to users; predictable delivery; a modular plug-in architecture with open stable design rules that mitigates the large size and complexity of the software), *complementary assets* that increase the value and attractiveness of the platform and lower the cost and effort of use (with various economic properties), and *infrastructure* that makes available artifacts, information, and services to anyone with access to the Internet.

Section 8.7 specified the IAD characteristics of the institution of individual participation that structures activity within the Eclipse developer community. Four categories of rules-in-use (Table 44, p. 265) are (1) *mandatory requirements* of the *Eclipse Development Process*, (2) Eclipse Foundation *guidelines and community norms*, (3) *opt-in programs*, and (4) *local conventions* that vary between projects. The eight categories of requirements (section 8.2) are (a) a meritocratic committer nomination and election process, (b) IP management in compliance with the Eclipse *Intellectual Property Policy*, (c) openness, access, and transparency of project activity and decision-making, (d) active community building, (e) participation in the Eclipse roadmapping process, (f) a project lifecycle process with phases and managed transitions requiring community review and approval, (g) release management requiring community release reviews and approvals, and (h) committer obligations for active participation. Examples of guidelines and community norms comprising *The Eclipse Way* (subsection 8.3.1) include recommendations for consistent release numbering across projects and advice for effective community building. Projects may opt-in to the Eclipse annual release train (subsection 8.4.1) and Eclipse packages (subsection 8.4.2); projects that participate agree to follow additional operating rules. Examples of local rules that vary between projects include coding conventions, committer decision rules, choice of build systems, and preferences for utilization of project communication channels (section 8.6).

Collectively, the developer community rules-in-use partition developer activity into Eclipse projects, each of which is a *semistructure* (Brown & Eisenhardt, 1997), where some features are prescribed by requirements and community norms while others are not. Section 9.9 examined how the operating rules of developer community semistructures have changed over time – with new rules added, existing rules sharpened and improved, tacit practices made explicit and codified, and ineffective rules removed – while the big design decisions such as a commitment to openness, transparency and meritocracy remained constant. Section 10.4 reported that semistructure specification, enforcement, and change were central tensions within and between developer community and ecosystem participants, and examined how those tensions were managed – a theme examined more closely in the answer to question 1d below.

Section 9.10 specified the IAD characteristics of the Eclipse polycentric governance structures, including the membership at-large, the board of directors, the Executive Director, the Eclipse Foundation professional staff, and the architecture, planning, and requirements councils. It reported how these structures interact with the institutions of individual and company participation at both the *operating* level and at the *policy* level at which the operating rules are established and reshaped. The board and the councils bring together representatives of the community and ecosystem. *Constitutional* and policy rules are structured for *vendor neutrality*: each member organization receives the power and privileges of their membership class, with no entitlements or special treatment for any particular member. Higher classes of membership receive greater membership powers (including a larger role in governance) in exchange for higher dues and expectations for contribution, but all membership classes are available to all qualified organizations, and all membership classes have a voice on the board. In community matters, each member receives one vote regardless of the member's size or contribution.

Section 10.3 specified the IAD characteristics of the Eclipse ecosystem institution of company participation. Three categories of IAD rules-in-use (Table 57, p. 331) are (1) the property rights of platform assets, (2) the Eclipse *Membership Agreement*, and (3) opt-in programs for Eclipse Foundation members. The bundle of property rights – comprising the *Eclipse Public License (EPL)*, the *Eclipse.org Terms of Use*, the Eclipse *Intellectual Policy Policy*, *Guidelines for Eclipse Logos and Trademarks*, and other documents – treats all organizations equally to protect the Eclipse brand, encourage commercial adoption, and encourage community contribution. The Eclipse IP due diligence process is said to provide ecosystem companies with a level of confidence that Eclipse platform assets can be safely distributed in commercial products (subsection 9.4.2). The Eclipse *Membership Agreement* structures Eclipse Foundation membership into multiple tiered membership classes, each with specific powers and obligations, attractive to different participants with varying business needs and interests in participation (see Table 22, p. 170). Opt-in membership programs are voluntary bundles of rules-in-use that provide specific services and benefits to Eclipse Foundation member organizations (subsection 9.4.4).

Question 1d: Are there tensions between the Eclipse ecosystem and community? How are those tensions managed?

Section 10.4 reported three areas of ongoing tension between the Eclipse developer community and the Eclipse ecosystem: (1) the "right" amount of process, (2) resolving discrepancies between rules-in-use and rules-as-written and enforcement of obligations, and (3) effective governance of the Eclipse Foundation by a traditional board of directors while providing transparency and access to the community.

The Eclipse Foundation managed each area of tension between ecosystem and community with some combination of (1) transparency of decisions and decision-making, (2) frequent and ongoing conversation, and (3) deliberately attending to the interests of individuals in the community and companies within the ecosystem without overtly favouring one at the expense of the other. For each tension, the Eclipse Foundation also employed tension management strategies specific to that tension, for example (1) initiatives to automate and otherwise improve community processes, (2) revisions to re-align the rules-as-written with rules-in-use, and (3) enforcement of formal governance rules that limit the extent of influence of any single company, respectively.

Within the theoretical explanation developed here, the first two tensions concern specification, enforcement, and change of project semistructures – specifically, *which* structural features should be prescribed and by whom, and which structural features should be left unfettered to the discretion of local participants. Sections 8.1 and 8.2 examined the *Eclipse Development Process* and its explicit recognition of this tension in its *guiding principles* – intended to "provide as much freedom and autonomy to the Projects as possible while ensuring the collective qualities benefit the entire community" (EDP 2.5, section 2.4, p. 4) and recognizing that "part of the strength of this document is in what it does not say, and thus opens for community definition" (EDP 2.5, section 2.4, p. 4). Within the technology management literature, Brown & Eisenhardt (1997, p. 29) speculated that effective semistructures "are sufficiently rigid so that change can be organized to happen, but not so rigid that it cannot occur," and reported evidence that firms with semistructured processes were more successful innovators than firms with processes that were too extensive or too chaotic. Follow-on studies associated the fit between semistructures and context with the capability to innovate effectively in complex, uncertain, or rapidly changing environments (Brown & Eisenhardt, 1998; Okhuysen & Eisenhardt, 2002; Okhysen & Waller, 2002; Vlaar et al. 2007; Berggren et al. 2008; Jarvenpaa & Majchrzak, 2008; Eisenhardt et al. 2010). Although these arguments were developed from research conducted within business firms, they apply conceptually as well (or better) to community innovation conducted outside of traditional firms. If so, tension around semistructures, dialogues between participant stakeholders, and any resulting actions could be essential parts of the ongoing process (discussed in section 9.9) to realign rules-as written and rules-in-use with the needs of the day.

Tension between stakeholders is in no way unique to business ecosystems, being evident also in strategic alliances (Culpan, 2009; Makadok & Coff, 2009) and in other systems with partially-aligned incentives, where participants have some shared interests and some divergent interests (Brandenburger & Nalebuff, 1996). The relationship between ecosystem and community differs from the stakeholder relationships of other systems in at least one important way: influential individuals are typically embedded in *both* systems, as contributors and committers in the community, and as employees and owners of companies in the business ecosystem. In this sense, effective management and harnessing of tension within the Eclipse field setting goes beyond good alliance management. Although some tension management strategies noted here are found also in the strategic alliances literature, others are unique, and the process of harnessing tension to improve the system for both stakeholder groups is a promising area for further study.

11.1.1 A note about labels and language

Notes throughout this document, particularly in chapters 2, 3, and 5, speak to potential semantic confusion about the use of language in this research[164]. In particular, the labels for the three central constructs of the Eclipse resource cycle – the platform, community, and ecosystem – each have alternative meanings in scholarly writing and in Eclipse practitioner discourse. This subsection is a brief synthetic treatment of the rationale for selecting each label and the implications of these choices.

Of the three constructs, the most contentious label is "Eclipse platform," which has multiple alternative meanings within Eclipse discourse, often much narrower than the definition used here. Examples include the "Eclipse Platform Project" (the original Eclipse top-level project that produces the software components of the Eclipse software development kit, or SDK), the "Eclipse Rich Client Platform" (RCP; e.g., McAffer et al. 2010a), the "Eclipse Tooling Platform", and the "Eclipse Runtime Platform" (RT; e.g., McAffer et al. 2010b). Some Eclipse technical and architectural documents situate the "Eclipse Platform" as software functionality that sits below and is common to the Eclipse software tools, frameworks, and runtimes (e.g., *Executive Director Report*, 2006Q1). This multiplicity of meanings was a recurring source of initial confusion when reviewing preliminary findings with practitioners within the Eclipse developer community. One respondent recommended "Eclipse commons" as an alternative label that avoids the semantic encumbrance of an already contested term within the Eclipse field setting. Carefully weighing all of these arguments, "platform" was selected here for use within

[164] See, for example, sections 2.4, 2.5, 2.6 and 2.7, sections 3.2, 3.3, 3.5 and 3.6, and footnotes 21, 22, 23, 26, 28, 43, 60, 74, 80 and 86.

this document for consistency with extant management research on platforms (section 3.3) and the broader scholarly community. However, a different label such as "Eclipse commons" may be more effective for future dialogue with Eclipse practitioners.

Sections 3.6 and 7.5 discussed alternative meanings of "community" in theory and practice. The label and construct of interest here is the "Eclipse developer community" – a specific technical term in Eclipse governance documents and practitioner discourse (examined in section 7.5.1) that is congruent with the "development core" of the open source software community construct in the scholarly literature (Crowston & Howison, 2003; Herraiz et al. 2006; Antikainen et al. 2007; West & O'Mahony, 2008; Kilamo et al. 2010). It is comprised of individuals – not organizations – for reasons previously explained in sections 3.6 and 7.5. It is important to recognize that Eclipse discourse also includes several other communities: the "Eclipse developer community" is a specific subset of the broadly inclusive "Eclipse community" and distinct from the "Eclipse user community" and "Eclipse adopter community" (section 7.5). Consistent use of the complete three-word label throughout this document is intended to reduce the potential for confusion between these various notions of Eclipse communities[165].

Chapter 3 and section 7.6 discussed alternative meanings of "ecosystem" in theory and practice. The label and construct of interest here is the "Eclipse ecosystem" as understood by Eclipse practitioners. Discovering, articulating, and theoretically specifying this construct was one of the operational questions of this research (case study

[165] See also subsection 7.8.2, especially footnote 92, for an explanation of how these other "Eclipse communities" are implicit in the resource dependence explanation of the Eclipse field setting, and opportunities for future research to further examine the roles of users and adopters.

question 1a). Previous subsections answered this question by defining the Eclipse ecosystem as a business ecosystem in the technical sense of Moore (2006), consistent with field observations, case evidence, and explanations provided by the Eclipse Executive Director and Foundation staff (e.g., Milinkovich 2008a), and situated this construct within the conceptual frameworks developed in chapter 2.

Section 7.6 also observed the existence of a second alternative meaning in practitioner discourse: the "Eclipse ecosystem" as a *technology ecosystem* of interoperable and co-dependent software components based on Eclipse software technology. As a metaphor, the technology ecosystem brings into sharp focus the interconnectedness, co-dependence, and co-evolution of Eclipse software components in the same way that the business ecosystem metaphor discussed in section 2.3 focused attention on the ecological properties of business organizations. The scholarly literature on modularity and design, surveyed in section 3.4, offers a theoretical explanation to rationalize and join these two co-existing notions – the business ecosystem metaphor for the *producers of innovation* and the technology ecosystem metaphor for the *products of innovation.* Baldwin & Clark (2000) argued that modularity links the design structure of a product architecture to the task structure of a design organization and to the corresponding industry structure of firms and markets. A highly modular product architecture enables a *modular cluster* industry structure, and both co-evolve together in a multilevel complex adaptive system (Holland, 1992; Miller & Page, 2007). By this argument, the business ecosystem of economic actors and technology ecosystem of software components can be understood as different levels of the same system: through modularity, ecological properties at one level *imply* ecological properties at the other. Observations that some practitioner discourse about the "Eclipse ecosystem" applies equally well to either the business ecosystem of economic actors or the technology ecosystem software components supports an explanation of multilevel isomorphism.

The Eclipse ecosystem, as operationalized as a business ecosystem in subsection 7.8.1 and the answer to question 1a, is more than a metaphor to shape practitioner thought (section 2.3) – it is also a label for a theoretical construct with a corresponding literature (surveyed in section 2.4), and an ecological analogy contributing vocabulary, constructs, and prescriptions from the source domain of natural ecology (sections 2.2 and 2.5). Conceptually, the interconnected system of software components can be situated within the business ecosystem construct as analogous to the abiotic habitat in which the biotic economic agents operate. In this way, the Eclipse ecosystem construct developed here can also accommodate the implications of the technology ecosystem metaphor.

This completes the answer to the first of two guiding research questions that motivated this thesis. These answers were neither known nor assumed *a priori*, but were induced from detailed field work – observations of community discourse, case evidence, and informant interviews – following the research design and method of chapter 6. The next section continues this discussion by answering the second guiding research question.

11.2 Why do companies participate in the Eclipse ecosystem?

The second guiding research question is answered in two parts.

Question 2a: What are the motives, incentives and benefits for company participation?

Analysis of case evidence in section 10.2 discovered broad diversity in company participation in the Eclipse field setting. The eight case companies participated in the Eclipse ecosystem in a variety of different ways (subsection 10.2.1): as users of Eclipse technology, as adopters of Eclipse technology within product and service offerings, as members of the Eclipse Foundation, as employers of committers and contributors to Eclipse projects, as employers of council members and representatives on the board of directors, as event sponsors, as participants in member programs, as donors of IT infrastructure, as leaders of strategic initiatives on behalf of the board of directors, and in various other roles.

Companies participating in platform development assigned (or at least permitted) paid employees to contribute to Eclipse projects as part of their job function. In addition to assigning staff to general project work, companies selectively contributed code to specific Eclipse projects in one or more of four different ways (subsection 10.2.2): (1) by assigning employees to work on particular features or capabilities within the Eclipse developer community, (2) by developing code outside of the Eclipse community with the intent to contribute it later when certain conditions were met, (3) by contributing code that had been developed for the company's own internal use and consumption, and (4) by contributing code developed for a customer consulting contract.

Companies employed one or more of five different entry strategies to place their own employees in committer and leadership roles on an Eclipse project (subsection 10.2.3): (1) launch and lead a new project, (2) participate in a supporting role at the launch of a new project, (3) assign staff to contribute and work up through the project meritocracy, (4) hire the project's existing committers, and (5) contribute code from a different Eclipse project through a move or restructuring review (creating strong incentives for committers on the destination project to provide committer status to the developers within the originating project who had previously maintained that code).

Companies benefited from participation in one or more of three different ways (subsection 10.2.4): (1) all case companies obtained access to capabilities required to strengthen their business model, (2) companies that performed a portion of their R&D within an Eclipse project benefited from the cost savings, efficiencies and performance advantages of open source development – community contributions and bug reports on company-driven projects, and pooled R&D (West & Gallagher, 2006) and cumulative innovations (Allen, 1983; Scotchmer, 2006; Fiona & O'Mahony, 2007) from the activities of other developers on community-driven projects, and (3) some companies adopted new Eclipse-based business models that were not otherwise viable.

Based on these observations of the case evidence, section 10.2 argued that *profit-motivated companies participate in the Eclipse ecosystem to strengthen or transform their business models*. In an adaptation of the Johnson et al. (2008; Johnson, 2010) business model framework reviewed in section 3.2, a business model consists of a *customer value proposition*, a *profit formula* of how the company generates attractive profits (including a way to *generate revenue* and a *cost structure* that effectively places a lower bound on offer pricing), and set of *capabilities* – resources and processes – required to deliver on the value proposition and profit formula. The following paragraphs explain how each of the observed benefits of participation in the Eclipse ecosystem can be situated within a business model framework.

All of the eight case companies gained access to specialized and cospecialized complementary assets necessary to appropriate value from innovation (Teece, 1986, 2009). Examples included specialized knowledge and expertise which is difficult to codify and not easily obtained in other ways (Teece, 1981, 1982, 2000), unique branding and competitive positioning within the company's target market, access to a network of potential complementors and partners, and visibility to potential customers through various Eclipse members programs. Rather than purchase all the required capabilities of their business model through market transactions or develop them in-house, these companies obtained capabilities through participation in the Eclipse ecosystem – a form of what Chesbrough & Appleyard (2007) call *open strategy*.

Some case companies performed a portion of their R&D within the Eclipse developer community, by employing committers to develop required features and capabilities and consuming the assets released by those projects as inbound open innovation. These companies benefited from community contributions to the projects that they developed and consumed. The quantity of community code contributions varied widely between the case projects examined here, from approximately two dozen large code contributions on the BIRT project to more than a thousand on the CDT project

(Table 39, p. 258)[166]. Companies participating in community-driven projects obtained further leverage from the community through pooled R&D (West & Gallagher, 2006) of sharing costs and effort with others, and cumulative innovation (Allen, 1983; Scotchmer, 2006; Fiona & O'Mahony, 2007) of others extending and improving on the company's contribution. Within the business model framework, these benefits variously resulted in lower cost structures, greater revenue opportunities, and stronger value propositions.

Finally, some case companies invented or re-invented a new business model, anchored around the Eclipse platform and the Eclipse ecosystem, that would not otherwise have been viable. Example include Actuate's launch and leadership of the Eclipse BIRT project enabling a new value proposition of transitioning enterprises from in-house open source solutions to managed integrated solutions, EclipseSource's leadership of Eclipse runtime and platform projects enabling a value proposition and brand uniquely anchored in Eclipse services, and Tasktop's leadership of the Mylyn project providing it with unique differentiation and advantage at providing Mylyn-based application lifecycle management software products and services.

This explanation, induced from case evidence, that companies participate in the Eclipse ecosystem to strengthen or transform their business models, is consistent with practitioner explanations identified in section 7.7 – that companies participate in the Eclipse ecosystem in pragmatic pursuit of profits and competitive advantage over rivals, and contribute resources to the Eclipse development community in order to obtain

[166]"Large" code contributions are greater than 250 lines of code, triggering special scrutiny and tracking through the Eclipse IP review process. The number of smaller code contributions, which are not individually tracked in the project IP logs, is potentially much higher than the numbers reported here.

tangible business benefits. More specifically, the case evidence supports the practitioner propositions that companies (1) build and protect differentiating capability on top of Eclipse open source assets and collaborate on non-differentiating platform development with other companies (sometimes including direct competitors), (2) re-invest some earnings in Eclipse platform development by contributing resources to the developer community, (3) engage with the Eclipse community and ecosystem at different levels, for different reasons, and with different corresponding benefits, and (4) variously benefit from participation in Eclipse by increasing revenue, reducing costs, and reshaping perceptions. This explanation sharpens, integrates and extends practitioner propositions by bringing them together into an integrated business model explanation for participation, and elucidating deeper levels of substructure on participation, motivation, and benefits.

Question 2b: What is it about the Eclipse ecosystem that enables, promotes, and sustains participation?

This subsection employs an explanation-building analytic strategy (Yin, 2003a, pp. 120-122) to bring together the explanations of the Eclipse field setting (questions 1a, 1b, 1c, and 1d) and the induced proposition that companies participate in the Eclipse ecosystem to strengthen or transform their business models (question 2a) to develop a theoretical explanation linking company participation to the properties of Eclipse.

Table 58 is a compilation of various findings reported in chapters 7, 8, 9 and 10 on the nodes and ties that comprise the Eclipse production resource cycle of Figure 10 (p. 206) and Figure 11 (p. 213), and the role of the Eclipse Foundation and the salient institutional features impacting each component.

Table 58: Institutions, governance action, and the production resource cycle

Node or tie	Component	Desired outcome	Role of the Eclipse Foundation (Board of directors, ED, EMO)	Institutional features
Node (*E*)	Eclipse business ecosystem	Ecosystem health. Evidence of high health metrics: • *Productivity* of innovative output. • *Robustness* to external shocks. • *Niche creation* of new initiatives and industry verticals.	"Cultivate ... an ecosystem of complementary products, capabilities, and services." • Grow the Eclipse Foundation membership. • Promote member companies and their commercial offers. • Create opportunities for collaboration.	• The not-for-profit ecosystem keystone does not compete with ecosystem companies. • Vendor neutral governance means that all companies participate under the same rules. • Tiered Eclipse Foundation membership. • All membership classes have representation in Eclipse governance.
Resource flow (R_{EC})	Contributors	Companies within the Eclipse ecosystem contribute resources to the Eclipse developer community.	• Encourage members to fulfil obligations. • Create opportunities for collaboration between marketplace competitors.	• Multiple modes of project entry and company contribution to the platform. • Contributors retain copyright.
Node (*C*)	Eclipse developer community	Sustained production of high quality software tools, frameworks, and runtimes. Evidence of a thriving, diverse, and active *developer* community.	"Cultivate ... an open source community." • Encourage projects to come to Eclipse. • Organize community events. • Mentor projects. • Provide IT infrastructure for projects.	• Open source rules of engagement are appealing to developers, and associated with production of high quality software.
Resource flow (R_{CP})	Contributions	Contributors and committers within the Eclipse developer community maintain and extend the Eclipse platform.	• Provide IT infrastructure for projects that welcomes new contributions. • Reduce the effort required to contribute.	• Community-building is required by the EDP and reinforced by community norms. • *Terms of use* intended to simplify contribution.
Node (*P*)	Eclipse platform	Assets within the Eclipse platform are valued by users and adopters. Evidence of an active and engaged *user* community, and an active and engaged *adopter* community.	"Advance the creation, evolution, promotion, and support of the Eclipse Platform" • Promote the use and adoption of Eclipse platform software. • Coordinate the annual release train. • Perform and enforce IP due diligence.	• Modular architecture with public design rules. • The annual release train provides coordinated software releases on a predictable timeline. • The Eclipse IP policy and IP due diligence process provide enterprises with assurance that code is unencumbered by IP issues.
Resource flow (R_{PE})	Platform assets (inbound open innovation)	Companies within the Eclipse ecosystem use and adopt the Eclipse platform to develop complementary products, technologies and services.	• Promote the use and adoption of Eclipse platform software in commercial products. • Provide infrastructure and complementary assets for platform use and adoption.	Software is ready for commercial adoption: • The EPL and legal framework is friendly to commercial adoption. • The EDP delivers well defined and stable APIs for extensions by commercial products.

