Copying Effects on Diversity, Growth and Innovation in Mashup Ecosystem – An Evolutionary Approach

by

Solange Sari

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Technology Innovation Management

Department of Systems and Computer Engineering

Carleton University

Ottawa, Canada, K1S 5B6

December 2010

Your file Votre référence
ISBN: 978-0-494-79551-4
Our file Notre référence
ISBN: 978-0-494-79551-4

ABSTRACT

This research aims to describe copying effects over time on diversity, innovation and growth in mashup ecosystems. Mashup ecosystem has been mapped as a network describing the relationship between mashup application and API provider. Previous research speculates that copying processes are a source of network growth. The dataset includes mashups and APIs from the ProgrammableWeb directory from 2005 to 2010. This research outcomes a simulation model to quantify copying factor and a phylogenetic tree framework to estimate evolution rates. It shows the important role of natural copying mechanism which accelerates the innovation process, increases the growth, and drives diversity of the mashup ecosystem. The contribution of this research is to provide a copying centric tool to sense and predict opportunity and threats in the mashup ecosystem.

Key Words: Mashup, Internet Application, Opportunistic Programming, Innovation, Growth, Diversity, Evolution, and Ecosystem.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor Professor Michael Weiss, for his guidance, support, patience and encouragement throughout this research. Also, I am thankful to all professors of the program for their valuable feedback and constructive suggestions at various stages of the research. And, I would like to state my appreciation to other professors and staff of the department for their services.

I would like to take an opportunity to thank my family, friends and colleagues for their constant support and source of encouragement throughout this journey.

TABLE OF COMMENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

1. INTRODUCTION

Technological advances in communications bring new global marketplace i.e. fast growing business ecosystems engaging different new species of products, services, vendors, customers, competitors and partners. To survive and succeed in this environment an organization has to understand both market space and business strategies around multiple players and measure its performance in the ecosystem using metrics such as robustness, productivity and niche construction. In particular, niche creation allows creating new, valuable functions and foster diversity that creates real value. In other words, diversity is one of the key characteristics of a healthy ecosystem. In order to discover gaps and opportunities through many niches, the ecosystem landscape needs to be mapped from diverse perspectives such as value, customer, and technology. Therefore, managers are looking for tools that enable them to evaluate the organization position in the environment.

Businesses adopting open innovation, for instance open source software, need even more a tool to visualize and evaluate partner relationship in the complex and dynamic innovation network. It also may be used to choose the strategic position in the business ecosystem – when it is appropriated to compete or cooperate. Organizational structures have changed to fit the open model by creating internal and external competences to develop innovation. These capabilities are strongly supported by the web platform that offers a wide range of applications to explore, exploit and manage knowledge through the innovation network. For instance, crowdsourcing, open content, and social network are applications for exploring user knowledge as part of the innovation process.

Data management is the new challenge for the next generation of n-sided business platform. A pattern of innovation development and opportunity creation is identified in the business ecosystem. Companies are motivated to expose knowledge in order to create business opportunities leveraging new products or services which customers need. This pattern is found in the mashup ecosystem when API providers exploit data or services to

be combined in mashup applications serving vast segments of businesses. For example, Google provides an interface to query locations on the earth for elevation data – GoogleMaps which is categorized as mapping and used in many mashup domains such as travel, events and real estate.

A mashup application is an interactive web application that combines data and services from multiples sources into one new integrated service. Thus, recombination is the major driver of innovation in the mashup ecosystem. Following an opportunistic programming approach, mashup developers take advantage of existing components to create their own application. Experienced developers use this approach to integrate more functionality in less time and cost, and the combination strategy depends on the knowledge of sequential programmers. Nonetheless, mashup application is intended to be created by end-users without programming skills. To overcome development issues the software industry offers several development tools making easy to create new applications; although, design issues remain in question - how end-users search for, learn and deploy knowledge to develop mashup application. In spite of these challenges, end-users as innovators bring differentiated innovation to the environment, so it is named user innovation. Therefore, these innovation models, user and recombinant, drive the growth and innovation in the mashup ecosystem.

1.1. Related Work

Yu & Woodard (2008) characterize the mashup ecosystem into a three tiers structure: central tier around the Google Maps, middle tier with the most popular APIs, and peripheral tier with less popular APIs. Each tier has an important role in merging the ecosystem's rich network structure. In addition, Weiss & Gangadharan (2009) describe the mashup ecosystem growth as a degree distribution function which follows a power law distribution if the function is scale invariant (it doesn't change if length scales are multiplied by a common factor). They also point out a valuable question, that is, how to

detect and measure the degree of imitation in the creation of mashups versus other growth mechanisms.

In previous studies, Kleinberg (1999) affirms that the copying process is a stochastic mechanism for creating power law degree distributions. In the edge process, for some $\beta \in (0,1)$, at each step, the newly-created node points to a node chosen uniformly at random. An extension of this model presented by Kumar et al. (2000) is the evolving copying models – linear and exponential growth. In similar approach, Sole et al. (2002) and Vazquez (2003) recognize that duplication and mutation (divergence) mechanisms explain the scale-free nature of the biological networks. In these models, two parameters are established when a node is selected at random to be copied: a link with all the neighbors is created with probability p, and one of the two links is removed with probability q.

1.2. Research Proposal

Innovation is understood as a process of "doing things differently", Schumpeter (1939). It suggests that something is created based on previous things. Likewise, recombinant innovation is the construction of new ideas based on past experiences. Thus, it seems natural to say that end-user mashup developers innovate by replicating exiting mashups. Furthermore, related works on network growth suggest that copying phenomenon may exist in the mashup ecosystem evolution.

This phenomenon leads to the main question of this research. How does evolution happen in the mashup ecosystem despite copying mechanism? Many other issues follow it, for instance how copying is measured, what role it plays in the evolution, and how it affects growth, innovation and diversity of the ecosystem. And so, the objective of this study is to find techniques and methods to map and evaluate ecosystem evolution through growth and innovation processes. An exploratory analysis is required crossing technology

innovation management boundaries and including domains such as economy, ecology, biology and network science.

As deliverables, this research aims to provide a network growth model to estimate the copying factor and a tree framework to model innovation, growth and diversity. Furthermore, the proposed methods are applied to evaluate real data mashup ecosystem. This study is relevant to researchers and students to learn about methods to evaluate ecosystem that can be extended to other domains. Moreover, this research is relevant to managers and entrepreneurs to use these methods as a tool to choose market entry strategy, evaluate performance of the organization and its partners, describe patterns of innovation, discovery business opportunity, and so on. Therefore, the value of this research relies on providing a copying centric tool to sense and predict opportunity and threats in the mashup ecosystem. Moreover, this research recommends to stakeholders how the proposed tree framework can be exploited to reveal new forms of competitive advantage.

The remaining document reviews the literature, describes the research method, discusses the results, and makes some final recommendations. Section 2 presents the state of art of the mashup ecosystem in the three main streams of the literature: mashup development, market structure and strategy, and market evolution. Section 3 describes the methodology used in the development of this research including the theoretical framework, proposed evolution models, data acquisition, research method, and research design. Results and discussion in section 3 follow the experiments defined in the research design. The last section consists of the summary of the contributions, conclusions and future work. References and appendices are presented in the end of this document.

2. MASHUP ECOSYSTEM

This chapter describes three main streams of the literature related to the mashup ecosystem. At the micro level, mashup development covers aspects of design, programming, and platform. And, at the macro level, mashup market highlights characteristics and relationships among the main players of the ecosystem, and the evolution of the ecosystem.