The following discussion of Table 58 is ordered hierarchically; it begins at the technology resource (the Eclipse platform), then considers the organization of individuals (the Eclipse developer community), the organization of economic activity (the Eclipse ecosystem), and the role of governance (the Eclipse Foundation). Each level of discussion examines the node and the outbound resource flow of the production resource cycle (Figure 11, p. 213). Excerpts of documents, community discourse, and informant interviews are provided as illustrative examples of key points. Discussion of each field setting component enfolds the salient literature, beginning with the practitioner literature on architectures of participation (reviewed in section 4.1), and the nascent scholarly literature that seeks to formalize, generalize and extend practitioner heuristics into robust management theory (reviewed in section 4.2). Of particular interest is identifying the aspects of the Eclipse field setting which can partially explain company participation, but are not accounted for by extant theory.

The central argument developed from this discussion is that participation in the Eclipse field setting is driven by a multilevel socio-technical *architecture of participation* (O'Reilly, 2005a) that includes high *modularity* and *option value* of the Eclipse platform (Baldwin & Clark, 2006), *transparency and accessibility* of the Eclipse developer community (West & O'Mahony, 2008), and institutional features of the field setting that enable and motivate resource flows from economic actors in the Eclipse business ecosystem to the developer community. Also important are *keystone activities* performed by the Eclipse Foundation to promote the platform and the interests of Eclipse Foundation members. Previous scholarly work to operationalize architectures of participation has addressed modularity and option value of the platform (Baldwin & Clark, 2006) and transparency and accessibility of the developer community (West & O'Mahony, 2008); the institutional features of the Eclipse business ecosystem and the keystone activities of the Eclipse Foundation are new contributions not found in the extant scholarly literature on architectures of participation.

At the level of the technology resource, the Eclipse platform is structured by the Eclipse software architecture: a modular plug-in structure with a small runtime kernel and well-specified public design rules (examined in section 7.4). The architecture plan of the *2006 Eclipse Roadmap* describes the Eclipse software architecture as follows:

> *The overall architecture of the Eclipse projects is driven by the elegant plug-in mechanism of the underlying Eclipse Platform. Each project contributes a variety of plug-ins that extend the basic functions of the platform and tailor it into a useful environment for that project's users: both API users and GUI users. A key feature of the Eclipse architecture is that each plug-in not only extends its prerequisites but also contributes additional extension points for other plug-ins. In this way, each Eclipse project provides an extensible framework for the Eclipse ecosystem to build upon, to use, and to commercialize.*

Overlapping technical solutions compete with one another – within projects, between projects, and between projects and complementary solutions developed outside the platform. For example, there are multiple Eclipse plug-ins providing integration with the Subversion source code repository (e.g., "Subversive" and "Subclipse"), and competing plug-ins providing integration with the Maven build management system (e.g., the "m2e" project led by Sonatype and the "IAM" project – "integration for Apache Maven"). Ian Skerrett, Eclipse Director of Marketing, writes (Skerrett blog, June 2 2011):

> *[O]pen source projects compete with each other. Eclipse and Apache have a number of competing projects, ex. Equinox/Felix, Jetty/Tomcat plus others. The*

> *Linux distros compete with each other. Competition is good. Open source is not all one happy family...*
>
> *[T]rying to force individuals or organization to work together is not how you run a successful open source project. Successful open source is about collaboration of individual and organizations. You can't force this collaboration.*

These observations are consistent with the practitioner heuristics of O'Reilly (2005a) and the theoretical arguments of Baldwin & Clark (2000, 2006) regarding architecture at the platform level. All of the O'Reilly platform-level practitioner heuristics identified in section 4.1 are evident in the Eclipse case data: small modular applications, well-defined application interfaces, transparency of design, a small tightly controlled "cathedral" core with a well-defined extension mechanism linking to a surrounding "bazaar", competing solutions in a free market for ideas, with user value (as assessed by users) as a criterion for selection. These observations are consistent with the Baldwin & Clark (2006) theoretically-developed argument that software code base architectures that are more modular and have more option value increase the incentives for developers to join and remain involved in an open source development effort. In such systems, rival technical solutions compete within a multilevel complex adaptive system with strong ecological parallels (Baldwin & Clark, 2000; Baldwin & Woodard, 2009; see also Table 6, p. 79).

Additionally, there are platform-related actions by the Eclipse Foundation and other institutional features of the Eclipse platform that make participation by companies more attractive. "Advance the creation, evolution, promotion, and support of the Eclipse Platform" (*Bylaws of the Eclipse Foundation*, 2003, 2008, article 1.1) is the first of three legal purposes of the Eclipse Foundation (sections 7.1 and 7.4). The Eclipse Foundation, through the various structures comprising the Eclipse Management Organization, actively markets the platform to users and adopters (subsection 9.4.4), coordinates the annual release train (subsection 8.4.1), and performs and enforces IP due diligence (section 8.2.2 and subsection 9.4.2), and the rules of the *Eclipse Development Process* require project committers to actively nurture communities around their projects (subsection 8.2.4). It actively and deliberately promotes the use and adoption of Eclipse platform software in commercial products (resource flow R_{PE} in Table 58). According to the logic of the *Eclipse Development Process*, active and engaged communities of users and adopters are evidence that a project's software is useful and extensible, respectively (subsection 8.2.4). Ian Skerrett, the Eclipse Foundation Director of Marketing, explains:

> *Eclipse technology is valuable to ecosystem companies not only because it's great software – which it is – but because it has huge numbers of users and adopters. For many companies that become part of the Eclipse ecosystem, one reason why it makes business sense to build their systems on Eclipse software is that lots of people already use Eclipse software.*

Eclipse practitioners associate high levels of use and adoptions with both technological and institutional factors (*2006 Eclipse Roadmap*):

> *The high rate of adoption of the Eclipse technology can be traced to two key factors: great technology, and the ease with which it can be adopted by others, both commercial and open source. This ease of adoption has, in turn, several dimensions. The EPL and its CPL predecessor provide terms which are conducive to both commercial and open source use. The focus on extensible frameworks has made it relatively simple to re-use Eclipse Technology.*

The Eclipse annual release train provides coordinated software releases on a predictable timeline for consumption by enterprises (subsection 8.4.1), and the Eclipse IP policy provides enterprises with assurance that code is unencumbered by IP issues and can be safely used in commercial products (subsection 8.2.2). The semistructures of the *Eclipse Development Process* include rules and processes to deliver well-defined and stable APIs for extension by commercial products, and the EPL and legal frameworks of the platform are friendly to commercial adoption by companies. Without contradicting the findings of the extant literature, these field observations provide richer and more concrete insight on the Eclipse architecture of participation.

At the level of individual organization within the developer community, extant literature argues that sponsors of open source communities consider the transparency and accessibility of three community design dimensions when developing a sociotechnical community architecture – the processes of production (the way in which the community produces software), governance (the way in which decisions are made), and intellectual property (the allocation rights to use the community's output) – and that higher levels of transparency and accessibility in these three dimensions are associated with attracting external participants and growing communities (West & O'Mahony, 2008). Field observations and theorizing about the Eclipse field setting support and extend those arguments. "Cultivate ... an open source community" (*Bylaws of the Eclipse Foundation*, 2003, 2008, article 1.1) is the second of three legal purposes of the Eclipse Foundation (sections 7.1 and 7.5). The Eclipse open source rules of engagement (subsection 8.1.2) – a central feature of the *Eclipse Development Process* (EDP 2.5, section 2, p. 4) – identify openness, transparency, and meritocracy as principles that motivate the rules and processes of the Eclipse developer community[167]. Case evidence presented in chapter 8 supports the notion that transparency and accessibility are deeply embedded in the

[167] According to West & O'Mahony (2008), "openness" is a larger concept that includes both transparency and accessibility. Eclipse Foundation documents use the term "openness" more narrowly for the concept that West & O'Mahony label "accessibility" – thus Eclipse is said to be open and transparent in Eclipse Foundation documents, and transparent and accessible in the parlance of the scholarly literature.

community production, governance, and IP structures of the Eclipse field setting. Many of O'Reilly's community-level practitioner heuristics, reviewed in section 4.1, are evident on the community-driven case projects: low barriers to entry for new users, contributions welcomed from outside the community, rival solutions compete on a level playing field, user value (as assessed by users) is the criterion for selecting one solution rather than a different one, users have the credible capability to fork the project (providing strong incentives for developers to be responsive to users), and ordinary users can extend the platform with plug-ins – something which users could not do for themselves on other systems at the time when the Eclipse platform was first introduced in 2001.

Additionally, there are community-related actions by the Eclipse Foundation and institutional features of the Eclipse developer community that go beyond transparency and accessibility that make participation more attractive to companies. The Eclipse Foundation Executive Director comments on the Eclipse Foundation's active role in bringing new projects to Eclipse (Milinkovich blog, January 28 2009):

> *Eclipse's technology has relatively little to do with the Eclipse Foundation per se, since we do not direct the projects or their technical decisions. However, I think that we at the Foundation can claim that we have helped to recruit new projects and new technology innovations to come to Eclipse... And that has resulted in a much larger and more diverse collection of cool projects than I think any of us imagined five years ago.*

Other keystone activities of the Eclipse Foundation include organizing community events, mentoring projects, and providing IT infrastructure for projects. The role adopted by the Eclipse Foundation is one of *creating opportunities* rather than directing community activity. Don Smith, the Eclipse Foundation Director of Ecosystem Development, explains:

> *One of the common misunderstandings is that the Eclipse Foundation can tell the community what to do. That doesn't work. If the Foundation tried to do that, the community could ignore it, or community members could choose to leave and go elsewhere.*
>
> *Sometimes we play the role of referee. Sometimes we play the role of explaining things.*

Similarly, a former Eclipse Committer Community Director writes (Freeman-Benson blog, January 2008):

> *If there's one thing I've learned at Eclipse so far is that I can't tell you what to do. I can't sell you all on something. When I started, I thought I could, but I was wrong. Instead, what I can (and have), and will continue to do is to create attractive opportunities for the right things to happen.*

The community institutional features are *semistructures* (Brown & Eisenhardt, 1997), where some features are prescribed by requirements and guided by community norms and other features are unconstrained, and the partitioning and right level of structure within community semistructures is an ongoing tension resolved through transparent, accessible and ongoing community discussion, and periodic revisions to re-align rules-as-written with rules-in-use (section 10.4). An example of an area of autonomy is project organization (Beaton blog, July 14 2010; see also section 8.1):

> *It's left to the project teams to decide how they want to organize. Typically, mid-level projects tend to be used to provide some hierarchical organization for related projects. Very often mid-level projects (and top-level projects in some cases) provide handy aggregate builds and downloads of the software produced by the projects they contain.*

A second example is the timing and aggregation of releases (Bug #301065, comment 6, February 12 2010; see also subsection 8.2.7):

> *Web Tools, BIRT, and DTP [projects] all release as a unit. Each separate project under modeling.mdt does a separate release.... Ultimately, each project is doing what they think is right. And I think that's the right approach: let the project (with help/direction from the PMC) decide what is best for their project and community.*

Active community-building is required by the *Eclipse Development Process* (subsection 8.2.4) and reinforced by community norms (section 8.3). Terms of use are intended to simplify contribution (subsection 8.2.2), and project infrastructure is intended to welcome new contributions with low effort (subsection 9.4.1).

At the level of economic activity within the Eclipse ecosystem, extant literature on architectures of participation, both practitioner and scholarly, is largely silent. Three salient institutional features of the Eclipse field setting at the ecosystem level are (1) the not-for-profit status of the governing keystone organization, (2) institutionalized rules for vendor-neutral governance, and (3) tiered membership with representation of all tiers in the governing keystone. As a membership-based not-for-profit corporation, the Eclipse Foundation does not compete with companies in the Eclipse ecosystem. According to the Eclipse Executive Director, this difference between the Eclipse Foundation and for-profit keystone companies that steward open source technology platforms is important and noteworthy (Milinkovich blog, January 28 2007):

> *It is incorrect to compare the Eclipse Foundation to an open source company. Eclipse is a not-for-profit. That is a huge difference. No matter how much an open source company behaves well, and acts in the interests of its community, its role is to make money for its shareholders. Or in the terms used in Open Business Models, the business model of open source companies is based on capturing value created by its community.*
>
> *At the Eclipse Foundation, we work not to make money for the Foundation itself, but to look for ways to make money for others: namely, our members and the broader Eclipse ecosystem. Or in other words, our role is to enable others to capture value from the technology and innovation developed within the Eclipse projects.*

Vendor neutral governance (chapter 9; see also footnote 1) means that all participating companies of a particular membership class receive the same the membership powers and member obligations with no special status for any organization. According to participants, this enables collaboration within Eclipse by companies that may be fierce competitors in the marketplace. Adoption by rival companies in competing products – and collaboration by the employees of those companies on project maintenance and extension – is a characteristic of many of the most widely adopted Eclipse projects (Skerrett blog, October 11 2005):

> *In open source, if you can get your competitors to adopt your project, you have just grown the entire community. It is not just you talking about the project, it is your competitors too. The Eclipse Web Tools project is a great example: IBM, BEA, JBoss, ObjectWeb, Oracle are all building on top of Web Tools. Now, that is a lot of people talking about Web Tools.*

Membership in the Eclipse Foundation is structured into tiered membership classes (Table 22, p. 170), each with a distinct value proposition, fee structure, and bundle of membership powers and obligations, matched to a specific segment of the ecosystem[168]. *Strategic members* – both strategic developers and strategic consumers – have board-level influence over the foundation, and are said to send a clear signal of their commitment to Eclipse up and down their supply chain. *Enterprise members* benefit through access to industry working groups, close alignment with the Eclipse community, and clear signals to customers and suppliers about the need for plugable software; they consume Eclipse technology, use open source collaboration to drive efficiencies, and gain access to Eclipse Foundation services and member benefits without the obligations of the strategic member or solutions member tiers. *Solutions members* benefit through marketing, networking, ecosystem influence, information feeds, event discounts, and

[168] The membership information in this paragraph is compiled from an Eclipse Foundation document titled *Proposed membership changes* (June 24 2008, pp. 4, 5, 10).

other member benefits and services. *Associate members* signal their support for Eclipse and benefit through opportunities for information and networking. Power to shape the rules of the community and ecosystem at the policy and constitutional levels is shared among members – not equally, because the *Bylaws* and *Membership Agreement* provide greater formal influence to companies at higher membership tiers than to companies at lower tiers – but with every membership class having a voice in governance, and no single organization dominating decision-making (chapter 9).

Additionally, there are ecosystem-related actions by the Eclipse Foundation that make participation by companies more attractive. "Cultivate ... an ecosystem of complementary products, capabilities, and services" (*Bylaws of the Eclipse Foundation*, 2003, 2008, article 1.1) is the third of three legal purposes of the Eclipse Foundation (sections 7.1 and 7.6). In this role of the ecosystem keystone (Iansiti & Levien, 2004a, 2004b), the Eclipse Management Organization and Eclipse Foundation staff work to grow the Eclipse membership, promote member companies and their commercial offers, and create opportunities for collaboration (section 9.4).

Finally, there are institutional features of Eclipse Foundation governance that span multiple levels of organization and impact both the developer community and ecosystem. The first is the multiple simultaneous purposes of the Eclipse Foundation (section 7.1) – to advance the Eclipse platform, to cultivate the Eclipse open source community, and to cultivate a commercial ecosystem – which are endemic to Eclipse institutions and the actions undertaken by the Eclipse Management Organization: parallel purposes are "built into our bylaws, our organization, and I wound assert, our DNA"

(Milinkovich, 2010, p. 29). The second – closely related to the first – is an approach to managing and constructively harnessing tensions within the Eclipse field setting that attends to both the individual committers and contributors of the Eclipse developer community and the companies of the Eclipse ecosystem without one dominating the other (section 10.4).

In summary, this explanation argues that participation in the Eclipse field setting is driven by a multilevel socio-technical *architecture of participation* (O'Reilly, 2005a) that includes high *modularity* and *option value* (Baldwin & Clark, 2006) of the Eclipse platform, *transparency and accessibility* (West & O'Mahony, 2008) of the Eclipse developer community, and institutional features of the field setting that enable and motivate resource flows from economic actors in a business ecosystem to the developer community. Also important are *keystone activities* performed by the Eclipse Foundation to promote the platform and the interests of Eclipse Foundation members. The institutional features of the Eclipse business ecosystem and the keystone activities of the Eclipse Foundation are not well addressed in the extant scholarly literature on architectures of participation, which has thus far emphasized features of the technology platform and of the developer and user communities. This explanation brings together three insights – (1) the institutional explanation of the Eclipse field setting induced in section 11.1 and Table 58 (p. 368), (2) the observation that most committers in the Eclipse developer community are employed by companies in the Eclipse ecosystem (section 8.7), and (3) the induced explanation that companies participating in Eclipse platform development do so to strengthen or transform their business models (section

11.2, question 2a) – to suggest possible candidates for architecture of participation heuristics in addition to those identified from the practitioner literature in section 4.1:

- Vendor-neutral keystone governance. Power is shared – not equally, but according to clearly-defined governance rules that give every member a voice, provide all stakeholders with opportunities to get involved, and ensure that no single organization dominates governance decisions.
- The keystone foundation is a membership-based not-for-profit corporation that does not compete with ecosystem companies to appropriate platform value.
- Dual identity as both a collaborative community and a competitive commercial ecosystem, reflected in governance rules, guidelines, and norms of behaviour. The keystone foundation deliberately cultivates *both* a developer community creating value and a business ecosystem capturing value.
- Graduated benefits and obligations of participation. Participants receive benefits that increase with increasing contribution in multiple tiers – for example, through tiered membership classes, each with a value proposition and fee structure matched to a well-defined segment of potential participants.
- Harness tensions between the community and ecosystem to improve the system.
- Minimally-specified community and ecosystem *semistructures* (Brown & Eisenhardt, 1997); the structured portion of each semistructure exists largely to protect the interests of participants in the other component.
- Periodically revise the rules-as-written to re-align with the rules-in-use.
- Launch with *deliberate strategy*; grow with *emergent strategy* (Mintzberg, 1987).
- Institutional rules that favour company participation in extending and maintaining the platform: contributors retain copyright; weak copy-left license, etc.
- The keystone foundation performs actions that add value to participants but which would otherwise be vulnerable to collective action problems, commons dilemmas, and free riding if performed voluntarily by participant contributions.

Each candidate within this bundle of heuristics is offered as a tentative proposition for future research. Theoretical contributions of these results are examined in section 11.3.

This section completes the answer to the second of two guiding research questions that motivated this research. The next section continues this discussion by examining the theoretical contribution of these results and situating that contribution within the extant scholarly and practitioner literature.

11.3 Theoretical contribution

The theoretical contributions of this research include (1) a clear distinction between the developer community and business ecosystem, both as theoretical constructs and as structures of practical consequence to practitioners, and a systems-level explanation for the interdependence and co-evolution of the community and ecosystem, (2) extensions to the nascent theory of participation advanced by Baldwin & Clark (2006) and West & O'Mahony (2008), (3) insights enriching both the literature on commons governance and the management literature on business ecosystems and community-developed platforms, and (4) insights for institutional theories of management and how aspects of an institutional perspective can enrich and complement other management perspectives. The first and second theoretical contributions are discussed together in this section. The third and forth theoretical contributions are discussed in section 11.5.

Figure 19 reinterprets the explanations of the Eclipse field setting from sections 11.1 and 11.2 at a higher level of generality. For clarity and precision, Figure 19 employs an algebraic notation. The four nodes of the system are *P*, a community-developed platform, *E*, a business ecosystem of economic actors commercializing the platform assets, *C*, a developer community that maintains and extends the platform assets, and *G*, a governance foundation. Resource flows (R_{XY}), indicated graphically by arrows and labeled by source (*X*) and destination (*Y*), connect the nodes. This systems-level explanation, linking the business ecosystem and developer community through resource flows, interdependent institutions, shared governance, and a shared knowledge commons platform, is the first of four theoretical contributions to the scholarly literature.