> "An economic community supported by a foundation of interacting organizations and individuals - the organisms of the business world. This economic community produces goods and services of value to customer, who themselves are members of the ecosystem. The member organisms also include suppliers, lead producers, competitors and other stakeholders. Over time, they coevolve their capabilities and roles, and tend to align themselves with the directions set by one or more central companies. Those companies holding leadership roles may change over time, but the function of ecosystem leader is valued by the community because it enables members to move toward shared visions to align their investments and to find mutually supportive roles."
>
> Moore (1996) pg.26

2.1. Development

A mashup application is an interactive web application that combines data and services from multiples sources into one new integrated service. The data can be obtained from third party content providers and integrated through different methods such as screen scraping, open APIs, or other protocol used to access data or service such as HTTP, REST, SOAP, etc. Mashups are categorized according to their functionality - data

mashups, photo and video mashups, news mashups, and business mashups, Yu et al. (2008).

2.1.1. Design

To specify the architectural component design the mashup developer has to decide what is the core product concept based on the opportunities to pursue and how much sharing across the platform, Krishnan & Ulrich (2001). The business opportunities might be associated to the existing mashup designs, and components based on the popularity of resources available, Goldenberg et al. (2001). The success of mashup development relies on the supplier and customer involvement, Brown & Eisenhardt (1995). In other words, mashup development depends on component provider's quality of service and constant user/customers feedback. In this flexible process, an architectural design is required to permit several late changes and early versions, MacCormack et al. (2001).

When using existing applications and adaptive development, mashup innovation assumes an uncertain process and relies on improvisation, real-time experience, and flexibility, Eisenhard & Tabrizi (1995). In its modular characteristic, innovation and performance come from its local module or recombination of them. However, the speed and efficiency is a trade-off with the increased time spent in testing and integration, Ethiraj & Levinthal (2004).

Hargadon (2002) highlights the recombinant nature of the innovation process. From this perspective, innovation can be described as the construction of new ideas from existing ones. Benefits include shortening the learning curve by combining known elements, sharing of past experience, and the diversity of problem solving frames.

2.1.2. Programming

Gamble & Gamble (2008) describe mashups as web application hybrids that consume opportunistic assets. And, they identify four types of fitness necessary to evaluate opportunistic assets: function, QoS, contextual, and technical. The combination of the components depends on the knowledge of sequential developers and the particular situations, Ye (2001). Mashup developers are motivated to follow opportunistic approach to integrate functionality quickly, write certain preferred parts of the application over others, limit resources, and diminish the development costs. At the same time, they are building a collaborative community, Haefliger et al. (2008). And, adding new functionality via copy-and-paste is the strongest opportunistic programming characteristic, Brandt et al. (2008), adopted in mashup development. Copying and paste practices are applied to "Frankenstein"[1] hardware and software artifacts, Hartmann et al. (2008).

Two approaches covers mashup development. One, the manual development requires programming skills and intimate knowledge about the schemes and semantics of data sources or the business protocol conventions for message exchange. Other, tool-assisted development uses tools and frameworks to enable even inexperienced end users to mash up their own Web applications, Yu et al. (2008). Therefore, end users are able to create their own situated applications, Balasubramaniam et al. (2008), using mashup development tools which provide simplicity, usability, and ease of access to allow end-users concentrate on the essence of the problem. However, this task requires not only a user interface, but also identify, analyze, aggregate and manipulate the underlying data, Zang et al. (2008)Zang, N., Rosson, M.B., and Nasser, V. (2008).

---

[1] Put together different parts

Through a sensemaking process the users develop strategies for information seeking, gathering, and consumption to adapt to the flux of information in the environment, Pirolli & Card (1999). There are three distinct steps of the mashup creation processes: gathering, manipulation and visualization of the data, Zang (2008). Practical evaluating of the web-active user's mental model results that participants approached the task by trial and error; the combination of the modules happen without investigation; the naming schemas used for tools are very important for the comprehension of end users; and participants have problems to break down the task into procedural steps, Zang & Rosson (2009).

2.1.3. Platform

Mashup services take advantage of the collaborative web platform where upfront cost for creating service used to be high, but incremental costs were low, Shuen (2008). These services are the technological perspective on web 2.0 providing dynamic data processing and recombination. And, they describe a competition model of the value creation with limited production done by small number of professionals and open filtering done by mass, Iyer & Davenport (2008). This environment is an n-sided platform where API and development tools providers rely on network effects to create value. The positive network effects increase the value of service as more mashups use or adopt it, Balasubramaniam et al. (2008).

This platform does not fit properly into traditional competitive innovation models such as incremental, architectural and radical innovation. A combination of these models with companies and users as constructors brings alliances to expand and open up markets to create a win-win situation, Shuen (2008). The open innovation paradigm allows companies to exploit and explore technology in order to create maximum value from their technological capabilities or other competencies, Chesbrough (2003). It moves innovation models towards user innovation where the customer is the innovator – an early approach about lead users, von Hippel (1986). And the firm's fast adaptation

through product innovation, Eisenhardt & Tabriz (1995), should be aligned with each stage of the market development life cycle, Moore (2004).

2.2. Market

2.2.1. Strategy and Structure

Borrowing the metaphor from, biological ecosystem (Figure 1a) where the resources flow up to food chain, Rothschild (1990) describes economy ecosystem where resources flow up the value-added chain from mines and farms to manufacturer, assembler, and service firms. In this chain (Figure 1b), human work is the main source of energy which power the economic system. Supplier, organization and customer are entities in the ecosystem that supply products among them. At each link the energy depends on consumption or profit that may lead to bankruptcy or reinvestment, respectively. The environmental waste returns to the bottom of the value-added chain to be recycled. Furthermore, in the economic environment, technological information and cultural values, captured in books, journals, databases, and the know-how of millions of individuals, are the ultimate source of all economic life. Thus, information is the essence of the system. Likewise the genetic information recorded in the DNA molecule is the basis of all life. Both nature and business systems present key phenomena such as competition, specialization, cooperation, exploitation, learning, and growth. Both competition and cooperation phenomena have produced the variety and productivity of the global market economy.

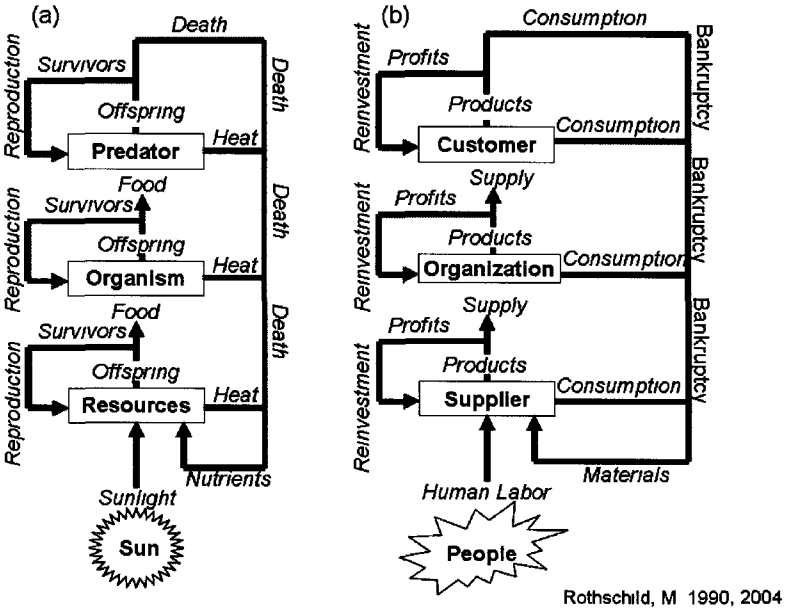


Figure 1 – Ecosystem food chain (a), and economy value chain (b), Rothschild (1990).