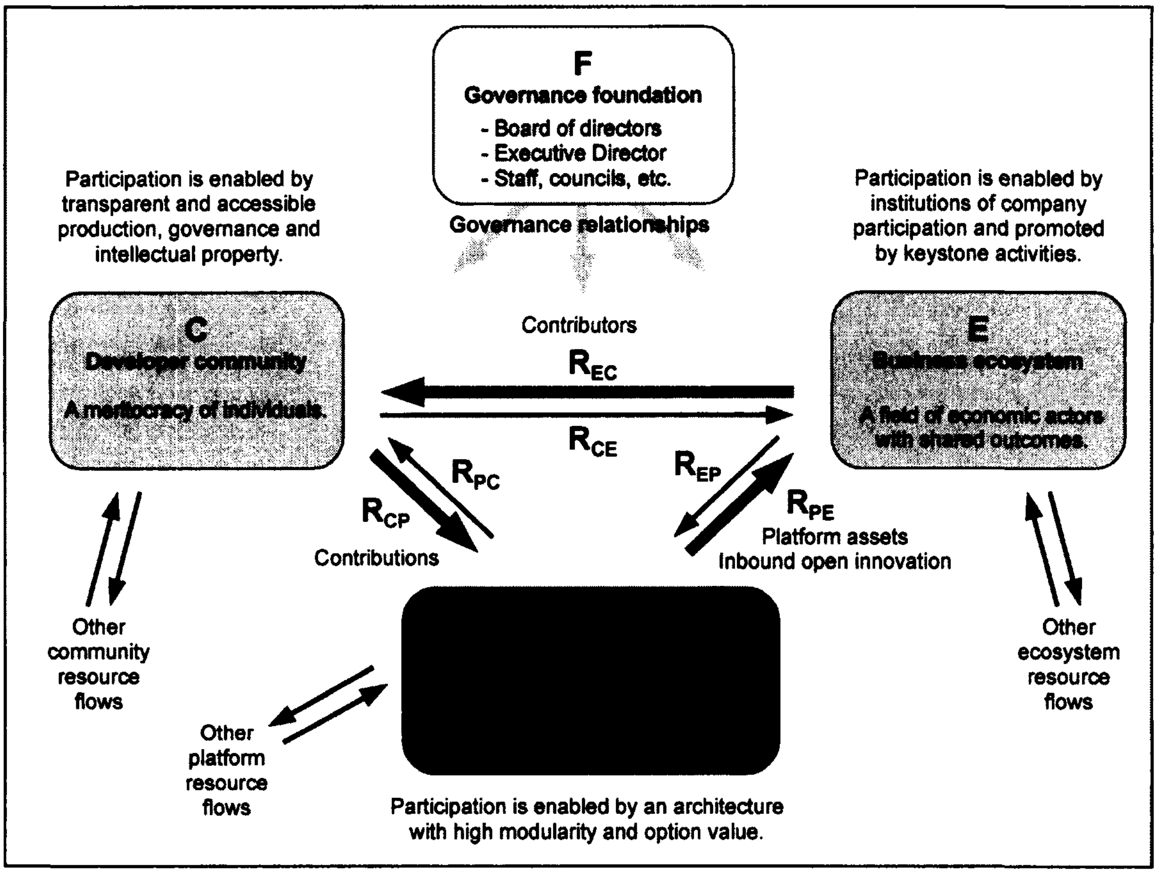

F
Governance foundation
- Board of directors
- Executive Director
- Staff, councils, etc.
Governance relationships
Participation is enabled by transparent and accessible production, governance and intellectual property.
Participation is enabled by institutions of company participation and promoted by keystone activities.
C
Developer community
A meritocracy of individuals.
E
Business ecosystem
A field of economic actors with shared outcomes.
Contributors
R_{EC}
R_{CE}
R_{PC}
R_{CP}
Contributions
R_{EP}
R_{PE}
Platform assets
Inbound open innovation
Other community resource flows
Other ecosystem resource flows
Other platform resource flows
Participation is enabled by an architecture with high modularity and option value.

Figure 19: Resource cycle of participation (situating new and extant theory)

The community-developed platform (P) is a set of building blocks and complementary assets which firms and individuals can use and consume to develop complementary products, technology and services. It is a platform in the technical sense of Gawer (2009a) and the platforms literature (section 3.3), and a knowledge commons in the technical sense of Hess & Ostrom (2003) and the institutional literature (section 5.3). *The developer community (C) is the collective of individuals, organized as a meritocracy, who together maintain and extend the platform.* It is an open source developer community in the technical sense of West & O'Mahony (2008) and the research literature on community software development (section 3.6). *The business ecosystem (E) is the set* *of economic actors whose individual business activities, anchored around the platform, share in some large measure the outcomes of whole ecosystem.* It is a business ecosystem in the technical sense of Moore (2006) and the business ecosystems literature (section 2.4), and an organizational field in the technical sense of the institutional literature (DiMaggio & Powell, 1983; Scott, 2008). The governance foundation (G) is at once an open source software foundation (section 3.6; O'Mahony, 2005; Xie, 2008; Peters, 2009), a business ecosystem keystone (section 2.5; Iansiti & Levien, 2004a, 2004b), and a set of polycentric governance structures (section 5.1; Ostrom, 1990).

Each node in Figure 19 transforms inbound resources into different outbound resources. The principal flow of resources is *contributions* from developer community to platform (R_{CP}), *platform assets* from platform to business ecosystem (R_{PE}), and *contributors* from business ecosystem to developer community (R_{EC}). As is typical for resource exchange relationships in resource dependence theory (Pfeffer & Salancik, 1978), each principal resource flow is paired with a resource flow in the opposite direction. In the case evidence of the Eclipse field setting, those reciprocal resource flows included information and recruitment of staff by ecosystem companies (R_{CE}), assets of use-value to the community and the inputs to incremental community innovation (R_{PC}), and code contributions from ecosystem companies to the platform (R_{EP}), respectively.

Figure 19 also situates the prior work of Baldwin & Clark (2006) and West & O'Mahony (2008). According to Baldwin & Clark (2006), code base architectures that are more modular and have more option value increase the incentives for developers to join and remain involved in an open source development effort, relating the properties of the platform (*P*) to community contributions (R_{CP}) and code contributions from ecosystem organizations (R_{EP}). According to West & O'Mahony (2008), sponsors of open source communities consider the transparency and accessibility of three community design dimensions – the processes of production (they way in which the community produces software), governance (the way in which decisions are made), and intellectual property (the allocation rights to use the community's output) – and higher levels of transparency and accessibility are associated with attracting external participants and growing communities. This argument relates the properties of the developer community (*C*) to incoming resource flows (including R_{EC}), and connects the developer community (*C*) with the governing foundation (*F*) through the rules of production, governance, and IP. Both Baldwin & Clark (2006) and West & O'Mahony (2008) partially operationalize O'Reilly's (2005a) argument that some systems possess an "architecture of participation" that provides favourable incentives for contribution and participation (chapter 4). None of these extant explanations explicitly address the business ecosystem (*E*).

The second theoretical contribution of this research is a theory of participation that explicitly includes the business ecosystem (node *E*), and links the flow of contributors from the ecosystem to the developer community (R_{EC}) and the flow of code contributions from the ecosystem to the platform (R_{EP}) to the business incentives of ecosystem companies. More specifically, companies participate to strengthen or transform their business models – by securing access to required capabilities, benefiting from cost savings and efficiencies of open source development, and enabling new business models that were not otherwise viable. One implication is that business ecosystems that more readily provide these benefits will be more appealing to participants and thus more likely to attract higher levels of participation, resulting in more community resources and direct code contributions to the platform. Participation in the system of Figure 19 is driven not only by an *architecture of participation* (O'Reilly, 2005a) with high *modularity* and *option value* (Baldwin & Clark, 2006) and a *transparent and accessible open source community* (West & O'Mahony, 2008), but also by interdependent *institutions of participation* and actions by the keystone governing foundation that motivate resource flows from economic actors in a business ecosystem to development activities within the open source software community. Explanations for participation in the Eclipse ecosystem (section 11.2) included several institutional factors on which Baldwin & Clark (2006) and West & O'Mahony (2008) were silent:

- As a membership-based not-for-profit organization with a vendor-neutral governance model, the Eclipse Foundation does not compete with ecosystem companies to appropriate value from the Eclipse platform.
- The Eclipse IP due diligence process provides adopters with confidence that Eclipse software can be safely distributed in commercial products.
- The Eclipse annual release train provides coordinated software releases on a predictable timeline.
- The *Eclipse Development Process* includes requirements for active community-building by project committers (beyond accessibility and transparency).

Furthermore, the Eclipse Foundation promotes participation through *keystone activities*:

- The Eclipse Foundation actively promotes awareness of the Eclipse platform.
- Recognizing that the attractiveness of the Eclipse platform is related, in part, to large, active, and engaged user and adopter communities, the Eclipse Foundation actively works to grow the user and adopter communities.
- The Eclipse Foundation performs ecosystem development services that directly benefit member companies.
- The Eclipse Foundation attends deliberately to both the open source developer community and the business ecosystem, adequately managing and sometimes harnessing the tensions that arise.

These results have several practical implications that suggest prescriptions for practice. First, an organization seeking to advance a community-developed platform should attend not only to the platform and development community, but also the business ecosystem and the flow of resources from the business ecosystem to the community and platform. Second, harming the community to benefit the ecosystem would break the resource cycle just as surely as neglecting or harming the ecosystem. Thus, an organization seeking to advance a community-developed platform should identify and deliberately manage the tensions that naturally arise between community and ecosystem, so as not to favour one to the detriment of the other. Ideally, tensions can be harnessed to improve the system. Third, growing the number of economic actors in the ecosystem that use and consume the platform does not necessarily grow the flow of resources from the ecosystem to the community; case evidence from Eclipse suggests that this resource flow from ecosystem companies is motivated by pragmatic business considerations: enhancing a business model, increasing the stakeholder value of a firm's offer, and improving a firm's competitive advantage with respect to its rivals. Thus, an organization seeking to advance a community-developed platform should also attend closely to the incentives for company contribution to platform development, and the ways in which a governance foundation can enhance these incentives or remove obstacles.

Likewise, these results have several implications for management theory and future research. First, this research has joined the research literature on open source developer communities to the research literature on business ecosystems. Treating the community construct and the ecosystem construct as distinct yet codependent provides theoretical clarity over the conceptual alternative of enfolding one construct into the other – a common practice in prior research that typically emphasized a subset of the larger field setting examined here. Second, the IAD framework has been a useful tool for structuring data collection, organizing inquiry, and focusing attention on institutional factors. Future management studies of other field settings may likewise find IAD useful for similar purposes. Third, the practitioner writing on architectures of participation (e.g., O'Reilly, 2005a; chapter 4) was a valuable source for conceptual ideas when interpreting empirical observations and theorizing about institutions of participation. There may yet be more to learn by growing the list of observed heuristics compiled in section 4.1, and from operationalizing and rigorously specifying these participation heuristics.

More generally, these results are further evidence of the theoretical relevance of the business ecosystem concept. The Eclipse ecosystem, as envisioned by Eclipse practitioners and specified as an operationalized construct in section 11.1, is unlike any other inter-firm construct from the scholarly literature surveyed in section 3.5. Its membership cuts across industry boundaries, including firms variously associated with the electronic hardware, computer software, consumer electronics, enterprise services, mobile telephony, automotive, aerospace, and education sectors, while other firms from those same sectors choose not to participate. It is not a value chain, a value network, a value constellation, a community of practice, a business group, or a strategic alliance (Table 9, section 3.5); it is at once both broader and narrower than each of these constructs, including phenomena that each excludes, while excluding phenomena within each construct's scope. Scholars such as Simon (1947), Cyert & March (1963), Weick

(1989, 1995), Senge (1990) and Morgan (2006) argue that manager behaviour and discourse are directly relevant to research programs seeking to describe, explain and predict the manager's world, and the field observations of chapters 7, 8, 9, and 10 are evidence of the central role of the ecosystem concept in practitioner mental models (Senge, 1990), metaphors (Morgan, 2006), vocabulary (Astley & Zammuto, 1992), and the enacted theories implied by business models (Chesbrough & Rosenbloom, 2002; Chesbrough, 2006; Johnson et al. 2008; Johnson, 2010). These results sharpen the arguments of Iansiti & Levien (2004a, 2004b), Moore (2006), Chesbrough & Appleyard (2007), Teece (2007, 2009), and others, that business ecosystems should be central to scholarly theories of innovation, strategy, organization, and entrepreneurship. For example, the business ecosystem construct extends scholarly rigour to notions that are valued by practitioners but absent or under-appreciated in traditional business strategy (Chesbrough & Appleyard, 2007) – such as the dynamics of cooperative behaviour among competitive rivals (chapter 10), strategic choices of firms to intentionally forgo excludability of innovation (chapter 10), actions to drive adoption of non-monitized and non-excludable assets (chapter 9), the interdependent architectural designs of technology platforms and communities of voluntary participants (chapter 8), and dependence on strategically-critical resources that a firm neither owns nor controls (chapter 7).

This section has discussed the theoretical contribution of this study's results, and closely examined the first two of four theoretical contributions. The next section briefly considers alternatives to the explanation of an interdependent community and ecosystem.

11.4 Alternative formulations and constructs

An alternative to representing Eclipse as co-dependent community and ecosystem constructs would be to extend either the community construct or the ecosystem construct to encompass the other. This is the implicit approach taken in many previous studies of communities and ecosystems.

The developer community construct defined here is comprised of individuals who maintain and extend the platform; it emphasizes the *creation* aspect of innovation, and the creation of economic value. Alternatively, some researchers have defined community more broadly to include membership by firms and other organizational actors as well as individuals, thus admitting the economic actors of the ecosystem into the community construct and obviating the ecosystem construct. Examples of community constructs that include organizational actors include Lynn et al. (1996), von Burg & Kenney (2003), Miles et al. (2005), von Hippel (2005), Culpan (2009), and Snow et al. (2009).

The business ecosystem construct defined here is comprised of economic actors with business activities anchored around a platform, the outcomes of which are joined to those of the platform and of other economic actors in the collective. It emphasizes the *commercialization* aspect of innovation, and the appropriation (or capture) of economic value. Ecosystem participants – primarily for-profit corporations – add value to the platform by providing complementary products and services, and by providing commercial offers built on Eclipse assets (through network effects, signaling, and evidence of platform value). Studies of business ecosystems typically identify *niches* – categories of participants classified according to the ways in which actors utilize or add value to the platform. Examples include Iansiti & Levien (2004a), Iansiti & Richards (2006), Iyer & Davenport (2008), West & Wood (2008), and Isckia (2009). A developer community could be incorporated into this theoretical framework as one more niche of an ecosystem, anchored around a multi-sided platform (Rochet & Tirole, 2003; 2006; Armstrong, 2006; Boudreau & Hagiu, 2009; Evans, 2009; Gawer, 2009b; Hagiu, 2009).

The co-dependent community and ecosystem constructs proposed here offer several benefits over either alternative approach. First, separate co-dependent constructs are consistent with the working vocabulary of Eclipse practitioners, the depiction of the Eclipse field setting in governance documents, and practitioner explanations of the Eclipse field setting (section 7.7). The Eclipse Foundation recognizes and deliberately nurtures both an Eclipse community and an Eclipse ecosystem as part of its purposes set forth in its *Bylaws* and in its observed behaviour (section 7.1), and manages and harnesses the tensions between them (section 10.4). Second, distinct community and ecosystem constructs linked through resource flows have interesting implications for theory and practice that are not evident when the constructs are merged or blended. In particular, the different ways in which ecosystem companies resource the developer community and the ways in which ecosystem companies obtain resources from the developer community, are brought into sharp focus (sections 10.2 and 11.3). Third, this approach accommodates the growing research literatures of both constructs, which have developed in different but complementary directions, starting from different source literatures, motivations, and assumptions. It joins these two literatures, without reshaping one to fit the other. A community comprised of individuals rather than higher-level social actors is consistent with common usage of the community construct in sociology (Brint, 2001) and organizational behaviour (Brown & Duguid, 1991; Wenger & Snyder, 2000; Amin & Roberts, 2008), research on open source software communities (O'Mahony, 2003, 2005, 2007; von Krogh et al. 2003; Shah, 2006; Markus, 2007; O'Mahony & Ferraro, 2007; Sadowski et al. 2008; Mateos-Garcia & Steinmueller, 2008; West & O'Mahony, 2008), and common dictionary definitions: e.g., a group of people having common interests and a similar sense of identity (*American Heritage Dictionary of the English Language*, 2009). There is an appealing symmetry between the platform, community, and ecosystem constructs of the participation resource cycle of section 11.3, and the technological, organizational, and industry structure levels of Baldwin & Clark's (2000; 2006) design rule theory surveyed in section 3.4, further linking these literatures at multiple levels of analysis.

11.5 Enfolding the commons and institutional literatures

The third theoretical contribution of this research is a set of insights enriching both the commons and ecosystems literatures. This is made possible by the IAD specification of the Eclipse field setting, which joins the scholarly literature on commons governance with the literatures on business ecosystems, communities, and platforms.

Figure 20 summarizes three sets of posited associations from prior IAD commons research: the general design principles of Ostrom (1990), the requirements of adaptive governance in complex systems (Dietz et al. 2003; Ostrom & Hess, 2007), and factors under which effective commons governance is easier to achieve (Dietz et al. 2003). These provide an interpretive lens through which to re-examine the workings of Eclipse.

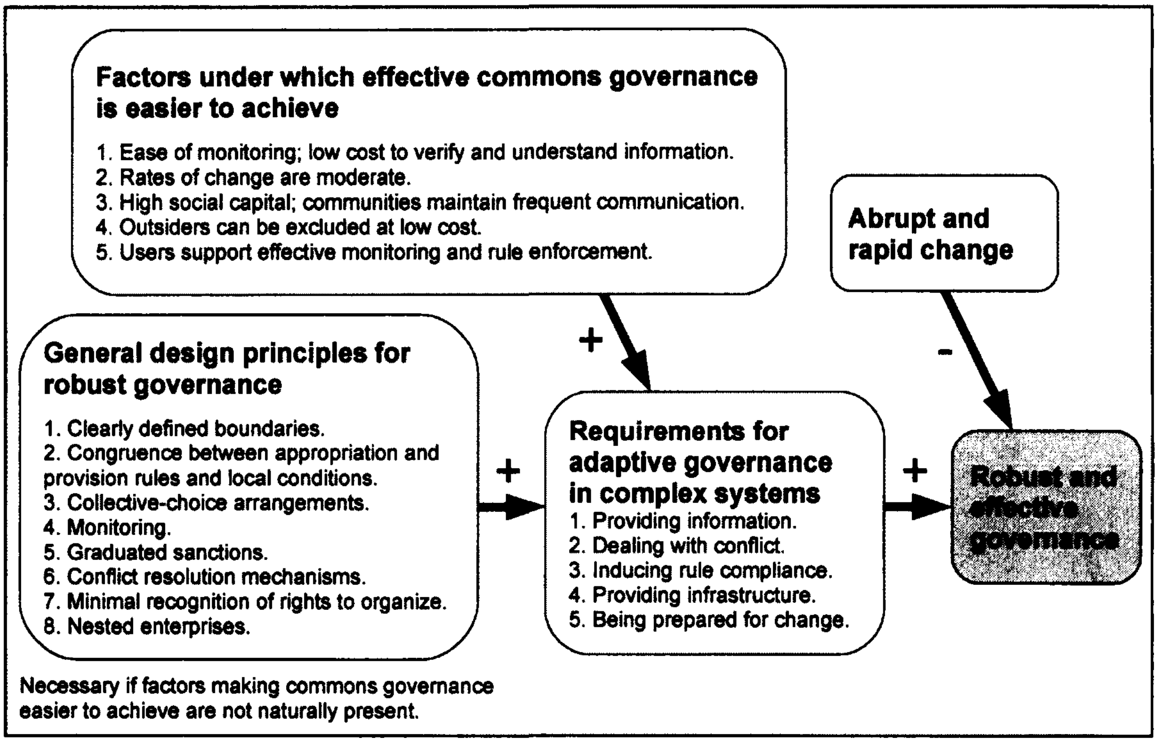

Factors under which effective commons governance is easier to achieve
1. Ease of monitoring; low cost to verify and understand information.
2. Rates of change are moderate.
3. High social capital; communities maintain frequent communication.
4. Outsiders can be excluded at low cost.
5. Users support effective monitoring and rule enforcement.
Abrupt and rapid change
+
-
General design principles for robust governance
1. Clearly defined boundaries.
2. Congruence between appropriation and provision rules and local conditions.
3. Collective-choice arrangements.
4. Monitoring.
5. Graduated sanctions.
6. Conflict resolution mechanisms.
7. Minimal recognition of rights to organize.
8. Nested enterprises.
+
Requirements for adaptive governance in complex systems
1. Providing information.
2. Dealing with conflict.
3. Inducing rule compliance.
4. Providing infrastructure.
5. Being prepared for change.
+
Robust and effective governance
Necessary if factors making commons governance easier to achieve are not naturally present.

Figure 20: Synthesis of prior research on robust and effective commons governance

From cross-case analysis of hundreds of IAD case studies, Ostrom (1990) found that an extremely rich variety of specific rules were used in systems sustainable over a long time period, and no set of specific rules had a clear association with success. Ostrom (1990, pp. 90-102) identified *eight general design principles for robust governance* of common pool resources: (1) clearly defined boundaries, (2) congruence between appropriation and provision rules and local conditions, (3) collective-choice arrangements, (4) monitoring, (5) graduated sanctions, (6) conflict resolution mechanisms, (7) minimal recognition of rights to organize, and (8) nested enterprises. Hess & Ostrom (2007, p. 7) write:

> *The eight factors identified were those found to exist in most robust institutions – but they were absent in failed systems. These principles have inspired hundreds of studies. And they are, indeed, helpful as a possible place to start an*

> *investigation. But they are in no way prescriptive – nor are they models. Rather, they are insightful findings in the analysis of small, homogeneous systems. Whether they apply to the study of large and complex systems like the knowledge commons is a question for further research.*

Dietz et al. (2003; see also Ostrom & Hess, 2007, pp. 66-68) argue that the general design principles are effective because they combine in various ways to create *five requirements of adaptive governance in complex systems*: (1) providing information, (2) dealing with conflict, (3) inducing rule compliance, (4) providing infrastructure, and (5) being prepared for change. Each principle helps meet several requirements, and the principles and requirements interact together to complement one another.

In a broad survey of commons research, Dietz et al. (2003, p. 1908) identify *five factors under which effective commons governance is easier to achieve*: (1) resources and use of resources can be monitored, and the information can be verified and understood at relatively low cost, (2) rates of change in resources, user populations, technology, and economic and social conditions are moderate, (3) communities maintain frequent face-to-face communication and dense social networks that increase potential for trust, allow people to express and see emotional reactions to distrust, and lower costs of monitoring and inducing rule compliance, (4) outsiders (who may add to harvesting pressure and lack understanding of the rules) can be excluded at relatively low cost, and (5) users support effective monitoring and rule enforcement. Two consistent results in empirical commons research are (1) successful commons governance requires rules that evolve (Dietz, 2003, p. 1908) and (2) institutional arrangements for collective action can and do often fail when rapid change occurs faster than rules can evolve (Dietz, 2003, p. 1907).

Within the Eclipse field setting, each of the five Dietz et al. (2003) and Ostrom & Hess (2007) requirements for adaptive governance of complex systems are present. First, there is vast publicly available information about nearly all aspects of Eclipse (subsection 6.2.6) and infrastructure to obtain that information (subsection 9.4.1). Second, the Eclipse Foundation effectively manages conflict and tension through transparency of decisions and decision-making, frequent and ongoing conversation, and deliberately attending to the interests of both the developer community and the commercial ecosystem (section 10.4). In this way, it harnesses tensions to improve the system. Third, Eclipse institutions induce rule compliance through regular project reviews (8.2.6), the IP due diligence process (subsection 8.2.2), community norms to conform to process (section 8.3), and enforcement mechanisms that are rarely used by nonetheless credible and available (subsection 8.2.8). Fourth, project and community infrastructure is extensive and high quality (subsections 8.1.4 and 9.4.1). Fifth, continuous change is normal and expected; the polycentric governance structures of the Eclipse Foundation revise the developer community and business ecosystem operating rules by updating guidelines and recommendations on short time-lags (days or weeks), periodically updating the *Eclipse Development Process* on medium time-lags (months to complete; has happened three times with one pending), and rarely updating the *Bylaws* on long time-lags (~6 months; requires membership vote; has happened once with one pending as of March 2011).

The five Dietz et al. (2003) factors under which effective commons governance is easier to achieve are not naturally present in the Eclipse field setting. (One exception is high social capital and frequent community conversation which are endemic to the culture of experienced open source software developers but unfamiliar to many companies, and thus may be said to be *partially* present.) Rather, the Eclipse Foundation creates the conditions for adaptive governance, in part by attending to the general design principles (Ostrom, 1990) in various ways. To be clear, there is no evidence in the case data or in previous Eclipse studies that Eclipse Foundation staff or other Eclipse practitioners were aware of the extant research on commons governance, or that anyone *deliberately* structured any aspects of Eclipse governance according to this literature. Nonetheless, the Eclipse field setting embodies many of the design principles that prior commons research has associated with robust and effective commons governance.

This analysis of extant research on commons governance offers two tentative insights for consideration by Eclipse practitioners – both related to the theme of monitoring. First, practitioners could consider enhancing the infrastructure for reporting and monitoring activity within the developer community and business ecosystem. The Eclipse Dash project, publicly assessable through the Eclipse.org website, provides detailed reports and graphical plots of committer activity based on commits to the Eclipse source code repositories[169]. However, there are many ways to contribute to an open source project, and as of 2011, Dash reports only code commits with several known issues with the data and its presentation. There are occasional calls from the community for upgrading the reporting capabilities of Dash (Rapicault blog, November 29 2009):

> *If we want to use dash as a reliable hint on the activity and diversity, then we should revise how the numbers are being computed to take into account: patch author instead of committer, activity in bugs, activity on [mailing lists], number of [people] asking questions in forums, etc...*

[169] Dash queries provided many of the data for the tables in chapter 8 comparing activity on case projects.

For example, the annual *Eclipse Community Survey* (2007, 2009, 2010, 2011) inquires about six forms of contribution in addition to code commits: submitting bug reports to Eclipse Bugzilla, responding to forum posts, submitting a patch to Bugzilla (i.e., a code contribution), writing a blog post or article, making a conference presentation, and participating in testing. In particular, code contributions from non-committers ("contributors" in Eclipse parlance) are especially valued, but contribution information is tracked by each project in different ways and is presently less accessible and less detailed than commit information[170]. Second, as a refinement of the first point, the Eclipse Foundation could consider hosting the capability to map and dashboard the Eclipse ecosystem. Iyer et al. (2006), Li (2009), Weiss & Gangadharan (2010) and others (reviewed in section 3.5) argue that much can be learned from mapping an ecosystem, but the data to produce useful and accurate network diagrams is often difficult to obtain. The various Eclipse archival databases include many of the needed data, and the automated capability to easily obtain, manipulate, and visualize these data could be a natural extension to reporting capability already in place. The perennial issue with both recommendations is the scarcity of time and attention. The individuals with the capability to extend the community monitoring resources are the committers to Eclipse projects and the Eclipse Foundation staff who maintain the Eclipse IT infrastructure, all of whom have many alternative demands on their time. Unless new participants come

[170] As an illustrative example, the contributor data about case projects in chapter 8 (i.e., Table 39) were obtained by manually examining the IP Log documents for the most recent release of each case project. IP logs are a PDF document that includes the name and email address provided by the contributor of each code contribution of more than 250 lines of code. Organizational affiliations were inferred from email addresses, but this approach is prone to various sources of error and under-counts the employers of contributors who chose to provide an email address with no organizational affiliation.

forward or existing participants increase their participation specifically for this work, enhancements to the Eclipse Dash project would divert resources from other development and community-building activities core to the Eclipse field setting. Nonetheless, past research on commons governance suggests that improved and expanded monitoring, along the lines proposed above with easy access to more and better contribution metrics on the community and ecosystem, may warrant further consideration.

The larger contribution from joining these literatures may be from the business ecosystems literature to the nascent knowledge commons literature. At least three specific insights from this research could usefully inform the commons literature: (1) the co-dependent resource exchanges between the Eclipse developer community and the Eclipse business ecosystem, (2) the mechanisms through which Eclipse manages platform complexity and provides predictability of a large and complex commons resource, and (3) the keystone activities performed by the Eclipse Foundation to promote the platform and to add value to Foundation membership. Each – described briefly below – is a promising avenue for further contribution to the commons literature[171].

The first insight to the commons literature builds on the recognition of Hess & Ostrom (2007) that the communities (in the IAD sense of chapter 5) of actors around a knowledge commons may be partitioned into multiple roles, such as resource providers, resource users, and policy makers, and that those roles may often be nested. This study's representation of the Eclipse developer community as an institution of individual

[171] A full integration of any of these insights into the scholarly literature on IAD and the new knowledge commons is outside the scope of the guiding research questions that motivated this study. However, each is a possible future project and an opportunity to re-interpret the study's data and results for a new audience.

participation in maintaining and extending the Eclipse platform, the Eclipse ecosystem as an institution of company participation, and the Eclipse Foundation as a collection of polycentric governance structures, provides a concrete example of multiple commons roles that are at once richer and more nuanced than the provider, user, and policy maker constructs of the extant knowledge commons literature (Ostrom & Hess, 2007).

The second insight builds on the commons literature observation that collective action on commons resources that are large, complex and unpredictable is more difficult to sustain than collective action on commons resources that are small, simple, and predictable. Although the Eclipse platform is both large and complex in comparison to other software systems (subsection 7.8.4), and software development in general is difficult to plan and predict (Brooks, 1995; Pressman, 2009), Eclipse employs specific mechanisms to manage its complexity and to provide predictability which, according to practitioner explanations, are factors in the adoption and continued growth of the platform. More specifically, the size and complexity of the Eclipse resource platform are mitigated, in part, through architectural modularity and well-specified public design rules (section 7.4). The modularity and design literature reviewed section 3.4 argued that strong modularity enables new work to proceed in parallel with few external dependencies, simultaneously multiplying and decentralizing design options (Baldwin & Clark, 2000). Although the over-all system is enormously complex, through modularity and design rules, much of the system complexity is hidden much of the time. The predictability of the Eclipse platform results from deliberate design and significant effort. The Eclipse annual release train (subsection 8.4.1 and section 9.6) delivers a coordinated release of tested software on the fourth Wednesday of June each year without ever missing a ship date. Requirements and community norms for transparency of developer decisions and project plans inform users and adopters of goals and progress towards goals (subsections 8.1.2, 8.2.3 and 8.2.5). The Eclipse IP due diligence process (subsections 8.2.2 and 9.4.2) provides adopters with confidence that Eclipse software can be safely distributed in commercial products. The Eclipse Foundation has little direct influence on which tasks the developer community chooses to undertake, but it has rules to place to ensure that project activities are well communicated to users and adopters. A close study of these specific mechanisms employed within Eclipse, and the applicable insights from the modularity and design literature (section 3.4) and the practitioner literature on architectures of participation (chapter 4) could potentially enrich the commons literature on effective collective action to preserve and maintain large and complex resources.

The third insight identifies a cluster of salient observations of the Eclipse field setting that fit awkwardly within the IAD framework. The Eclipse Foundation undertakes various activities to promote the Eclipse platform and to benefit member companies that feature prominently in the emergent explanation for company participation (section 11.2). Activities to promote the Eclipse platform include professionally-managed public relations and media relations for platform releases and other major events, deliberately growing the Eclipse Foundation membership, participation in standards and industry groups to represent Eclipse Foundation interests, and organizing community events to promote Eclipse open source tools and frameworks (section 9.4). Activities to benefit member companies include optional programs such as marketing services, assistance with press releases, and premium placements of member products. These marketing and ecosystem development activities are collectively and deliberately tracked at each quarterly board of directors meeting, and figure prominently in the explanations developed here for the benefits of company participation (subsection 10.2.4 and section 11.2) and the characteristics of the Eclipse field setting that promote company participation (section 11.2). In the IAD table shells developed in subsection 6.2.2, these *keystone activities* (Iansiti & Levien, 2004a) have been represented variously as rules-in-use or resource attributes in chapters 8 and 10. Future researchers could benefit from refining these notions and including them in the IAD checklists of variables and concepts for studies of knowledge commons (section 5.1).

In summary, various insights from this discussion can enrich both the ecosystems literature and the commons literature. Past research on robust and effective commons governance suggests several considerations for structuring business ecosystems anchored around community-developed platforms. Factors such as the availability of timely and high quality information, conflict management, infrastructure, trust and social capital, the rate of change, and frequency and quality of communication have been key explanatory variables in prior IAD commons studies of other systems. Although the institutions of the Eclipse field setting already attend to many of these themes, two possible areas for further attention are (1) monitoring infrastructure to report a broader range of metrics on activity within the Eclipse developer community and Eclipse projects, and (2) monitoring infrastructure to dashboard and visualize the Eclipse ecosystem and the relationships between companies in the ecosystem and activity within Eclipse projects. Likewise, the

IAD specification of the Eclipse field setting offers several insights to the commons literature. In particular, (1) the co-dependent resource exchanges between the developer community and the business ecosystem can enrich the notion of IAD community roles, (2) the mechanisms to manage platform complexity and provide predictability can enrich the commons notions of resource size, complexity and predictability, and (3) keystone activities to promote platform adoption and membership value could be incorporated into IAD checklists to benefit future knowledge commons researchers.

The fourth and final theoretical contribution of this research is a set of broader insights about institutional theories of management and complementarities between an institutional perspective and other management perspectives. Institutional theories have traditionally emphasized conformity and inertia, through imprinting that persists over time (Sinchcombe, 1965), isomorphism (Hawley, 1968), legitimation (Meyer & Rowan, 1977), mechanisms that "make organizations more similar without necessarily making them more efficient" (DiMaggio & Powell, 1983), and limited repertoires of action (Oliver, 1991). Scott (2008, p. 152) notes "a remarkable similarity in the structural features of organizational forms operating within the same organizational field." This research is an existence proof of novel theory production by applying institutional concepts differently – to the study of technical and business model innovation, semi-structures combining constrained and unfettered action, and voluntary joining and exiting – and as a complement to resource dependence, theories of design, and systems thinking.

This chapter discussed the research results, answered the research questions, and situated the contribution of the research. The next chapter concludes the research.

12 Conclusions

Platforms – and business ecosystems of co-dependent companies with businesses anchored around platforms – are increasingly ubiquitous in information and communication technology and other technology-intensive industries. Yet there is a paucity of management research and theory regarding open and community-developed platforms, and participation by the individuals and organizations that maintain and extend these platforms. This doctoral thesis has addressed that gap with a multilevel empirical investigation of participation in the Eclipse Foundation, community, and ecosystem. It developed and analyzed eighteen case studies of projects, companies, and polycentric governance groups nested within the top-level case of the Eclipse field setting. Employing the constructs and conceptual apparatus of the *Institutional Analysis and Design* (IAD) framework (Ostrom, 1990, 2005; Ostrom & Hess, 2007), it induced a multilevel theory of participation that relates together the community, platform, and ecosystem through resource flows and shared institutional structures. It joins the literatures on business ecosystems and open source developer communities, extends the nascent theory of participation advanced by Baldwin & Clark (2006) and West & O'Mahony (2008), and offers prescriptive implications for management practice.

Like all research designs, this study has specific limitations. First, it has all of the well-known limitations of single case inductive studies (March, 1991; Yin, 2003a, 2003b; Eisenhardt & Graebner, 2007; Poteete et al. 2010). Section 6.4 documented the tactics implemented here to produce results with strong construct validity, internal validity, and reliability, and the use of theory as a best practice (Yin, 2003a) to increase the likelihood of strong external validity; nonetheless, the specific extent to which the results of a single case study generalize to other field settings remains an empirical question for follow-on research (Whetten, 1989; Siggelkow, 2007). Second, by drawing deeply on a theoretical sample of projects, companies, and data sources, the research attends closely to events within the sample cases but is uninformed of events outside the sample. Section 6.2 and chapters 8 and 10 explained the sampling rationale and the appropriateness of this sample for the guiding research questions, the operational case study questions, the field setting of this research, and the relatively immature state of extant research, but the possibility remains that more could be learned by further study of other cases. Third, by emphasizing specific aspects of the Eclipse field setting – particularly the maintenance and extension of the Eclipse platform and the companies that participate in activities to maintain and extend the commons – it is a partial view of a more complex system that also includes (for example) a user community, an adopter community, and other categories of participant in the Eclipse ecosystem (such as independent individuals and organizations other than business enterprises). It is possible that understanding the microstructure of these other structures and the behaviour of other ecosystem participants could further enrich a theory of company participation. Likewise, by focusing attention on the main effects of the resource cycle of platform innovation, this thesis has attended less closely to other resource flows which may also have interesting implications.

There are at least four areas of promising follow-on research that build on the results of this research and partially address some of its limitations. First, there may be more to be learned from further close study of Eclipse. Additional nested case studies could include (1) mature open source projects that moved to Eclipse from other institutional settings and the organizations participating in their development, (2) enterprise participation in industry verticals such as the new Eclipse automotive working group – potentially a fourth wave of Eclipse Foundation membership growth by organizations with motivations that may differ from those examined here, with interactions structured by different rules-in-use[172], (3) companies that formerly participated actively in Eclipse but later became less active or inactive, or (4) companies that heavily use and consume Eclipse platform assets but do not participate in maintaining and extending the platform. As well, this nested case study design could be complemented by other research designs, such as ecosystem mapping and visualization, or computation of quantitative scales for variables of interest, for a more complete picture. Furthermore, there may be opportunity to enfold other literatures to enrich and sharpen the explanations of other parts of the Eclipse field setting – for example, the technology adoption literature could have much to say about the role of specific institutional factors in the broad user adoption of Eclipse tools, which practitioners associated with the attractiveness of the Eclipse platform to participating companies.

Second, there is little extant research on the *process* of forming a healthy vendor neutral business ecosystem. The IAD representation of the Eclipse field setting, the nested case studies, and the governance timeline of this research could provide the

[172] Section 7.4 briefly discussed the Eclipse automotive working group within the context of the expanding scope of the Eclipse platform. See also the *Eclipse Automotive Working Group Charter* (2011) and the announcement on the eclipse.org-membership-at-large mailing list (Milinkovich, February 7 2011). Industry working groups operate outside of the *Eclipse Development Process*, as currently written.

starting point for a rigorous innovation process study employing the methods of Van de Ven et al. (1999) and Poole et al. (2000).

Third, much could be learned from replication of this research design on other systems. For example, the Apache Software Foundation, the Linux Foundation, and the GNOME Foundation each steward community-developed open source platforms, but with institutional structures that differ from those of Eclipse. The IAD framework and nested design of this research could provide a standard research design and case template for case studies that are easily and directly comparable. Standards of data collection and analysis could potentially benefit management research on business ecosystems in the same way that such standards were said to benefit interdisciplinary research of natural resources commons governance in the 1980s and 1990s (discussed in subsection 6.1.2).

Fourth, the emergent theoretical framework developed in section 11.3 relates architectural design, institutional design, participation, and resource flows. While it has strong internal validity to account for patterns in the field setting and cases examined here, further work is needed to refine, test, and improve it, and explore the bounds of its external validity to other contexts. Research using quantitative methods or observing natural experiments could seek to test various hypotheses implied by this framework.

Although much remains unknown, the rich description, explanation-building, and theorizing of this doctoral thesis are offered as a stepping stone to better understanding innovation around open community-developed platforms, and towards developing the capabilities to reliably launch, sustain, and grow new platforms, innovation communities, business ecosystems, and effective governance foundations.

13 References

References follow the Academy of Management style guidelines (*Academy of Management Review*, 2007) with two important exceptions. First, version-controlled Eclipse Foundation governance documents and reports with no explicit author information are listed separately in section 13.1, sorted by title and either version number or date. This category also includes Eclipse Wiki pages that are moderated by the Eclipse Management Organization (EMO). In-text citations to these documents use document name (italicized) and date or version number in parentheses if required, e.g., "*Eclipse Community Survey* (2010)," or "*Bylaws of the Eclipse Foundation* (2003)." The titles of wiki documents and meeting minutes capitalize the initial letter only; titles of stand-alone documents employ title case. Second, community discussions such as blog entries, posts to mailing lists, bug reports, and enhancement requests are listed separately in section 13.2 sorted by source type. In-text citations use a source identifier, followed by the date and other information if required, normally all in parentheses, e.g., "(Beaton blog, July 28 2010)." All other references, including press releases, are provided in section 13.3 formatted according to Academy of Management conventions.

13.1 Eclipse governance documents and reports

2005 Eclipse Roadmap. Version 1. A subset of the Version 1 Roadmap, the Version 1 ***Themes and Priorities*** created by the Requirements Council, is available online: http://www.eclipse.org/org/councils/themes.html

2006 Eclipse Roadmap. Version 2. Available online: http://www.eclipse.org/org/councils/roadmap_v2_0

2007 Eclipse Roadmap. Version 3. Available online: http://www.eclipse.org/org/councils/roadmap_v3_0

2008 Eclipse Roadmap. Version 4. Available online: http://www.eclipse.org/org/councils/roadmap_v4_0

2010 Eclipse Roadmap. Version 5. Available online: http://www.eclipse.org/org/councils/roadmap_v5_0

2011 Eclipse Roadmap. Version 6. Available online: http://www.eclipse.org/org/councils/roadmap_v6_0

Abridged minutes of the Eclipse Foundation board of directors. Meetings are held quarterly in March, June, September, and December. Abridged minutes are available for nearly all quarterly meetings since Q2 2004, and for some conference calls held between quarterly meetings. Available online: http://www.eclipse.org/org/foundation/minutes.php#board

Antitrust Compliance Policy. 2003. Version 1.0. Available online: http://www.eclipse.org/org/documents/Eclipse%20ANTITRUST %202003_11_10%20Final.pdf

Bylaws of the Eclipse Foundation. 2003. Available online: http://www.eclipse.org/org/documents/Eclipse%20BYLAWS %202003_11_10%20Final.pdf
[This is the original and founding version of the Eclipse Bylaws]

Bylaws of the Eclipse Foundation. 2008. Available online: http://www.eclipse.org/org/documents/Eclipse%20BYLAWS %202008_07_24%20Final.pdf
[This is the current version of the Eclipse Bylaws, revised July 24 2008. It is the first and only approved revision to the original 2003 Bylaws]

Committer Employer Consent Form. Available online: http://www.eclipse.org/legal/committer_process/employer_consent.pdf

Committer Guidelines. 2008. Accessed February 20 2011. Available online: http://www.eclipse.org/legal/committerguidelines.php

Common Public License (CPL). 2001. Version 1.0. Available online: http://www.opensource.org/licenses/cpl1.0.php

Community development for Eclipse projects. 2011. A section of the Eclipse.org wiki maintained by the Eclipse Architecture Council. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Community_Development_for_Eclipse_Projects

Development resources. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources

Eclipse.org Terms of Use. November 12 2008. Available online: http://www.eclipse.org/legal/termsofuse.php

Eclipse.org website. Last accessed August 1 2011. http://www.eclipse.org

Eclipse Automotive Working Group Charter. 2011. Available online: http://www.eclipse.org/org/industry-workgroups/autowg.php

Eclipse Community Survey. 2007. See Lawton (2007) in section 13.3.