Based on this ecological trend Moore (1993) suggests that companies should be part of a business ecosystem that covers a variety of industries. In this environment, organizations co-evolve capabilities around a new innovation working cooperatively and competitively to support new products, satisfy customer needs, and eventually incorporate the next round of innovations. The death of the competition in Moore (1996) emphasizes the new strategic concept that also was called as co-opetition in Brandenburger & Nalebuff (1996). Competitive interactions happen as a sequence of competitive events, Ferrier (2001). An entrant has to choose their position in the ecosystem which impacts the evolution of market structure, Gans & Stern (2003). Therefore, innovators need to envision the new market equilibrium and align players across the market, Chakravorti (2004). Helping out in this choice Iansiti & Levien (2004) present a framework to decide which organizational strategy is more appropriated based on the level of the turbulence and innovation, and complexity of relationships Figure 2. The organizational strategies are summarized in Table 1.

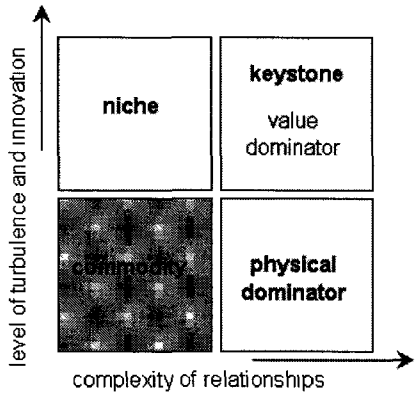


Figure 2 - Organizational strategy decision, Iansiti & Levien (2004).

Strategy	Organization
Niche	Business faces rapid and constant change and, by leveraging the assets of other firms, can focus on a narrowly and clearly defined business segment
Keystone	Business is at the center of a complex network of asset-sharing relationships and operates in a turbulent environment
Physical dominator	Business relies on a complex network of external assets but operates in a mature industry
Commodity	Business in a mature and stable environment and operate relatively independently of other organizations

Table 1 – Organization descriptions for each ecosystem strategy, Iansiti & Levien (2004).

Miller & Olleros (2007) refer this structure as cluster of interdependent firms contributing to the building of a set of interacting products and services tend to self-organize themselves into distinct and relatively persistent "games of innovation". Six games of innovation around value creation exchanges are identified:

- market creation - patent-driven discovery, systems integration, and platform orchestration;
- Market maintenance - cost-based competition, systems extension and engineering, and customized mass production.

The open innovation model is well accepted in the business ecosystem literature. In contrast to the traditional vertical integration model Chesbrough (2003) suggests that

firms can and should use external ideas as well as internal ideas, and internal and external path to market, as they look to advance their technology. Figure 3 depicts the points of distinction between the two innovation processes. Because projects can only enter in one way, at the beginning, and can only exit in one way, by going into the market the process is called closed innovation (Figure 3a). On the other hand, open innovation process allows projects to be launched from either internal or external technology sources, and new technology can enter into the process at various stages (Figure 3b). As well, there are many ways projects can go to market for instance through out-licensing, or a spin-off venture company, or yet through the company's marketing and sales channels. A good example of open innovation is open source software project when including external ideas to the software development and business model to capture the value creation. Chesbrough et al. (2006) describe other interesting examples of firms and institutions adopting open innovation, as well their innovation networks.

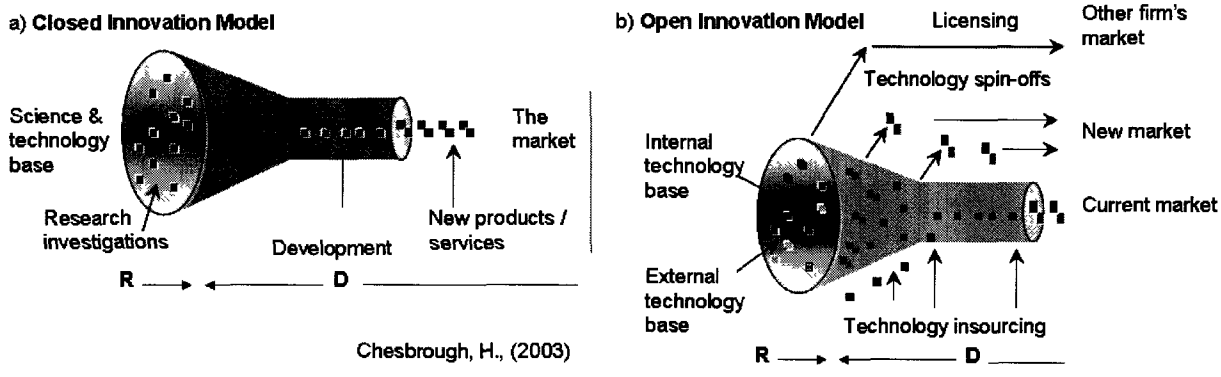


Figure 3 – Closed and open models of innovation, Chesbrough (2003).

Profiting from Innovation – PFI is a framework proposed by Teece (1986) that explains the distribution of profits from innovation among innovator, customers, suppliers, and imitators (and followers). The hypothesis is that appropriability and innovation performance is not generally related to the innovator's market share, but to the (complementary) asset structure of the innovator, management's market entry timing decisions, and the contractual structures employed to access missing complementary assets. The innovator challenge is not only creating value from innovation, but also

capturing that value as well. One strategy is to build protective barriers around innovations in order to assure the returns. These barriers can take the form of legal protection (such as patents, copyrights, or trade-secrets) as well as other strategies such as investing in complementary assets (such as manufacturing, distribution, brand, services, and technologies). Alternative strategy is to lower down the barriers depending on the innovator position in the market. For instance, innovators can sometimes benefit by weakening the intellectual property environment and opening the architecture of the industry, Pisano & Teece (2007).

2.2.2. Capability

Dynamic capability is the ability to achieve new forms of competitive advantage. It is the capacity of adapting, integrating, and reconfiguring intemal and extemal organizational skills, resources, and functional competences to match the requirements of a changing environment, Teece et al. (1997). Therefore, companies create opportunities to leverage and capitalize on the range of internal and external capabilities through dynamic capabilities focusing on the innovative combination and orchestration of a multicompany ecosystem of global partners, users, and customers, Shuen (2008).

Teece (2007) presents a framework to orchestrate capacities giving fundamental support to a company to successfully innovate and capture sufficient value to deliver superior longterm financial performance. In this model, dynamic capabilities are separated into the capacity (1) to sense and shape opportunities and threats, (2) to seize opportunities, and (3) to maintain competitiveness through enhancing, combining, protecting, and, when necessary, reconfiguring the business enterprise's intangible and tangible assets.

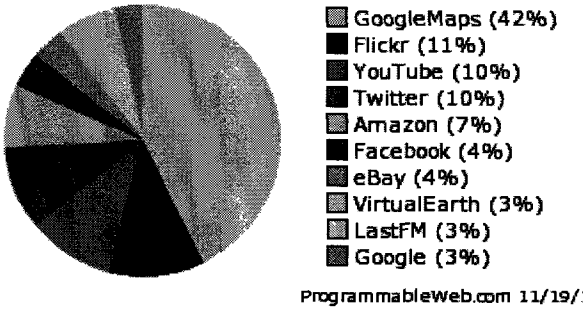
Organizational structures have changed internally and externally to fit company's competitive strategy. As example, O'Connor & Ayers (2005) describe competences to development radical innovation capabilities within the organization. And, Vanhaverbeke et al. (2008) explain the benefits of open innovation in firm's business strategies. The

open external capabilities can be, for instance, innovations contest and niche construction, Terwiesch & Xu (2008) and Luksha (2008).