Eclipse Community Survey. 2009. Available online: http://www.eclipse.org/org/press-release/Eclipse_Survey_2009_final.pdf

Eclipse Community Survey. 2010. Available online: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2010_Report.pdf

Eclipse Community Survey. 2011. Available online: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2011_Report.pdf

Eclipse Development Process (EDP). 2003. Revision 1.0. Available online: http://www.eclipse.org/org/documents/Eclipse%20Development%20Process%202003_11_09%20FINAL_old.pdf

Eclipse Development Process (EDP). 2007. Revision 2.3.1. [Approved by the Eclipse Foundation Board of Directors on January 17 2007; first approved revision to the 2003 EDP 1.0] Available online: http://wiki.eclipse.org/Development_Process_2006_Revision_Final

Eclipse Development Process (EDP). 2008. Revision 2.4. [Approved by the Eclipse Foundation Board of Directors on August 20 2008] Available online: http://www.eclipse.org/projects/dev_process/development_process_2008.php

Eclipse Development Process (EDP). 2010. Revision 2.5. [Approved by the Eclipse Foundation Board of Directors on May 19 2010] Available online: http://www.eclipse.org/projects/dev_process/development_process_2010.pdf

Eclipse Public License (EPL). Version 1.0. Available online: http://www.eclipse.org/org/documents/epl-v10.php

Eclipse Simultaneous Release. 2010. A document maintained by the Eclipse Planning Council. Available online: http://eclipse.org/indigo/planning/EclipseSimultaneousRelease.php

Eclipse Standard Top-Level Charter v1.1. http://www.eclipse.org/projects/dev_process/Eclipse_Standard_TopLevel_Charter_v1.1.php

EPP/How to create a package. 2010. A document in the Eclipse.org wiki maintained by the Eclipse Packaging Project. Last accessed February 26 2011. Available online: http://wiki.eclipse.org/EPP/How_to_create_a_package

Executive Director Report. The Eclipse Foundation Executive Director issues annual and quarterly reports to the Eclipse Membership. From 2004 Q3 to 2008 Q1, these reports were posted online with minutes of the quarterly Members Meetings: http://www.eclipse.org/org/foundation/minutes.php#members

Guidelines for Eclipse Logos & Trademarks. 2009. Available online: http://www.eclipse.org/legal/logo_guidelines.php

Incubation phase. A document in the Eclipse.org wiki maintained by the Eclipse Management Organization. Last accessed February 26 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Incubation_Phase

Indigo Simultaneous Release Plan. A document in the Eclipse.org wiki maintained by the Eclipse planning council. Last accessed February 26 2011. http://wiki.eclipse.org/Indigo_Simultaneous_Release

Individual Committer Agreement. Accessed February 20 2011. Available online: http://www.eclipse.org/legal/committer_process/ EclipseIndividualCommitterAgreementFinal.pdf

Intellectual Property Policy (IP Policy). Effective September 15 2008. Available online: http://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf

Membership Agreement. January 5 2010. http://www.eclipse.org/org/documents/ Eclipse%20MEMBERSHIP%20AGMT%202010_01_05%20Final.pdf

Member Committer Agreement. Accessed February 20 2011. Available online: http://www.eclipse.org/legal/EclipseMemberCommitterAgreementFinal.pdf

Minutes of the architecture council. Available online: http://wiki.eclipse.org/Architecture_Council/Meetings

Mylyn restructuring review. April 7 2010. Available online: http://eclipse.org/project-slides/mylyn-restructuring-review.html

New committer handbook. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/New_Committer_Handbook

Nominating and electing a new committer. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/ HOWTO/Nominating_and_Electing_a_New_Committer

Parallel IP process. 2010. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Parallel_IP_Process

Project meta-data. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Project_Meta-Data

Project plan. 2011. A section of the Eclipse.org wiki maintained by the Eclipse Architecture Council. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Project_Plan

Project Plan for Eclipse Packaging Project. Version 1.3. Last accessed February 15 2011. Available online: http://eclipse.org/projects/project-plan.php?projectid=technology.packaging

Proposed membership changes. June 24 2008. Available online: http://www.eclipse.org/membership/vote2008/v080624%20-%20PROPOSED%20MEMBERSHIP.pdf

Release reviews. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Release_Reviews

Simultaneous release compliance grid. 2011. A report maintained by the Planning Council. Last accessed February 20 2011. Available online: http://eclipse.org/indigo/planning/SimultaneousReleaseGrid.php

Starting a new project. 2011. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/HOWTO/Starting_A_New_Project

Version numbering. 2009. A section of the Eclipse.org wiki maintained by the Eclipse Platform team. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Version_Numbering

Words of wisdom and bits of advice. 2011. A section of the Eclipse.org wiki moderated by the Eclipse Management Organization. Last accessed February 15 2011. Available online: http://wiki.eclipse.org/Development_Resources/Words_of_Wisdom_and_Bits_of_Advice

13.2 Eclipse community discussions

In-text citations to archived Eclipse community discussions, in the various forms of blog posts, bug reports, and posts to electronic mailing lists, employ the notation conventions described in the following subsections.

13.2.1 Blogs

Planet Eclipse is an Eclipse blog aggregator: http://www.planeteclipse.org

In-text citations to individual blog posts follow the format "NAME blog (date)" or "(NAME blog, date)" where "NAME" is the surname of the blog post author – e.g., "Beaton blog (January 28 2010)" or "(Beaton blog, January 28 2010)."

Blogs maintained by current and former Eclipse Foundation staff

Beaton, W. Eclipse Evangelist, Eclipse Foundation.
September 2005 – present: http://waynebeaton.wordpress.com

Freeman-Benson, B. Director of Committer Community, Eclipse Foundation.
August 2005 – July 2010: http://eclipse-projects.blogspot.com

Milinkovich, M. Executive Director, Eclipse Foundation.
April 2005 – present: http://mmilinkov.wordpress.com

Mueller, R. Director of Ecosystem Development (Europe), Eclipse Foundation.
January 2008 – present: http://ralph-at-eclipse.blog.de

Skerrett, I. Director of Marketing, Eclipse Foundation.
April 2005 – present: http://ianskerrett.wordpress.com

Smith, D. Director of Ecosystem Development, Eclipse Foundation.
December 2005 – April 2011: http://eclipse-ecosystem.blogspot.com

Other individual blog posts cited in this document

Azad, D. July 7 2009. IBM Rational open house.
http://thecoderlounge.blogspot.com/2009/07/ibm-rational-open-house.html

Kersten, M. September 16 2010. Mylyn promoted to top-level Eclipse project.
http://tasktop.com/blog/eclipse/mylyn-top-level-eclipse-project

Kersten, M. February 1 2011. Prediction #7: Web-based code editors become viable, cloud-based IDE platforms remain a pipe dream.
http://tasktop.com/blog/eclipse/prediction-7-web-ide-pipe-dream

Phipps, S. February 16 2011. Is Eclipse open-by-rule?
http://webmink.com/2011/02/16/is-eclipse-open-by-rule/

Rapicault, P. November 29 2009. Project diversity/Those lying numbers.
http://lenettoyeur-on-eclipse.blogspot.com/2009_11_01_archive.html

Ryall, K. December 2 2009. Eclipse Debugger for C/C++ (EDC).
http://nokiacarbideoneclipse.blogspot.com/2009/12/eclipse-debugger-for-cc-edc.html

Schaefer, D. April 7 2009. On the future of Eclipse.
http://cdtdoug.blogspot.com/2009/04/another-great-post-by-bjorn-on-life-at.html

13.2.2 Bug reports and enhancement requests

Bugzilla is the Eclipse Foundation system for tracking bug reports and enhancement requests. Each bug report is assigned a unique ID number. http://bugs.eclipse.org

In-text citations to Eclipse Bugzilla reports follow the format “Bugzilla (#)” or “(Bugzilla, #)” – e.g., “Bugzilla (#301065)” or “(Bugzilla #301065).” Comments within a Bugzilla report are individually and sequentially numbered. If an in-text citation refers to a specific comment of a Bugzilla report, the comment number is noted in parentheses following the Bugzilla ID number.

Bugzilla #108146. Eclipse Release Train Name.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=108146

Bugzilla #215301. Standard project plan format.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=215301

Bugzilla #251304. Provide PDT 1.0.x builds on emft.eclipse.org.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=251304

Bugzilla #283745. Provide Maven repositories of stuff built at Eclipse.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=283745

Bugzilla #300000. Rename {Eclipse Project, DTP, WTP} Incubator project to ...Labs.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=300000

Bugzilla #300002. [EDP] PMC makeup description is wrong, or at least not right.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=300002

Bugzilla #300006. [EDP] Project Leadership selection process is wrong, or confusing.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=300006

Bugzilla #304878. [EDP] No more review calls. Ever.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=304878

Bugzilla #301065. EDP does not make clear the role of a container project.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=301065

Bugzilla #330226. Inclusion of m2e.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=330226

Bugzilla #337004. Eclipse security policy and procedures.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=337004

Bugzilla #342328. [EDP] Updates to coincide with proposed Bylaw changes in 2011.
http://bugs.eclipse.org/bugs/show_bug.cgi?id=342328

13.2.3 Developer mailing lists

Mailing lists are a primary communication channel for Eclipse developer discussions. Messages posted to a developer mailing list are distributed by email to subscribers of the list, and are archived for perpetuity at the Eclipse.org website. As of March 2011, there were more than 400 active Eclipse mailing lists.

In-text citations to specific posts on Eclipse mailing lists follow the format "LIST mailing list (NAME, DATE)" or "(LIST mailing list, NAME, DATE)" where LIST is the name of the mailing list, NAME is the name of the poster, and DATE is the date of the post – e.g., "eclipse.org-membership-at-large mailing list (Milinkovich, February 7 2011)" or "(eclipse.org-membership-at-large mailing list, Milinkovich, February 7 2011)" depending on context.

birt-dev mailing list archive:
http://dev.eclipse.org/mhonarc/lists/birt-dev

cdt-dev mailing list archive:
http://dev.eclipse.org/mhonarc/lists/cdt-dev

cross-project-issues-dev mailing list archive:
http://dev.eclipse.org/mhonarc/lists/cross-project-issues-dev

eclipse.org-architecture-council mailing list archive:
http://dev.eclipse.org/mhonarc/lists/eclipse.org-architecture-council

eclipse.org-membership-at-large mailing list archive:
http://dev.eclipse.org/mhonarc/lists/eclipse.org-membership-at-large

eclipse.org-planning-council mailing list archive:
http://dev.eclipse.org/mhonarc/lists/eclipse.org-planning-council

epp-dev mailing list archive:
http://dev.eclipse.org/mhonarc/lists/epp-dev

13.3 All other references

Academy of Management Review. 2007. Style guide for authors. 32(1): 313-316. Available online: http://www.aom.pace.edu/amr/AMRstyleguide.pdf

Abernathy, W. & Utterback, J. 1978. Patterns of industrial innovation. ***Technology Review***, 80: 40-47.

Ackoff, R.L. 1981. ***Creating the corporate future***. New York, NY: John Wiley & Sons.

Adner, R. 2006. Match your innovation strategy to your innovation ecosystem. ***Harvard Business Review***, 84(4): 98-107.

Adner, R. & Kapoor, R. 2010. Value creation in innovation ecosystems: how the structure of technological interdependence affects firm performance in new technology generations. ***Strategic Management Journal***, 31: 306-333.

Adomavicius, G., Bockstedt, J. Gupta, A. & Kauffman, R.J. 2008. Understanding evolution in technology ecosystems. ***Communications of the ACM***, 51(1): 117-122.

Alexander, C. 1964. ***Notes on the synthesis of form.*** Cambridge, MA: Harvard University Press.

Allen, R.C. 1983. Collective invention. ***Journal of Economic Behavior and Organization***, 4(1): 1-24.

Amabile, T.M., Patterson, C., Mueller, J, Womcik, T., Odomirok, P.W., March, M. & Kramer, S.J. 2001. Academic-practitioner collaboration in management research: a case of cross-profession collaboration. ***Academy of Management Journal***, 44(2): 418-431.

American Heritage Dictionary of the English Language (fourth edition). 2009. Boston, MA: Houghton Miffin Company.

Amin, A. & Roberts, J. 2008. Knowing in action: beyond communities of practice. ***Research Policy***, 37: 353-369.

Amit, R. & Schoemaker, P.J.H. 1993. Strategic assets and organizational rent. ***Strategic Management Journal***, 14: 33-46.

Amit, R. & Zoff, C. 2001. Value creation in e-business. ***Strategic Management Journal***, 22: 493-520.

Amoui, M., Salehie, M. & Tahvildari, L. 2009. Temporal software change prediction using neural networks. ***International Journal of Software Engineering & Knowledge Engineering***, 19(7): 995-1014.

Anderson, T. 2004. ***Eclipse now and in the future: an interview with Mike Milinkovich***. Available online: http://www.itwriting.com/eclipse1.php

Antikainen, M., Aaltonen, T. & Väisänen, J. 2007. The role of trust in OSS communities. ***Proceedings of the 3rd International Conference on Open Source Systems*** (OSS 2007): 223-228.

Anvik, J., Hiew, L, & Murphy, G.C. 2006. Who should fix this bug? ***Proceedings of the International Conference on Software Engineering.*** Shanghai, China.

Armstrong, M. 2006. Competition in two-sided markets. ***RAND Journal of Economics***, 37: 668-691.

Arthur, W.B. 1994. ***Increasing returns and path dependence in the economy***. Ann Arbor, MI: University of Michigan Press.

Arthur, W.B. 1996. Increasing returns and the new world of business. ***Harvard Business Review***, July-August: 100-109.

Arthur, W.B. 2000. Cognition: the black box of economics. In David Colander (ed.), ***The complexity vision and the teaching of economics***, Northampton, MA: Edward Elgar Publishing.

Astley, W.G. & Zammuto, R.F. 1992. Organization science, managers, and language games. ***Organization Science***, 3(4): 443-460.

Astley, W. G. & Van de Ven, A.H. 1983. Central perspectives and debates in organization theory, ***Administrative Sciences Quarterly***, 28: 245-273.

Atwater, J.B., Kannan, V.R. & Stephens, A.A. 2008. Cultivating systemic thinking in the next generation of business leaders. ***Academy of Management Learning & Education***, 7(1): 9-25.

Austin, R.D. & Nolan, R.L. 2000. ***IBM corporation turnaround***. Harvard Business School Case 9-600-098. Boston, MA: Harvard Business School Publishing.

Axelrod, R. & Cohen, M.D. 1999. ***Harnessing complexity***. New York, NY: Free Press.

Bacon, J. 2009. ***The art of community: Building the new age of participation***. Sebastopol, CA: O'Reilly.

Bailetti, A.J. 2007a. ***Five level model of company interactions with open source projects***. An invited presentation to the Q3 2007 meeting of the Eclipse Board of Directors, Chicago, IL, September 19. Slides available online: http://www.eclipse.org/org/foundation/boardminutes/2007_09_exhibits/ExhibitB.pdf

Bailetti, A.J. 2007b. ***Appropriate value (as in make money) from OS!*** Presented at the Eclipse Members Meeting, Chicago, IL, September 20. Slides available online: http://www.eclipse.org/org/foundation/membersminutes/20070920MembersMeeting/MakeMoney.pdf

Bailetti, A.J. 2009. ***Open source maturity curve and ecosystems***. Presented at Eclipse Summit Europe 2009, Ludwigsburg, Germany, Thursday October 29. Slides available online: http://www.eclipsecon.org/summiteurope2009/sessions?id=1045

Baker, W.E. & Faulkner, R.R. 2002. Interorganizational networks. In J.A.C. Baum (ed.), ***The Blackwell companion to organizations***, Malden, MA: Blackwell Publishers: 520-540.

Baldwin, C.Y. & Clark, K.B. 1997. Managing in an age of modularity. ***Harvard Business Review***, September-October: 84-93.

Baldwin, C.Y. & Clark, K.B. 2000. ***Design rules: Volume 1: The power of modularity***. Cambridge, MA: MIT Press.

Baldwin, C.Y. & Clark, K.B. 2006. The architecture of participation: does code architecture mitigate free riding in the open source development model? ***Management Science***, 52(7): 1116-1127.

Baldwin, C.Y., Hienerth, C. & von Hippel, E. 2006. How user innovations become commercial products: a theoretical investigation and case study. ***Research Policy***, 35(9): 1291-1313.

Baldwin, C.Y. & von Hippel, E. 2009. ***Modeling a paradigm shift: from producer innovation to user and open collaborative innovation***, Harvard Business School Working Paper No. 10-038.

Baldwin, C.Y. & Woodard, C.J. 2009. The architecture of platforms: a unified view. In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 19-44.

Barnet, M.L. 2006. Review. ***Academy of Management Perspectives***, 20(2): 88-90.

Barney, J. 1991. Firm resources and sustained competitive advantage. ***Journal of Management***, 17: 99-120.

Bazerman, M.H. 2005. Conducting influential research: the need for prescriptive implications. ***Academy of Management Review***, 30(1): 25-31.

Beaton, W. 2008. ***What is Eclipse?*** Presentation available online: http://www.eclipse.org/resources/resource.php?id=420

Beck, K. 1999. ***Extreme programming explained: Embrace change***. Reading, MA: Addison-Wesley.

Benkler, Y. 2002. Coase's penguin, or, Linux and *The Nature of the Firm*. ***Yale Law Journal***, 112(3): 369-446.

Berggren, C., Jarkvik, J. & Soderlund, J. 2008. Lagomizing, organic integration, and systems emergency wards: Innovative practices in managing complex systems development projects. ***Project Management Journal***, 39: S111-S122.

Bertalanffy, L. 1950. An outline of general systems theory. ***British Journal of Philosophy of Science***, 1(2): 139-164.

Bertalanffy, L. 1968. ***General system theory: Foundations, development, applications***. New York, NY: George Braziller.

Bettenburg, N., Just, S., Schroter, A., WeiB, C. Premraj, R., & Zimmerman, T. 2007. Quality of bug reports in Eclipse. ***Proceedings of the Workshop on Eclipse Technology eXchange***, OOPSLA (Object-oriented Programming, Systems, Languages, and Applications). Montreal, Canada.

Bi, Y., & Beidler, J. 2007. An environment for large software team projects. ***Consortium for Computing Sciences in Colleges***: 190-197.

Bierstedt, R. 1950. An analysis of social power. ***American Sociological Review***, 15: 730-738.

Blau, P.M. 1955. ***The dynamics of bureaucracy: A study of interpersonal relations in two government agencies***. Chicago, IL: University of Chicago Press.

Bollier, D. 2007. The growth of the commons paradigm. In C. Hess & E. Ostrom (eds.), ***Understanding knowledge as a commons: From theory to practice,*** Cambridge, MA: MIT Press: 27-40.

Boudreau, K.J. & Hagiu, A. 2009. Platform rules: multisided platforms as regulators. In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 163-191.

Boudreau, K.J. & Lakhani, K.R. 2009. How to manage outside innovation. ***MIT Sloan Management Review,*** Summer: 69-76.

Bovet, D. & Martha, J. 2000. ***Value nets: Breaking the supply chain to unlock hidden profits***. New York, NY: John Wiley & Sons.

Boyle, J. 2003. The second enclosure movement and the construction of the public domain. ***Law & Contemporary Problems***, 66(1&2): 33-74.

Brandenburger, A.M. & Nalebuff, B.J. 1996. ***Co-opeticion***. New York, NY: HarperCollins Business.

Bresnahan, T.F. & Greenstein, S. 1999. Technological competition and the structure of the computer industry. ***Journal of Industrial Economics***, 47(1): 1-40.

Bricklin, D. 2000. ***The cornucopia of the commons: How to get volunteer labor***. http://www.bricklin.com/cornucopia.htm

Brint, S. 2001. Gemeinschaft revisited: a critique and reconstruction of the community concept. ***Sociological Inquiry***, 19(1): 1-23.

Brooks, F.P. 1995. ***The mythical man-month: Essays on software engineering*** (anniversary edition). Upper Saddle River, NJ: Addison-Wesley.

Brown, J.S. & Duguid, P. 1991. Organizational learning and communities-of-practice: toward a unified view of working, learning, and innovation. ***Organization Science***, 2(1): 40-57.

Brown, M.T. 2004. A picture is worth a thousand words: energy systems language and simulation. ***Ecosystem Modelling***, 178: 83-100.

Brown, S. & Eisenhardt, K. 1997. The art of continuous change: linking complexity theory and time-paced evolution in relentlessly shifting organizations. ***Administrative Science Quarterly***, 42: 1-34.

Brown, S. & Eisenhardt, K. 1998. ***Competing on the edge: strategy as structured chaos.*** Boston, MA: Harvard Business School Press.

Byman, A., Bell, E., Mills, A.J. & Yue, A.R. 2011. ***Business research methods*** (Canadian edition). Don Mills, ON: Oxford University Press.

Burns, T. & Stalker, G.M. 1961. ***The management of innovation***. London, UK: Tavistock.

Burt, R.S. 1980. Models of network structure. ***Annual Review of Sociology***, 6: 79-141.

Burt, R.S. 1992. ***Structural holes***. Cambridge, MA: Harvard University Press.

California Management Review. 2001. Business as a living system: the value of industrial ecology. 43(3): 16-25.

Callebaut, W. 2005. The ubiquity of modularity. In W. Callebaut and D. Rasskin-Gutman (eds.) (2005), ***Modularity: Understanding the development and evolution of natural complex systems***, Cambridge MA: MIT Press: 3-28.

Campbell, J. 2007. ***Clarifying the IP trail***. Presented at Delivering Clean Software, an event in the OCRI Partnership Conference Series, Ottawa, Canada, May 29. Slides available online: http://www.smartcapital.ca/events/presentations/partnership/may2907/campbell.pdf

Campbell, J. 2009. ***IP for Eclipse committers***. Presented at EclipseCon 2009, Santa Clara, CA, Thursday March 26. Session recording is available online: http://live.eclipse.org/node/713

Capek, P.G., Frank, S.P., Gerdt, S., & Shields, D. 2005. A history of IBM's open source involvement and strategy. ***IBM Systems Journal***, 44(2): 249-257.

Carbone, P. 2006. ***Competing with open source software: Insights from recent research***. Presented at Competing with Open Source Software, an event in the OCRI Partnership Conference Series, Ottawa, Canada, Friday April 21. Slides available online: http://smartcapital.ca/events/presentations/partnership/April2106/carbone.pdf

Carbone, P. 2007a. ***Value derived from open source is a function of maturity levels.*** Presented at The Alchemy of Open Source Business, an event in the OCRI Partnership Conference Series, Ottawa Canada, April 19. Slides available online: http://www.smartcapital.ca/events/presentations/partnership/April1907/PeterCarbone.pdf

Carbone, P. 2007b. Competitive open source. ***Open Source Business Resource (OSBR)***, July: 4-6.

Carlile, P.R. & Christensen, C.M. 2005. ***The cycles of theory building in management research*** (version 6.0), January 6 2005. http://www.innosight.com.

Carlson, D. 2005. ***Eclipse distilled.*** Upper Saddle River, NJ: Addison-Wesley.

Castells, M. 2000. ***The rise of the network society*** (second edition). Malden, MA: Blackwell Publishing.

Chandler, A.D. 1962. ***Strategy and structure: Chapters in the history of the American industrial enterprise.*** Cambridge, MA: MIT Press.

Chandler, A.D. 1977. ***The visible hand: The managerial revolution in American business.*** Cambridge, MA: Harvard University Press.

Chandler, A.D. 1982. The M-form: industrial groups, American style. ***European Economic Review,*** 19: 3-23.

Chandler, A.D. 1990. ***Scale and scope: The dynamics of industrial capitalism.*** Cambridge, MA: Belknap Press.

Chesbrough, H.W. 2003a. ***Open innovation: The new imperative for creating and profiting from technology.*** Boston, MA: Harvard Business School Press.

Chesbrough, H.W. 2003b. The era of open innovation. ***Sloan Management Review***, 44(3): 35-41.

Chesbrough, H.W. 2006. ***Open business models: How to thrive in the new innovation landscape***, Boston, MA: Harvard Business School Press.

Chesbrough, H.W. & Appleyard, M.M. 2007. Open innovation and strategy. ***California Management Review,*** 50(1): 57-76.

Chesbrough, H.W. & Rosenbloom, R.S. 2002. The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies. ***Industrial and Corporate Change***, 11(3): 529-555.

Chesbrough, H.W., Vanhaverbeke, W. & West, J. (eds.). 2006. ***Open innovation: Researching a new paradigm.*** Oxford, UK: Oxford University Press.

Christensen, C.M. 1997. ***The innovator's dilemma: When new technologies cause great firms to fail***, Boston, MA: Harvard Business School Press.

Christensen, C.M. 2006. The ongoing process of building a theory of disruption. ***Journal of Product Innovation Management***, 23(1): 39-55.

Christensen, C.M. & Carlile, P.R. 2009. Course research: using the case method to build and teach management theory. ***Academy of Management Learning and Education***, 8(2): 240-251.

Christensen, C.M. & Raynor, M.E. 2003a. ***The innovator's solution: Creating and sustaining successful growth***. Boston, MA: Harvard Business School Press.

Christensen, C.M. & Raynor, M.E. 2003b. Why hard-nosed executives should care about management theory. ***Harvard Business Review***, September: 66-74.

Christensen, C.M., Raynor, M. & Verlinden M. 2001. Skate to where the money will be. ***Harvard Business Review***, 79(10): 72-81.

Christensen, C.M. & Rosenbloom, R.S. 1995. Explaining the attacker's advantage: the technological paradigms, organizational dynamics, and the value network. ***Research Policy***, 24: 233-257.

Clenahan, P. 2009. ***Open source as part of a successful product strategy***. Presented at EclipseCon 2009, Santa Clara, CA, Tuesday March 24.

Cloninger, E. 2010. ***Eclipse for mobile application development panel***. A panel session at EclipseCon 2010, Santa Clara, CA, Monday March 22.

Corallo, A., Passiante, G. & Prencipe, A. 2007. ***The digital business ecosystem***. Cheltenham, UK: Edward Elgar Publishing.

Crowston, K. & Howison, J. 2003. ***The social structure of open source software development teams***. Presented at the IFIP 8.2 Working Group on Information Systems in Organizations Organizations and Society in Information Systems (OASIS) Workshop, Seattle, WA, December 14.

Culpan, R. 2009. A fresh look at strategic alliances: research issues and future directions. ***International Journal of Strategic Business Alliances***, 1(1): 4-23.

Cusumano, M.A. 2004. ***The business of software: What every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad***. New York, NY: Free Press.

Cusumano, M.A. 2010. ***Staying power: Six enduring principles for managing strategy & innovation in an uncertain world***. Oxford, UK: Oxford University Press.

Cusumano, M.A. & Gawer, A. 2002. The elements of platform leadership. ***MIT Sloan Management Review***, 43(3): 51-58.

Cyert, R.M. & March, J.G. 1963. ***A behavioral theory of the firm***. Englewood Cliffs, NJ: Prentice-Hall.

Dahl, R.A. 1957. The concept of power. ***Behavioral Science***, 2: 201-215.

Dahlander, L. 2007. Penguin in a new suit: a tale of how de novo entrants emerged to harness free and open source software communities. ***Industrial and Corporate Change***, 16(5): 913-943.

Dahlander, L. & Gann, D.M. 2010. How open is innovation? ***Research Policy***, 39: 699-709.

Dahlander, L. & Magnusson, M.G. 2005. Relationships between open source software companies and communities: observations from Nordic firms. ***Research Policy***, 34: 481-493.

Dahlander, L. & Magnusson, M.G. 2008. How do firms make use of open source communities? ***Long Range Planning***, 41: 629-649.

Dahlander, L. & Wallin, M.W. 2006. A man on the inside: Unlocking communities as complementary assets. ***Research Policy***, 35(8): 1243-1259.

Davis, J.P., Eisenhardt, K.M. and Bingham, C. B. 2007. Developing theory through simulation methods. ***Academy of Management Review***, 32(2): 480-499.

de Geus, A. 1997. ***The living company: Growth, learning and longevity in business***. Boston, MA: Harvard Business School Press.

de Geus, A. 2008. Learning together for good decision making. ***Reflections***, 8(1): 28-35.

des Rivières, J., & Wiegand, J. 2004. Eclipse: a platform for integrating development tools. ***IBM Systems Journal***, 43(2): 371-383.

DiBona, C., Ockman, S. & Stone, M. (Eds.). 1999. ***Open sources: voices from the open source revolution***, Sebastopol, CA: O'Reilly Media.

Dietz, T., Ostrom, E. & Stern, P.C. 2003. The struggle to govern the commons. ***Science***, 302: 1907-1912.

DiMaggio, P.J. & Powell, W.W. 1983. The iron cage revisited: institutional isomorphism and collective rationality in organizational fields. ***American Sociological Review***, 48: 147-160.

Dueñas, J.C, Parada, G.H.A., Cuadrado, F., Santillán, M, & Ruiz, J.L. 2007. Apache and Eclipse: comparing open source project incubators. ***IEEE Software***, 24(6): 90-98.

Eclipse Consortium press release. November 29 2001. ***Eclipse.org Consortium forms to deliver new era application development tools***. Available online: http://www.eclipse.org/org/pr.html

Eclipse Consortium press release. December 16 2002. ***Momentum builds as Eclipse eco-system grows***. Available online: http://www.eclipse.org/org/press-release/dec162002oneyearpr.html

Eclipse Consortium press release. April 15 2003. ***Eclipse platform R2.1 now available.*** Available online: http://www.eclipse.org/org/press-release/apr15200321pr.html

Eclipse Consortium press release. November 25 2003. ***Eclipse celebrates second anniversary***. Available online: http://www.eclipse.org/org/press-release/nov252003twoyearpr.html

Eclipse Foundation press release. February 2 2004. ***Eclipse Forms Independent Organization***. Available online: http://www.eclipse.org/org/press-release/feb2004foundationpr.php

Eclipse Foundation press release. March 3 2004. ***Eclipse names founding board of directors***. Available online: http://www.eclipse.org/org/press-release/mar32004directorspr.html

Eclipse Foundation press release. June 1 2004. ***Mike Milinkovich named Eclipse's executive director***. Available online: http://www.eclipse.org/org/press-release/jun12004edpr.html

Eclipse Foundation press release. September 9 2008. ***New global Eclipse Training Series launch in 31 cities***. Available online: http://www.eclipse.org/org/press-release/20080909_training.php

Eclipse Foundation press release. June 23 2010. ***Eclipse Helios annual release train has arrived***. Available online: http://www.eclipse.org/org/press-release/20100623_heliosrelease.php

Eisenhardt, K.M. 1989. Building theories from case study research. ***Academy of Management Review***, 14(4): 532-550.

Eisenhardt, K.M., Furr, N.R. & Bingham, C.B. 2010. Microfoundations of performance: balancing efficiency and flexibility in dynamic environments. ***Organization Science***, 21(6): 1263-1273.

Eisenhardt, K.M. & Graebner, M.E. 2007. Theory building from cases: opportunities and challenges. ***Academy of Management Journal***, 50(1): 25-32.

Eisenmann, T.R. 2008. Managing proprietary and shared platforms. ***California Management Review***, 50(4): 31-53.

Eisenmann, T.R., Parker, G.G. & Van Alstyne, M.W. 2006. Strategies for two-sided markets. ***Harvard Business Review***, October: 92-101.

Eisenmann, T.R., Parker, G.G. & Van Alstyne, M.W. 2009. Opening platforms: how, when, and why? In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 131-162.

Enayat, H., Muegge, S. & Tanev, S. 2009. Impact of diversity on open source software. ***Proceedings of the 4th International MCETECH Conference on e-Technologies*** (MCETECH 2009), May 4-6, Ottawa, Canada.

Eppinger, S.D. 1991. Model-based approaches to managing concurrent engineering. ***Journal of Engineering Design***, 2(4): 238-290.

Eppinger, S.D., Whitney, D.E., Smith, R.P., & Gebala, D.A. 1994. A model-based method for organizing tasks in product development. ***Research in Engineering Design***, 6(1): 1-13.

Evans, D.S. 2009. How catalysts ignite: the economics of platform-based start-ups. In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 99-128.

Feller, J., Finnegan, P. & Hayes, J. 2008. Delivering the whole product: business model impacts and agility challenges in a network of open source firms. ***Journal of Database Management***, 19(2): 95-108.

Feller, J., Fitzgerald, B., Hissam, S.A, & Lakhani, K.R. (eds.). 2005. ***Perspectives on free and open source software***. Boston, MA: MIT Press.

Fichter, K. 2009. Innovation communities: the role of networks of promoters in open innovation. ***R&D Management***, 39(4): 357-371.

Fink, M. 2003. ***The business and economics of Linux and open source***, Upper Saddle River, NJ: Prentice Hall.

Fiona, M. & O'Mahony, S. 2007. Exploring the foundations of cumulative innovation: implications for organization science. ***Organization Science***, 18(6): 1006-1021.

Fischer, J., Lindenmayer, D.B. & Manning, A.D. 2006. Biodiversity, ecosystem function, and resilience: ten guiding principles for commodity production landscapes. ***Frontiers in Ecology and the Environment***, 4(2): 80-86.

Fitzgerald, B. 2006. The transformation of open source software. ***MIS Quarterly***, 30(3): 587-598.

Foddy, W. 1993. ***Constructing questions for interviews and questionnaires: theory and practice in social research***. Cambridge: Cambridge University Press.

Fogel, K. 2006. ***Producing open source software: How to run a successful free software project.*** Sebastopol, CA: O'Reilly.

Folk, C., Carpenter, S., Walker, B., Scheffer, M., Elmqvist, T., Gunderson, L, & Holling, C.S. 2004. Regime shifts, resilience, and biodiversity in ecosystem management. ***Annual Review of Ecology, Evolution, and Systematics***, 35: 557-581.

Folk, C., Carpenter, S., Walker, B., Scheffer, M., Elmqvist, T., Gunderson, L, & Holling, C.S. 2006. Resilience: The emergence of a perspective for social–ecological systems analyses. ***Global Environmental Change***, 16(3): 253-267.

Forrester, J.W. 1971. The counterintuitive behavior of social systems. ***Technology Review***, 73(3): 52-68.

Franke, N. & Shah, S. 2003. How communities support innovative activities: an exploration of assistance and sharing among end-users. ***Research Policy***, 32(1): 157-178.

Free Software Foundation. 1991. ***GNU General Public License*** (version 2). Available online: http://www.gnu.org/licenses/gpl-2.0.html

Free Software Foundation. 2010. ***Some confusing or loaded words and phrases to avoid (or use with care).*** Available online (accessed April 15 2010): http://www.gnu.org/philosophy/words-to-avoid.html#Ecosystem

Freeman, C. 1982. ***The economics of industrial innovation*** (2nd edition). Cambridge, MA: MIT Press.

French Jr., J.R.P. & Raven, B. 1959. The bases of social power. In D.P. Cartwright (ed.), ***Studies in social power***, Ann Arbor, MI: Institute for Social research, University of Michigan: 150-167.

Frost, R. 2007. Jazz and the Eclipse way of collaboration. ***IEEE Software***, 24(6): 114-117.

Gamma, E. 2008. ***How (7 years of) Eclipse changed my views on software development.*** Presented at QCon, London, UK, February 7. Video and slides available online: http://www.infoq.com/presentations/Eclipse-Lessons-Erich-Gamma

Gamma, E. & Beck, K. 2003. ***Contributing to Eclipse: Principles, patterns, and plug-ins.*** Boston, MA: Addison-Wesley.

Gamma, E. & Slobojan, R. 2008. Erich Gamma discusses Jazz, Eclipse, JUnit and design patterns. ***InfoQ***, September 16. Available online: http://www.infoq.com/interviews/gamma-jazz-eclipse-junit-design-patterns

Gamma, E. & Wiegand, J. 2005. ***The Eclipse way: Processes that adapt***. Presented at EclipseCon 2005, Burlingame, CA, Tuesday March 1. Slides available online: http://eclipsecon.org/2005/presentations/econ2005-eclipse-way.pdf

Gans, J.S., Hsu, D.H. & Stern, S. 2002. When does start-up innovation spur the gale of creative destruction? ***RAND Journal of Economics***, 33(4): 571-586.

Gans, J.S., Hsu, D.H. & Stern, S. 2008. The impact of uncertain intellectual property rights on the market for ideas: evidence from patent grant delays. ***Management Science***, 54(5): 982-997.

Gans, J.S. & Stern, S. 2003. The product market and the market for "ideas": commercialization strategies for technology entrepreneurs. ***Research Policy***, 32: 333-350.

Gardler, R. & Hanganu, G. 2011. ***Meritocratic governance model***. OSS Watch. http://www.oss-watch.ac.uk/resources/meritocraticGovernanceModel.xml.

Garud, R., Kumaraswamy, A. & Langlois, R.N. (eds.). 2003. ***Managing in a modular age: Architectures, networks, and organizations***. Malden, MA: Blackwell Publishing.

Gawer, A. (ed.). 2009a. ***Platforms, markets and innovation***. Northampton, MA: Edward Elgar.

Gawer, A. 2009b. Platform dynamics and strategies: from products to services. In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 45-76.

Gawer, A. & Cusumano, M. 2002. ***Platform leadership: How Intel, Microsoft and Cisco drive industry innovation***. Boston, MA: Harvard Business School Press.

Gawer, A. & Cusumano, M. 2008. How companies become platform leaders. ***MIT Sloan Management Review***, 49(2): 27-35.

Gawer, A., & Henderson, R. 2007. Platform owner entry and innovation in complementary markets: evidence from Intel. ***Journal of Economics & Management Strategy***, 16(1): 1-34.

Gay, J. 2002. ***Free software free society: Selected essays of Richard M. Stallman***. Boston, MA: Soho Books.

Geelan, J. 2008. Exclusive Q&A with Mike Milinkovich, Eclipse Foundation. ***Java Developers Journal***, July 14. Available online: http://java.sys-con.com/node/609086

Geer, D. 2005. Eclipse becomes the dominant Java IDE. ***Computer***, July: 16-18.

Gerstner, L.V. 2002. ***Who says elephants can't dance?*** New York: NY: HarperCollins Publishers Inc.

Ghosh, R.A., Glott, R., Kreiger, B. & Robles-Martinez, G. 2002. ***The free/libre and open source software developers survey and study: FLOSS deliverable D18: Final report, part 4: Survey of developers.*** University of Maastricht, The Netherlands: International Institute of Infonomics. Available online: http://www.flossproject.org/report/FLOSS_Final4.pdf

Ghosh, R.A. 2005. Understanding free software developers: findings from the FLOSS study. In J. Feller, B. Fitzgerald, S.A. Hissam & K.R. Lakhani (eds.), ***Perspectives on free and open source software***, Cambridge, MA: The MIT Press: 23-45.

Gibbert, M., Winfried, R. & Wicki, B. 2008. What passes as a rigorous case study? ***Strategic Management Journal***, 29: 1465-1474.

Gibson, C.C. 2005. In pursuit of better policy outcomes. ***Journal of Economic Behavior & Organization***, 57(2): 227-230.

Glaser, B. 1978. ***Theoretical sensitivity: Advances in the methodology of grounded theory***. Mill Valley, CA: Sociology Press.

Glaser, B. 1998. ***Doing grounded theory: Issues and discussions***. Mill Valley, CA: Sociology Press.

Glaser, B. & Strauss, A. 1967. ***The discovery of grounded theory: Strategies for qualitative research***. Berlin: Germany: Aldine de Gruyter.

Golley, F. 1993. ***A history of the ecosystem concept in ecology: More than the sum of the parts***. New Haven, CT: Yale University Press.

Goth, G. 2005. Beware the march of this IDE: Eclipse is overshadowing other tool technologies. ***IEEE Software***, 22(4): 108-111.

Goth, G. 2009. The task-based interface: not your father's desktop. ***IEEE Software***, 26(6): 88-91.

Goyal, S. 2007. ***Connections: An introduction to the economics of networks***. Princeton, NJ: Princeton University Press.

Grand, S., von Drough, G. Leonard, D. & Swap, W. 2004. Resource allocation beyond firm boundaries: a multi-level model for open source innovation. ***Long Range Planning***, 37: 591-610.

Granovetter, M.S. 1973. The strength of weak ties. ***American Journal of Sociology***, 78(6): 1360-1380.

Granovetter, M.S. 1994. Business groups. In N.J. Smelser & R. Swedberg (eds.), ***The Handbook of Economic Sociology***. Princeton, NJ: Princeton University Press and Russell Sage Foundation: 453-475.

Grant, D, & Oswick, C. (eds.). 1996. ***Metaphor and Organization***. London, UK: Sage.

Gruber, O., Hargrave, B.J., McAffer, J., Rapicault, P. & Watson, T. 2005. The Eclipse 3.0 platform: adopting OSGi technology. ***IBM Systems Journal***, 44(2): 289-299.

Guba, E.G. & Lincoln, Y.S. 1994. Computing paradigms in qualitative research. In N.K. Denzin, N.K. & Y.S. Lincoln (eds.), ***Handbook of qualitative research***. Thosand Oaks, CA: Sage: 105-117.

Gunderson, L.H. 2000. Ecological resilience – in theory and application. ***Annual Review of Ecology and Systematics***, 31: 425-439, Nov.

Hagiu, A. 2007. Merchant or two-sided platform? ***Review of Network Economics***, 6(2): 115-133.

Hagiu, A. 2009. Two-sided platforms: product variety and pricing structures. ***Journal of Economics & Management Strategy***, 18(4): 1011-1043.

Hannan, M.T. & Freeman, J. 1977. The population ecology of organizations. ***American Journal of Sociology***, 82(5): 929-964.

Hannan, M.T., Polos, L. & Carroll, G.R. 2007. ***Logics of organization theory: Audiences, code, and ecologies***. Princeton: Princeton University Press.

Hardin, G. 1968. The tragedy of the commons. ***Science***, 162:1243-1248.

Hawley, A. 1968. ***Human Ecology: A Theoretical Essay***. Chicago, IL: University of Chicago Press.

Henderson, R.M. & Clark, K.B. 1990. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. ***Administrative Science Quarterly***, 35: 9-30.

Henkel, J. 2006. Selective revealing in open innovation processes: the case of embedded Linux. ***Research Policy***, 35(7): 953-969.

Herraiz, I., Robles, G., Amor, J.J., Romera, T. & Gonzalez Barahona, J.M. 2006. The processes of joining in global distributed software projects. In P. Kruchten, D. Moitra, W. Strigel & C. Ebert (eds.), ***Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner***. New York, NY: ACM: 27-33.

Hertel, G., Niedner, S. & Herrmann, S. 2003. Motivation of software developers in open source projects: an Internet-based survey of contributors to the Linux kernel. ***Research Policy***, 32(7): 1159-177.

Hess, C. & Ostrom, E. 2007. ***Understanding knowledge as a commons: From theory to practice***. Cambridge, MA: MIT Press.

Hitt, M.A., Beamish, P.W., Jackson, S.E. & Mathieu, J.E. 2007. Building theoretical and empirical bridges across levels: multilevel research in management. ***Academy of Management Journal***. 50(6): 1385-1399.

Hoey, A.S. & Bellwood, D.R. 2009. Limited functional redundancy in a high diversity system: Single species dominates key ecological process on coral reefs. ***Ecosystems***, 12(8): 1316-1328.

Hofmokl, J. 2010. Towards an eclectic theory of the internet commons. ***International Journal of the Commons***, 4(1): 226-250.

Holland, J.H. 1992. ***Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*** (2nd edition). Ann Arbor, MI: University of Michigan Press.

Holling, C.S. 1973. Resilience and Stability of Ecological Systems. ***Annual Review of Ecology and Systematics***, 4: 1-23.

Hultink, E.J., Griffin, A., Hart, S. & Robben, H.S. 1997. Industrial new product launch strategies and product development performance. ***Journal of Product Innovation Management***, 14(4): 243- 257.

Iansiti, M & Levien, R. 2004a. ***The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation, and sustainability.*** Boston, MA: Harvard Business Press.

Iansiti, M. & Levien, R. 2004b. Strategy as ecology. ***Harvard Business Review***, 82(3): 68-78.

Iansiti, M. & Levien, R. 2004c. Strategy as ecology. ***Harvard Business Review***, 82(9): 132-133.

Iansiti, M. & Richards, G.L. 2006. The information technology ecosystem: structure, health, and performance. ***The Antitrust Bulletin***, 51(1): 77- 110.

IBM press release. November 5 2001. ***IBM donates $40 million of software to open source community***. http://www-03.ibm.com/press/us/en/pressrelease/1025.wss

Isckia, T. 2009. Amazon's evolving ecosystem: a cyber-bookstore and application service provider. ***Canadian Journal of Administrative Sciences***, 26: 332-343.

Ives, A.R. & Carpenter, S.R. 2007. Stability and Diversity of Ecosystems. ***Science***, 317(5834): 58 – 62.

Iyer, B. & Davenport, T.H. 2008. Reverse engineering Google's innovation machine. ***Harvard Business Review***, 86(4): 59-68.

Iyer, B., Lee, C. & Venkatraman, N. 2006. Managing in a "small world ecosystem": Lessons from the software sector. ***California Management Review***. 48(3), 27-47.

Jarvenpaa, S.L. & Majchrzak, A. 2008. Knowledge collaboration among professionals protecting national security: roles of transactive memories in ego-centered knowledge networks. ***Organization Science***, 19(2): 260-276.

Jick, T.D. 1979. Mixing qualitative and quantitative methods: Triangulation in action. ***Administrative Science Quarterly***, 24: 602-11.

Johnson, M.W. 2010. ***Seizing the whitespace: Business model innovation for growth and renewal***. Boston, MA: Harvard Business Press.

Johnson, M.W., Christensen, C.M. & Kagermann, H. 2008. Reinventing your business model. ***Harvard Business Review***, December: 50-56.

Kao, J. 2009. Tapping the world's innovation hot spots. ***Harvard Business Review***, March: 109-114.

Katz, M.L. & Shapiro, C. 1985. Network externalities, competition, and compatibility. ***American Economic Review***, 75: 424-440.

Kersten, M. 2007. ***Focusing knowledge work with task context***. Unpublished Ph.D. Thesis. University of British Columbia.

Kidane, Y.H., & Gloor, P.A. 2007. Correlating temporal communication patterns of the Eclipse open source community with performance and creativity. ***Computational & Mathematical Organization Theory***, 13: 17-27.

Kilamo, T., Aaltonen, T. & Heinimäki, T.J. 2010. BULB: Onion-based measuring of OSS communities. ***IFIP Advances in Information and Communication Technology***, 319: 342-347.

Kim, D.H. & Senge, P.M. 1994. Putting systems thinking into practice. ***System dynamics review***, 10 (2-3): 277-290.

Kiser, L.L. & Ostrom, E. 1982. The three worlds of action: a metatheoretical synthesis of institutional approaches. In E. Ostrom (ed.), ***Strategies of Political Inquiry***. Beverly Hills, CA: Sage.

Konigsberg, R. 2010. ***Eclipse in the enterprise: lessons from Google***. Presented at EclipseCon 2010, Santa Clara, CA, Tuesday March 23.

Kranich, N. 2007. Countering enclosure: reclaiming the knowledge commons. In C. Hess & E. Ostrom (eds.), ***Understanding knowledge as a commons: From theory to practice,*** Cambridge, MA: MIT Press: 85-122.

Krause, J. & Schmidt, K. 2010. ***Long term support for Eclipse – the next step in enterprise readiness***. Presented at Eclipse Summit Europe 2010, Ludwigsburg, Germany, Thursday November 4.

Lakhani, K.R. & von Hippel, E. 2003. How open source software works: "free" user-to-user assistance. ***Research Policy***, 32(6): 923-943.

Lakhani, K.R. & Wolf, R.G. 2005. Why hackers do what they do: understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S.A. Hissam & K.R. Lakhani (eds.), ***Perspectives on free and open source software***, Cambridge, MA: The MIT Press: 4-21.

Lakoff, G. & Johnson, M. 1980. ***Metaphors we live by***. Chicago, IL: The University of Chicago Press.

LaMantia, M.J., Cai Y., MacCormack A., Rusnak J. 2008. Analyzing the evolution of large-scale software systems using design structure matrices and design rule theory: two exploratory cases. ***Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture*** (WICSA 2008), Vancouver Canada, February 18-22.

Langlois, R.L. & Robertson, P.L. 1992. Networks and innovation in a modular system: lessons from the microcomputer and stereo components industries. ***Research Policy***, 21: 297-313.

Large, D. & Muegge, S. 2008. Venture capitalists' non-financial value-added: an evaluation of the evidence and implications for research. ***Venture Capital: An International Journal of Entrepreneurial Finance***, 10(1): 21-53.

Lawton, M. 2007. ***Eclipse community survey results***. IDC. Available online: http://cdn.idc.com/downloads/EclipseCommunitSurveyResults%5BNotesPages%5D.pdf

Lenfle, S. & Baldwin, C.Y. 2007. ***From manufacturing to design: An essay on the work of Kim B. Clark***. HBS Working Paper 07-057, March.

Lerner, J. & Schankermann, M. 2010. ***The comingled code: Open source and economic development***. Cambridge, MA: MIT.

Lerner, J. & Tirole, J. 2002. Some simple economics of open source. ***Journal of Industrial Economics***, 52: 197-234.

Lessig, L. 2000. ***Code and other laws of cyberspace***. New York, NY: Basic Books.

Leonard-Barton, D. 1990. A dual methodology for case studies: synergistic use of a longitudinal single site with replicated multiple sites. ***Organization Science***, 1(3): 248-266.

Lerner, J & Tirole, J. 2002. Some simple economics of open source. ***Journal of Industrial Economics***, 52: 197-234.

Levin, S.A. & Lubchenco, J. 2008. Resilience, robustness, and marine ecosystem-based management. ***BioScience***, 58(1): 27–32.

Levine, R. Searls, D., Weinberger, D. & Lock, C. 1999. ***The cluetrain manifesto: the end of business as usual.*** New York, NY: Perseus Books.

Levy, S. 1984. ***Hackers: Heroes of the computer revolution.*** New York, NY: Penguin Books.

Li, Y. 2009. The technological roadmap of Cisco's business ecosystem. ***Technovation***, 29: 379-386.

Lipset, S.M., Trow, M.A. & Coleman, J.S. 1956. ***Union democracy: The internal politics of the International Typographical Union.*** New York, NY: Free Press.

Locke, E. 2007. The case for inductive theory building. ***Journal of Management***, 33(6): 867-890.

Lynn, L.H., Reddy, N.M. & Aram, J.D. 1996. Linking technology and institutions: the innovation community framework. ***Research Policy***, 25(1): 91-106.

MacCormack, A., Rusnak, J., & Baldwin, C.Y. 2006. Exploring the structure of complex software designs: an empirical study of open source and proprietary code. ***Management Science***, 52(7), 1015-1030.

MacKechnie, G. & Donnelly-Cox, G. 1996. Metaphor in the Development of Organization Theory. In D. Grant & C. Oswick (eds.), ***Metaphor and organization***, London, UK: Sage.

Makadok, R. & Coff, R. 2009. Both market and hierarchy: an incentive-system theory of hybrid governance forms. ***Academy of Management Review***, 34(2): 297-319.

March, J.G., Sproull, L.S. & Tamuz, M. 1991. Learning from samples of one or fewer. ***Organization Science***, 2(1): 1- 13.

Markus, M.L. 2007. The governance of free/open source software projects: monolithic, multidimensional, or configuration? ***Journal of Management and Governance***, 11(2): 151-163.

Marple, D. 1961. The decisions of engineering design. ***IEEE Transactions of Engineering Management***, 2: 55-71.

Mateos-Garcia, J. & Steinmueller, W.E. 2008. The institutions of open source software: examining the Debian community. ***Information Economics and Policy***, 20: 333-344.

Maula, M., Keil, T. & Salmenkaita, J. 2006. Open innovation in systematic innovation contexts. In H. Chesbrough, W. Vanhaverbeke & J. West (eds.), ***Open innovation: Researching a new paradigm***, Oxford, UK: Oxford University Press: 241-257.

McAffer, J., Lemieux, J. & Aniszczyk, C. 2010a. ***Eclipse rich client platform*** (second edition). Upper Saddle River, NJ: Addison-Wesley.

McAffer, J., VanderLie, P. & Archer, S. 2010b. ***OSGI and Equinox: Creating highly modular Java systems***. Upper Saddle River, NJ: Addison-Wesley.

McGovern, G., Cout, D., Quelch, J.A., & Crawford, B. 2004. Bringing customers into the boardroom. ***Harvard Business Review***, November: 70-80.

McKelvey, B. 2002. Model-centered organization science epistemology. In J. A. C. Baum (ed.), ***The Blackwell companion to organizations***. Oxford, UK: Blackwell: 752-780.

Messerschmitt, D.G. & Szyperski, C. 2003. ***Software ecosystem : Understanding an indispensable technology and industry***. Cambridge, MA: MIT Press.

Meyer, J.W. & Rowan, B. 1977. Institutional organizations: formal structure as myth and ceremony. ***American Journal of Sociology***, 83: 340-363.

Meyer, M.H. & Lehnerd, A.P. 1997. ***The power of product platforms: Building value and cost leadership***. New York, NY: Free Press.

Miles, M. & Huberman, M. 1994. ***Qualitative data analysis: An expanded sourcebook*** (2nd Edition). Newbury Park, CA: Sage Publications.

Miles, R.E., Miles, G. & Snow, C.C. 2005. ***Collaborative entrepreneurship: How communities of networked firms use continuous innovation to create economic wealth***. Palo Alto, CA: Stanford University Press.

Milev, R., Muegge, S. & Weiss, M. 2009. Design evolution of an open source software project using an improved modularity metric. ***Proceedings of the 5th International Conference on Open Source Systems*** (OSS 2009).

Milgram, S. 1967. The small world problem. ***Psychology Today***, 2: 60-67.

Milinkovich, M. 2006. ***Open source: power to the technology consumer***. Presented at Competing with Open Source Software, an event in the OCRI Partnership Conference Series, Ottawa, Canada, Friday April 21. Slides available online: http://www.ocri.ca/events/presentations/partnership/April2106/Milinkovich.pdf

Milinkovich, M. 2007. ***The Eclipse open innovation network***. Presented at The Alchemy of Open Source Business, an event in the OCRI Partnership Conference Series, Ottawa Canada, April 19. Slides available online: http://www.ocri.ca/events/presentations/partnership/April1907/MikeMilinkovich.pdf

Milinkovich, M. 2008a. ***A practitioner's guide to ecosystem development***. Presented at the Technology Innovation Management (TIM) Lecture Series, Carleton University, Ottawa Canada, September 3. Slides available online: http://www.talentfirstnetwork.org/wiki/images/5/58/A_practitioners_guide_to_ecosystem_development_Sep_3.pdf

Milinkovich, M. 2008b. TIM Lectures: ecosystem development. ***Open Source Business Resource (OSBR)***, October: 40-42.

Milinkovich, M. 2008c. ***A practitioner's guide to software ecosystems***. Online webinar presented October 15. Video and slides available online at *Eclipse Live*: http://live.eclipse.org/node/587.

Milinkovich, M. 2010. Q&A: How can a community be considered "open source" if its primary objective is to promote commercialization? ***Open Source Business Resource (OSBR)***, January: 29-30.

Miller, J.H. & Page, S.E. 2007. ***Complex adaptive systems: An introduction to computational models of social life***. Princeton, NJ: Princeton University Press.

Mintzberg, H. 1987. The strategy concept I: Five Ps for strategy. ***California Management Review***, 30(1): 11-24.

Mitchell, W. 1991. Dual clocks: Entry order influences on industry incumbent and newcomer marketshare and survival when specialized assets retain their value. ***Strategic Management Journal***, 12(2): 85-100.

Moore, G.A. 1991. ***Crossing the chasm: Marketing and selling high-tech products to mainstream customers***. New York, NY: Harper-Business.

Moore, G.A. 2005. ***Dealing with Darwin: How great companies innovate at every phase of their evolution***. New York, NY: The Penguin Group.

Moore, J.F. 1993. Predators and prey: a new ecology of competition. ***Harvard Business Review***, May/June: 75-86.

Moore, J.F. 1996. ***The death of competition***. New York, NY: HarperBusiness.

Moore, J.F. 1998. The rise of a new corporate form. ***Washington Quarterly***. 21(1): 167-181.

Moore, J.F. 2006. Business ecosystems and the view from the firm. ***The Antitrust Bulletin***, 51(1): 31-75.

Morgan, G. 1980. Paradigms, metaphors, and puzzle solving in organization theory. ***Administrative Science Quarterly***, 25(4): 605.

Morgan, G. 1996. Is there anything more to be said about metaphor? In D. Grant & C. Oswick (eds.), ***Metaphor and organization***, London, UK: Sage.

Morgan, G. 2006. ***Images of organization*** (third edition). Thousand Oaks, CA: Sage.

Morgan, M.S. & Morrison, M. 1999. ***Models as mediators: Perspectives on natural and social science***. Cambridge, UK: Cambridge University Press.

Moriconi, M., Quain, X., Riemenschneider, R.A. & Gong, L. 1997. Secure software architectures. ***Proceedings of the IEEE Symposium on Security and Privacy***, May 4-7, Oakland, CA.

Murphy, G.C., Kersten, M., & Findlater, L. 2006. How are Java software developers using the Eclipse IDE? ***IEEE Software***, July/August: 76-83.

Muegge, S.M. 2010. ***Thesis proposal: The process of forming a vendor-neutral business ecosystem: a case study of the Eclipse Foundation.*** Unpublished Ph.D. thesis proposal. Ottawa, CA: Carleton University (Eric Sprott School of Business).

Muegge, S.M. 2011. ***Business ecosystems as metaphor, label, and analogy***. Presented at the 71st Annual Meeting of the Academy of Management, San Antonio, TX, August 16.

Nachira, F., Nicolai, A., Dini, P., Rivera Lèon, L. & Le Louarn, M. 2007. ***Digital business ecosystems: The results and the perspectives of the digital business ecosystem research and development activities in FP6***. Luxembourg: Office for Official Publications of the European Communities.

Nackman, L.R. 2005. ***The Eclipse phenomenon***. Presented at EclipseCon 2005, Burlingame, CA, Thursday March 3. Slides available online: http://eclipsecon.org/2005/presentations/EclipseCon2005_Nackman.pdf

Nelson, R.R. & Winter, S.G. 1982. ***An evolutionary theory of economic change***. Cambridge, MA: Belknap.

Newell, A. & Simon, H.A. 1972. ***Human problem solving***. Englewood Cliffs, NJ: Prentice Hall.

Nokia Corporation press release. November 8 2010. ***Nokia reaffirms commitment to Symbian platform***. Available online: http://press.nokia.com/2010/11/08/nokia-reaffirms-commitment-to-symbian-platform-2

Nokia Corporation press release. February 11 2011. ***Nokia and Microsoft announce plans for a broad strategic partnership to build a new global ecosystem.*** Available online: http://press.nokia.com/2011/02/11/nokia-and-microsoft-announce-plans-for-a-broad-strategic-partnership-to-build-a-new-global-ecosystem

Norberg, J., Wilson, J., Walker, B. & Ostrom, E. 2008. Diversity and resilience of social-ecological systems. In J. Norberg & G.S. Cumming (eds.), ***Complexity theory for a sustainable future***, New York, NY: Columbia University Press: 46-75.

Norman, R. & Ramirez, R. 1993. From value chain to value network: Designing interactive strategy. ***Harvard Business Review***, 71: 65-77.

Norris, J. 2010. ***Rocket science and The Republic.*** Keynote presentation at EclipseCon 2010, Santa Clara, CA, Wednesday March 24.

North, D.C. 1990. ***Institutions, institutional change and economic performance.*** Cambridge, UK: Cambridge University Press.

Object Technology International. 2001. ***Eclipse platform technical overview.*** A technical white paper distributed by the OTI division of IBM. Available online: http://web.archive.org/web/20030406125817/www.eclipse.org/whitepapers/eclipse-overview.pdf

Oberhuber, M. 2008. ***Eclipse architecture council update.*** Presented at Eclipse Summit Europe 2008, Ludwigsburg, Germany, November 17. Slides available online: http://www.eclipse.org/org/foundation/membersminutes/20081117EuropeanMembersMeeting/EAC_Update.pdf

Odum, E.P. 1997. ***Ecology: A bridge between science and society.*** Sunderland, MA: Sinauer Associates.

Odum, E.P. & Barrett, G.W. 2005. ***Fundamentals of ecology*** (fifth edition). Belmont, CA: Thomson Brooks/Cole.

Odum, H.T. 1971. ***Environment, power, and society.*** New York, NY: Wiley-Interscience.

Okey, T.A. 2004. Strategy as ecology. ***Harvard Business Review***, 82(9): 132.

Okhuysen, G.A. & Eisenhardt, K.M. 2002. Integrating knowledge in groups: how formal interventions enable flexibility. ***Organization Science***, 14(4): 370-386.

Okhysen, G.A. & Waller, M.J. 2002. Focusing on midpoint transitions: an analysis of boundary conditions. ***Academy of Management Review***, 45(5): 1056-1065.

Oliver, C. 1991. Strategic responses to institutional processes. ***Academy of Management Review***, 16: 145-179.

Olson, M. 1965. ***The logic of collective action: Public goods and the theory of groups***. Boston, MA: Harvard University Press.

O'Mahony, S. 2003. Guiding the commons: how community managed software projects protect their work. ***Research Policy***, 32: 1179-1198.

O'Mahony, S. 2005. Nonprofit foundations and their role in community-firm software collaboration. In J. Feller, B. Fitzgerald, S.A. Hissam, S.A. & K.R. Lakhani (eds.), ***Perspectives on free and open source software***, Cambridge, MA: The MIT Press: 393-413.

O'Mahony, S. 2007. The governance of open source initiatives: what does it mean to be community managed? ***Journal of Management and Governance***, 11(2): 139-150.

O'Mahony, S., Diaz, F.C., Mamas, E. 2005a. ***IBM and Eclipse (A)***. Harvard Business School case 9-906-007. Boston, MA: Harvard Business School Publishing.

O'Mahony, S., Diaz, F.C., Mamas, E. 2005b. ***IBM and Eclipse (B)***. Harvard Business School case 9-906-011. Boston, MA: Harvard Business School Publishing.

O'Mahony, S. & Ferraro, F. 2007. The emergence of governance in an open source community. ***Academy of Management Journal***, 50(5): 1079-1106.

O`Mahony & Raj, N. 2007. ***The Mozilla Foundation: Launching Firefox 1.0 (A)***. Harvard Business School case 9-907-105. Boston, MA: Harvard Business School Publishing.

O'Reilly, T. 2001. What's next for Linux and open source? ***Linux Magazine***, October. Available online: http://www.linux-mag.com

O'Reilly, T. 2003. ***The architecture of participation***. O'Reilly OnLamp.com, April 6. Available online: http://oreillynet.com/pub/wlg/3017

O'Reilly, T. 2004. ***The architecture of participation***. O'Reilly Network, June. http://oreillynet.com/pub/a/oreilly/tim/articles/architecture_of_participation.html

O'Reilly, T. 2005a. The open source paradigm shift. In C. DiBona, M. Stone & D. Cooper (eds.), ***Open sources 2.0: The continuing evolution***, Sebastopol, CA: O'Reilly Media: 253-272.

O'Reilly, T. 2005b. ***Open source business model design patterns***. Presented at EclipseCon 2005, Burlingame, CA, Tuesday March 1. Slides available online: http://eclipsecon.org/2005/presentations/EclipseCon2005_Tim_OReilly.pdf

O'Reilly, T. 2005c. ***What is web 2.0: Design patterns and business models for the next generation of software.*** O'Reilly Network, September 30. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

O'Reilly, T. 2006. ***Open communities vs. open source***. O'Reilly Radar, July 28. http://radar.oreilly.com/2006/07/open-communities-vs-open-sourc.html

Ostrom, E. 1990. ***Governing the commons: The evolution of institutions for collective action***. Cambridge, UK: Cambridge University Press.

Ostrom, E. 2005. ***Understanding institutional diversity***. Princeton, NJ: Princeton University Press.

Ostrom, E. 2009. Beyond markets and states: polycentric governance of complex economic systems. ***Prize Lecture for the 2009 Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel***, Stockholm Sweden, December 8.

Ostrom, E. & Hess, C. 2007. A framework for analyzing the knowledge commons. In C. Hess & E. Ostrom (eds.), ***Understanding knowledge as a commons: From theory to practice,*** Cambridge, MA: MIT Press: 41-82.

Ostrom, V. & Ostrom, E. 1977. Public goods and public choices. In E.S. Savas (ed.), ***Alternatives for delivering public services: Toward improved performance***, Boulder, CO: Westview Press: 7-49.

Page, S.E. 2007. ***The difference: How the power of diversity creates better groups, firms, schools, and societies***. Princeton, NJ: Princeton University Press.

Parnas, D.L. 1972. On the criteria to be used in decomposing systems into modules. ***Communications of the ACM,*** 15: 1053-1158.

Parnas, D.L. 1974. On a "buzzword": hierarchical structure. ***IFIP Congress '74***, North Halland Publishing Company: 336-339.

Parnas, D.L., Clements, P.C. & Weiss, D.M. 1985. The modular structure of complex systems. ***IEEE Transactions on Software Engineering***, SE-11 (March): 259-266.

Peltoniemi, M. 2006. Preliminary theoretical framework for the study of business ecosystems. ***Complexity & Organization***, 1: 10-19.

Peteraf, M. 1993. The cornerstone of competitive advantage. ***Strategic Management Journal***, 14: 179-191.

Peters, S. 2009. The GNOME Foundation is all about people. ***Open Source Business Resource (OSBR)***, June: 26-29.

Perens, B. 2007. ***The emerging economic paradigm of open source***. http://perens.com/works/articles/Economic.html

Perrow, C. 1961. The analysis of goals in complex organizations. ***American Sociological Review***, 26: 854-866.

Peterson, G., Allen, C.R. & Holling, C.S. 1998. Ecological resilience, biodiversity, and scale. ***Ecosystems***, 1: 6-18.

Pettigrew, A.M. 1990. Longitudinal field research on change: Theory and practice. ***Organization Science***, 1: 267-292.

Pfeffer, J. 1982. ***Organizations and organization theory***. Marshfield, MA: Pitman.

Pfeffer, J. & Salancik, G.R. 1978. ***The external control of organizations: A resource dependence perspective***. New York, NY: Harper & Row.

Pisano, G. 2006. Profiting from innovation and the intellectual property revolution. ***Research Policy***, 35(8): 1122-1130.

Pisano, G.P, & Teece, D.J. 2007. How to capture value from innovation: shaping intellectual property and industry architecture. ***California Management Review***, 50(1): 278-296.

Poole, M.S. Van de Ven, A.H., Dooley, K.J. & Holmes, M. 2000. ***Organizational change and innovation processes: Theory and methods for research***. New York, NY: Oxford University Press.

Porter, M.E. 1980. ***Competitive strategy: Techniques for analyzing industries and competitors***. New York, NY: Free Press.

Porter, M.E. 1985. ***Competitive advantage: Creating and sustaining superior performance***. New York, NY: Free Press.

Porter, M.E. 1990. ***Competitive advantage of nations***. New York, NY: Free Press.

Porter, M. E. 1998, Clusters and the new economics of competition. ***Harvard Business Review***, 76(6): 77-90.

Poteete, A.R., Janssen, M.A. & Ostrom, E. 2010. ***Working together: Collective action, the commons, and multiple methods in practice***. Princeton, NJ: Princeton University Press.

Poundstone, W. 1992. ***Prisoner's dilemma***. New York, NY: Anchor Books.

Powell, W.W. 1987. Hybrid organizational arrangements: new form or transitional development? ***California Management Review***, 30(1): 67-87.

Powell, W.W. 1990. Neither market nor hierarchy: network forms of organization. ***Research on Organizational Behavior***, 12: 295-336.

Prendergast, G. & Berthon, P. 2000. Insights from ecology: an ecotone perspective of marketing. ***European Management Journal***, 18(2): 223–232.

Pressman, R. 2009. ***Software engineering: A practitioner's approach*** (7th edition). New York, NY: McGraw-Hill.

Quarterman, J.S. & Wihelm, S. 1992. ***UNIX, POSIX, and open systems: The standards puzzle***. Boston, MA: Addison-Wesley Longman Publishing Co.

Rahman, M.M., Ruhe, G. & Zimmerman, T. 2009. Optimized assignment of developers for fixing bugs: an initial evaluation for Eclipse projects. ***Third International Symposium on Empirical Software Engineering and Measurement.***

Raymond, E.S. 1997. ***The cathedral and the bazaar.*** Presented at the Fourth Linux Kongress, Würzburg, Germany, Thursday May 22. Abstract available online: http://www.linux-kongress.org/1997/raymond.html

Raymond, E.S. 1998. The cathedral and the bazaar. ***First Monday***, 3(3-2). Available online: http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/issue/view/90

Raymond, E.S. 2001. ***The cathedral & the bazaar: musings on Linux and open source by an accidental revolutionary***. Sebastopol, CA: O'Reilly Media.

Reed, S.K. 2007. ***Cognition: Theory and applications***. Belmont, CA: Thompson Wadsworth.

Richardson, G.B.1972. The organization of industry. ***Economic Journal***, 82: 883-896.

Roberts, E.B. 1988. What we've learned: managing invention and innovation. ***Research-Technology Management***, 31(1): 11-29.

Roberts, E.B. 2007. Managing invention and innovation. ***Research-Technology Management***, 50(1): 35-54.

Roberts, J., Hann, I., & Slaughter, S. 2006. Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the Apache projects. ***Management Science***, 52(7): 984-999.

Roberston, D. & Ulrich, K. 1998. Planning for product platforms. ***Sloan Management Review***, 39(4): 19-31.

Rochet, J.C. & Tirole, J. 2003. Platform competition in two-sided markets. ***Journal of the European Economic Association***, 1(4): 990-1029.

Rochet, J.C. & Tirole, J. 2006. Two-sided markets: a progress report. ***RAND Journal of Economics***, 37(3): 645-667.

Rymer, J.R. 1993. Standards dream will die unless process is retooled. ***Network World***, 10(12): 31.

Sadowski, B.M., Sadowski-Rasters, G. & Duysters, G. 2008. Transition of governance in a mature open software source community: evidence from the Debian case. ***Information Economics and Policy***, 20: 323-332.

Samuelson, P.A. 1954. The pure theory of public expenditure. ***Review of Economics and Statistics***, 36: 387-389.

Samuelson, P. 2006. IBM's pragmatic embrace of open source. ***Communications of the ACM***, 49(10): 21-25.

Sanchez, R. & Mahoney, J. 1996. Modularity, flexibility, and knowledge management in product and organization design. ***Strategic Management Journal***, 17: 63-76.

Scacchi, W. 2007a. Free/open source software development: recent research results and emerging opportunities. ***Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*** (ESEC/FSE'07), Cavtat, Croatia, September 2-7.

Scacchi, W. 2007b. Free/open source software development: recent research results and methods, in M. Zelkowitz (ed.), ***Advances in Computers***, 69: 243-295.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. 2006. Guest editorial: understanding free/open source software development processes. ***Software Process Improvement and Practice***, 11: 95-105.

Schilling, M.A. 2009. Protecting or diffusing a technology platform: tradeoffs in appropriability, network externalities, and architectural control. In A. Gawer (ed.), ***Platforms, markets, and innovation***, Northampton, MA: Edward Elgar: 192-218.

Schlager, E. & Ostrom, E. 1992. Property rights regimes and natural resources: a conceptual analysis. ***Land Economics***, 68(3): 249-262.

Schumpeter, J.A. 1934. ***The theory of economic development: An inquiry into profits, capital, credit, interest and the business Cycle***. Cambridge, MA: Harvard University.

Schumpeter, J.A. 1942/2008. ***Capitalism, socialism, and democracy***. New York: NY: Harper Perennial.

Schwaber, K. & Beedle, M. 2001. ***Agile software development with Scrum***. Upper Saddle River, NJ: Prentice Hall.

Schweik, C.M. 2007. Free/open-source software as a framework for establishing commons in science. In C. Hess & E. Ostrom (eds.), ***Understanding knowledge as a commons: From theory to practice,*** Cambridge, MA: MIT Press, pp. 277-310.

Scotchmer, S. 2006. ***Innovation and incentives*** (paperback edition). Cambridge, MA: MIT Press.

Scott, W.R. 1981. ***Organizations: Rational, natural and open systems***. Upper Saddle River, NJ: Pearson Prentice Hall.

Scott, W.R. 2003. ***Organizations: Rational, natural and open systems*** (fifth edition). Upper Saddle River, NJ: Pearson Prentice Hall.

Scott, W.R. 2008. ***Institutions and organizations: Ideas and interests*** (3rd edition). Thousand Oaks, CA: Sage.

Scott, W.R. & Davis, G.F. 2007. ***Organizations and organizing: Rational, natural, and open system perspectives***. Upper Saddle River, NJ: Pearson Prentice Hall.

Selznick, P. 1949. ***TVA and the grass roots***. Berkeley, CA: University of California Press.

Senge, P.M. 2006. ***The fifth discipline: The art and practice of the learning organization*** (revised and updated edition). New York, NY: Currency Doubleday.

Shah, S. 2006. Motivation, governance, and the viability of hybrid forms in open source software development. ***Management Science***, 52(7), 1000-1014.

Shapiro, C. & Varian, H.R. 1998. Versioning: the smart way to sell information. ***Harvard Business Review***, 76: 106-114.

Shapiro, C. & Varian, H.R. 1999. ***Information rules: A strategic guide to the network economy***. Boston, MA: Harvard Business School Press.

Shatnawi, R. 2010. A quantitative investigation of the acceptable risk levels of object-oriented metrics in open-source systems. ***IEEE Transactions on Software Engineering***, 36(2): 216-225.

Shaw, M. & Garlan, D. 1996. ***Software architecture: An emerging discipline***. Upper Saddle River, NJ: Prentice-Hall.

Shirky, C. 2008. ***Here comes everybody: The power of organizing without organizations***. New York, NY: The Penguin Press.

Siggelkow, N. 2007. Persuasion with case studies. ***Academy of Management Journal***, 50: 20-24.

Simon, H.A. 1947. ***Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations***, New York: Macmillan Publishing Co.

Simon, H.A. 1962. The architecture of complexity: hierarchical systems. ***Proceedings of the American Philosophical Society***, 106 (December): 467-482.

Simon, H.A. 1969. ***The sciences of the artificial.*** Cambridge, MA: MIT Press.

Simon, H.A. 1973. The organization of complex systems. In H.H. Pattee (ed.), ***Hierarchy theory***. New York, NY: George Brziller.

Simon, H.A. 1996. ***The sciences of the artificial*** (third edition). Cambridge, MA: MIT Press.

Singer, J.G. 2006. Framing brand management for marketing ecosystems. ***Journal of Business Strategy***, 27(5): 50–57.

Skerrett, I. 2008a. ***Building technology communities***. Presented at the Technology Innovation Management (TIM) Lecture Series, Carleton University, Ottawa Canada, June 4. Slides available online: http://www.talentfirstnetwork.org/ wiki/images/6/69/Building_technical_communities_June_4.pdf

Skerrett, I. 2008b. TIM Lecture Series: Building technical communities. ***Open Source Business Resource (OSBR)***, June: 29-32.

Skerrett, I. 2009. Collaborative software development in the enterprise. ***Open Source Business Resource (OSBR)***, January: 5-8.

Skerrett, I. 2011. Best practices in multi-vendor open source communities. ***Open Source Business Resource (OSBR)***, January: 11-14.

Skerrett, I. & Beaton, W. 2010. ***Eclipse Foundation: raw and uncut***. Presented at EclipseCon 2010, Santa Clara, CA, Wednesday March 24.

Smith, A. 1776/2003. ***The Wealth of Nations*** (new title edition). New York, NY: Bantham Classics.

Smith, D. & Milinkovich, M. 2007. Eclipse: A premier open source community. ***Open Source Business Resource (OSBR)***, July: 7-10.

Snow, C.C., Strauss, D.R. & Culpan, R. 2009. Community of firms: a new collaborative paradigm for open innovation and an analysis of Blade.org. ***International Journal of Strategic Business Alliances***, 1(1): 53-72.

Stallman, R. 1996a. Free software definition. In J. Gay (ed.), 2002, ***Free software free society: Selected essays of Richard M. Stallman***. Boston, MA: Soho Books, pp. 43-45.

Stallman, R. 1996b. Words to avoid. In J. Gay (ed.), 2002, ***Free software free society: Selected essays of Richard M. Stallman***. Boston, MA: Soho Books, pp. 191-196.

Sterman, J.D. 2000. ***Business dynamics: systems thinking and modeling for a complex world***. New York, NY: Irwin-McGraw Hill.

Sterman, J.D. 2001. System dynamics modeling: tools for learning in a complex world. ***California Management Review***, 43(4): 8-25.

Steward, D.V. 1981. The design structure system: a method for managing the design of complex systems. ***IEEE Transactions on Engineering Management***, 28(3): 71-84.

Stinchcombe, A.L. 1965. Social structure and organizations. In J.G. March (ed.), ***Handbook of Organizations***, Chicago, IL: Rand McNally, pp. 142-193.

Strauss, A.L. & Corbin, J.M. 1990. ***Basics of qualitative research: Grounded theory procedures and techniques***. Thousand Oaks, CA: Sage.

Subramanyam, R. & Xia, M. 2008. Free/libre open source software development in developing and developed countries: a conceptual framework with an exploratory study. ***Decision Support Systems***, 46: 173-186.

Suddaby, R. 2006. What grounded theory is not. ***Academy of Management Journal***, 49: 633-642.

Tansley, A.G. 1935. The use and abuse of vegetational concepts and terms. ***Ecology***, 16(3): 284-307.

Tapscott, D. Ticoll, D. & Lowy, A. 2000. ***Digital capital: Harnessing the power of business webs***. Boston, MA: Harvard Business School Press.

Taylor, W.C. 2006. Here's an idea: let everyone have ideas. ***New York Times***. March 26.

Teece, D.J. 1981. The market for know-how and the efficient international transfer of technology. ***The Annals of the Academy of Political and Social Science***, 458: 81-96.

Teece, D.J. 1982. Towards an economic theory of the multiproduct firm. ***Journal of Economic Behavior and Organization***, 3: 39-63.

Teece, D.J. 1986. Profiting from technological innovation: implications for integration, collaboration, licensing and public policy. ***Research Policy***, 15: 285-305.

Teece, D.J. 1996. Firm organization, industrial structure, and technological innovation. ***Journal of Economic Behavior & Organization***, 31(2): 193-224.

Teece, D.J. 2000. Strategies for managing knowledge assets: the role of firm structure and industrial context. ***Long Range Planning***, 33: 35-54.

Teece, D.J. 2006. Reflections on "Profiting from Innovation". ***Research Policy***, 35: 1131-1146.

Teece, D.J. 2007. Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. ***Strategic Management Journal***, 28(13): 1319-1350.

Teece, D.J. 2009. ***Dynamic capabilities and strategic management***. Oxford, UK: Oxford University Press.

Teece, D.J. & Pisano, G. 1994. The dynamic capabilities of firms: an introduction. ***Industrial and Corporate Change***, 3(3): 537-556.

Thomas, D.A. 1995. Travels with Smalltalk. In ***Celebrating 25 years of Smalltalk***, SIGS Publications. Available online: http://mojowire.com/TravelsWithSmalltalk

Thomson, D. 2004. ***A different shade of blue: moving Eclipse from closed to open source.*** Presented at EclipseCon 2004, Burlingame, CA, Thursday March 3.

Tian, C.H., Ray, B.K, Lee, J., Cao, R. & Ding, W. 2008. BEAM: a framework for business ecosystem analysis and modeling. ***IBM Systems Journal***, 47(1): 101-114.

Tilman, D., Widen, D. & Knops, J. 1996. Productivity and sustainability in grassland ecosystems. ***Nature***, 379(6567): 718-720.

Tilman, D. 2000. Causes, consequences and ethics of biodiversity. ***Nature***, 405(6783): 208-211.

Tilman, D, Reich, P.B. & Knops, M.H. 2006. Biodiversity and ecosystem stability in a decade-long grassland experiment. ***Nature***, 441(1): 629-632.

Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans. 1998 (with 2000, 2002, 2005 amendments). Canadian Institutes of Health Research, Natural Sciences and Engineering Research Council of Canada, Social Sciences and Humanities Research Council of Canada. Available online: http://www.pre.ethics.gc.ca

Tushman, M. & Anderson, P. 1986. Technological discontinuities and organization environments. ***Administrative Science Quarterly***, 31: 439-465.

Udell, J. 2004. The rewards of open source. ***InfoWorld***, 16(33): 30.

Ulrich, K. 1995. The role of product architecture in the manufacturing firm. ***Research Policy***, 24: 419-440.

Utterback, J. 1994. ***Mastering the dynamics of innovation***. Boston, MA: Harvard Business School Press.

Van de Ven, A.H. 2007. ***Engaged scholarship: A guide for organizational and social research***. New York, NY: Oxford University Press.

Van de Ven, A.H., Polley, D.E., Garud, R. & Venkataraman, S. 1999. ***The innovation journey***. New York, NY: Oxford University Press.

Vanhaverbeke, W. 2006. The interorganizational context of open innovation. In H. Chesbrough, W. Vanhaverbeke & J. West (eds.), ***Open innovation: Researching a new paradigm***, Oxford, UK: Oxford University Press: 205-219.

Vanhaverbeke, W. & Cloodt, M. 2006. Open innovation in value networks. In H. Chesbrough, W. Vanhaverbeke & J. West, J. (eds.), ***Open innovation: Researching a new paradigm***, Oxford, UK: Oxford University Press: 258-284.

Vlaar, P.W.L., Van Den Bosch, F.A.J. & Volberda, H.W. 2007. Towards a dialectical perspective on formalization in interorganizational relationships: how alliance managers capitalize on the duality inherent in contracts, rules and processes. ***Organization Studies***, 28(4): 437-466.

von Burg, U. & Kenney, M. 2003. Sponsors, communities, and standards: Ethernet vs. token ring in the local area networking business. ***Industry and Innovation***, 10(1): 351-375.

von Hippel, E. 1986. Lead users: A source of novel product concepts. ***Management Science***, 32(7): 791-805.

von Hippel, E. 1988. ***The sources of innovation***. Oxford, UK: Oxford University Press.

von Hippel, E. 2001. Innovation by user communities: learning from open-source software. ***Sloan Management Review***, 42(4): 82-86.

von Hippel, E. 2005. ***Democratizing innovation***. Boston, MA: MIT Press.

von Hippel, E. & von Krogh, G. 2003. Open source software and the 'private-collective' innovation model: Issues for organization science. **Organization Science**, 14(2), 209-223.

von Krogh, G. & von Hippel, E. 2003. Special issue on open source software development. ***Research Policy***, 32: 1149-1157.

von Krogh, G. & von Hippel, E. 2006. The promise of open source software. ***Management Science***, 52(7): 975-983.

Wagstrom, P.A. 2009. ***Vertical integration in open software engineering communities***. Unpublished Ph.D Thesis. Pittsburgh, PA: Carnegie Mellon University.

Waters, D.J. 2007. Preserving the knowledge commons. In C. Hess & E. Ostrom (eds.), ***Understanding knowledge as a commons: From theory to practice,*** Cambridge, MA: MIT Press, pp. 145-168.

Watts, D. 1999. ***Small worlds: The dynamics of networks between order and randomness***. Princeton, NJ: Princeton University Press.

Watts, D. 2003. ***Six degrees: The science of a connected age***. New York, NY: Norton.

Weber, S. 2004. ***The success of open source***. Boston, MA: Harvard University Press.

Weick, K.E. 1989. Theory construction as disciplined imagination. ***Academy of Management Review***, 14(4): 516-531.

Weick, K.E. 1995. What theory is *not*, theorizing is. ***Administrative Science Quarterly***, 40(3): 385-390.

Weiss, M. & Gangadharan, G.R. 2010. Modeling the mashup ecosystem: Structure and growth. ***R&D Management***, 40(1): 40-49.

Wenger, E.C. & Snyder, W.M. 2000. Communities of practice: The organizational frontier. ***Harvard Business Review***, January-February: 139-145.

Wernerfelt, B. 1984. The resource-based view of the firm. ***Strategic Management Journal***, 5: 171-180.

West, J. 2003. How open is open enough? Melding proprietary and open source platform strategies. ***Research Policy***, 32: 1259-1285.

West, J. 2010. Open innovation's challenge: Letting go is hard to do. ***BusinessWeek Online***, April 1. Available online: http://www.businessweek.com/innovate/content/mar2010/id20100330_486211.htm

West, J. & Bogers, M. 2010. ***Contrasting innovation creation and commercialization within open, user, and cumulative innovation.*** Paper presented at the 70th annual meeting of the Academy of Management, Montreal, Canada, August 10.

West, J. & Gallagher, S. 2006. Patterns of open innovation in open source software. In H. Chesbrough, W. Vanhaverbeke & J. West, J. (eds.), ***Open innovation: Researching a new paradigm***, Oxford, UK: Oxford University Press: 82-108.

West, J., & Lakhani, K. 2008. Getting clear about communities in open innovation. ***Industry & Innovation***, 15(2), 223-231.

West, J. & O'Mahony, S. 2005. Contrasting community building in sponsored and community founded open source projects. ***Proceedings of the 38th Annual Hawaii International Conference on System Sciences***, Waikoloa, HI, January 3-6.

West, J. & O'Mahony, S. 2008. The role of participation architecture in growing sponsored open source communities. ***Industry and Innovation***, 15(2): 145-168.

West, J. & Wood, D. 2008. ***Creating and evolving an open innovation ecosystem: lessons from Symbian Ltd.*** Paper presented at 6th Annual International User and Open Innovation Workshop, Cambridge, MA, August.

Wheelwright, S.C. & Clark, K.B. 1992. ***Revolutionizing product development***. New York, NY: The Free Press.

Whetten, D.A. 1989. What constitutes a theoretical contribution. ***Academy of Management Review***, 14(4): 490- 495.

Whetten, D.A. & Godfrey, P.C. (eds.). 1999. ***Identify in organizations: Building theory through conversations***. Thousand Oaks, CA: Sage Publications.

Whitney, D., Crawley, E., de Weck, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. 2004. ***The influence of architecture in engineering systems. Engineering Systems Monograph***. Presented at The Engineering Systems Symposium (MIT Engineering Systems Division), Cambridge, MA, March 29-31.

Wiener, N. 1948. ***Cybernetics: Or control and communication in the animal and the machine***. New York, NY: The Technology Press, John Wiley.

Williams, B. 2006. ***Wall Street view of Eclipse ... for the Eclipse Members Meeting.*** Presented at the 2006 Eclipse Annual Members Meeting, Santa Clara, CA, March 20. Slides available online: http://www.eclipse.org/org/foundation/membersminutes/20060320MembersMeeting/BrentWilliams-EclipseMembersMeetingMar20-06.pdf

Williams, B. 2007. ***Open source business models: A Wall Street look at a wild 2006 and the prospects for even more fun in 2007.*** Presented at EclipseCon 2007, Santa Clara, CA, March 6. Slides available online: http://stephesblog.blogs.com/presentations/BrentWilliamsEclipseConV02.pdf

Williamson, O.E. 1975. ***Markets and hierarchies: Analysis and antitrust implications***. New York, NY: Free Press.

Williamson, O.E. 1985. ***Economic institutions of capitalism: Firms, markets, relational contracting***. New York, NY: Free Press.

Williamson, O.E. 1991. Comparative economic organization: the analysis of discrete structural alternatives. ***Administrative Science Quarterly***, 36: 269-296.

Williamson, O.E. 2009. Transaction cost economics: the natural progression. ***Prize Lecture for the 2009 Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel***, Stockholm, Sweden, December 8.

Wittgenstein, L.J.J. 1953/2001. ***Philosophical investigations.*** Blackwell Publishers.

Xie, Z. 2008. Open source software foundations. ***Open Source Business Resource (OSBR)***, October: 16-20.

Yang, Z, & Jiang, M. 2007. Using Eclipse as a tool-integration platform for software development. ***IEEE Software***, March/April: 87-89.

Yin, R.K. 2003a. ***Case study research: Design and methods*** (3rd edition). Newbury Park, CA: Sage Publications.

Yin, R.K. 2003b. ***Applications of case study research*** (2nd edition). Newbury Park, CA: Sage Publications.

Ying, A.T.T., Murphy, G.C., Ng, R., & Chu-Carroll, M.C. 2004. Predicting source code changes by mining change history. ***IEEE Transactions on Software Engineering***, 30(9).

Ying, A.T.T., Wright, J.L., & Abrams, S. 2005. Source code that talks: an exploration of Eclipse task comments and their implication to repository mining. ***Proceedings of the 2005 International Workshop on Mining Software Repositories (MSR'05)***, International Conference on Software Engineering (ICSE), Saint Louis MO, May 17.

Yiu, D., Bruton, G.D. & Lu, Y. 2005. Understanding business group performance in an emerging economy: Acquiring resources and capabilities in order to prosper. ***Journal of Management Studies***, 42(1): 183-206.

Zetie, C. 2005. ***Eclipse has won – What next for Eclipse?*** Forrester Research. www.forrester.com/go?docid=36165, July 22.

Zhang, H. & Kim, S. 2010. Monitoring software quality evolution for defects. ***IEEE Software***, 27(4): 58-64.

Zimmermann, T., Premraj, R., & Zeller, A. 2007. Predicting defects for Eclipse. ***Proceedings of the Third International Workshop on Predictor Models in Software Engineering (PROMISE'07)***, International Conference on Software Engineering (ICSE), Minneapolis, MI, May 20.