2.2.3. Opportunity Creation

In the public mashup scenario, API providers benefit from the ecosystem by creating a cost-effective distribution channel for their unique services and content what may lead them to a central position. Google and Yahoo expose APIs in different segments such as maps, search, and directory; thus they are dominators trying to establish a platform of development. Complementary API[2] providers assume a niche position when leveraging assets of other company. Similar positions are occupied by third-party developers that create value by combining the APIs and offering a specific service to the customers. In other words, API providers create opportunity for user developer to produce other opportunistic business. An expressive sample of mashup and APIs is listed in the well-known open directory ProgrammableWeb - PW[3], and the Figure 4 (a) and (b) show the proportion of top-10 APIs and mashup tags, respectively from Sept 2005 to Nov 2010.

a) 2364 APIs



b) 5376 Mashups

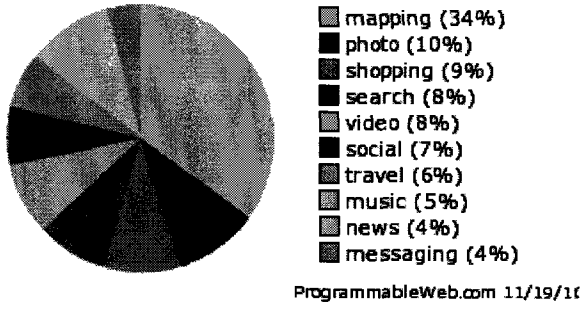


Figure 4 – Top-10 APIs and mashups in the ProgrammableWeb directory

[2] An API is complementary when used with other API in a mashup.

[3] PW – ProgrammableWeb, http://www.programmableweb.com/ dataset updated November 2010

In the enterprise scenario, organization's IT department provides APIs for no programmers to create information management applications. Hoyer & Fischer (2008) classify mashup development tools where market trend in context of enterprise mashups is fluctuating. Software developers urge for standardization and create the Open Mashup Alliance (OMA) to promote the Extensible Mashup Markup Language (EMML) for creating enterprise mashup, Kavanagh (2010).

2.3. Evolution

Moore (1993) establishes four stages to the development of a business ecosystem, each one with its overall challenge: 1) pioneering (value); 2) expansion (critical mass); 3) authority (lead co-evolution), and e) renewal (continuous performance improvement). Using this framework to evaluate a professional open source ecosystem Watson et al. (2005) affirms that ecosystems evolve over time and this evolution is separate from company evolution. According to Iansiti & Levien (2004) there are three determinants of business ecosystem health: *robustness* – ability to survive disruptions and unforeseen changes; *productivity* - ability to consistently transform technology and raw materials(labour, process) into lowered costs, new products, and functions; and *niche creation* - ability to create new, valuable functions and foster diversity that creates real value.

The ecosystem structure can be mapped as a network linking organizations relationships. Using concepts and methods from social network analysis, de Nooy et al. (2005) and Caldarelli (2009), Venkatraman & Lee (2004) identify overlap technology among companies in the video industry, and Iyer et al. (2006) recognize the *small world* effect[4]

---

[4] small-world effect - the diameter of network can remain very small when the number of vertices increases, Watts & Strogatz (1998).

in the network of alliance between software companies. Both works point out preferential attachment which is a growth mechanism in scale-free networks. In particular, Yu & Woodard (2008) characterize the mashup ecosystem into a three tiers structure: one central tier around the Google Maps API, other tier with most popular APIs, and another with less popular APIs which have important in merging the ecosystem's rich network structure.

2.3.1. Growth and Diversity

To reproduce the outstanding evidence of time growth, applied to many real networks, Barabási & Albert (1999) propose a model with two rules: 1) *growth* implies that new vertices enter the network at some rate; and 2) *preferential attachment*[5] means that these newcomers establish their connections preferentially with vertices that already have a large degree. These effects produce naturally scale-free networks in the sense that the degree distribution is power-law distributed $P(k) \propto k^{-\gamma}$.

Particularly, Weiss & Gangadharan (2009) examine the structure of the mashup ecosystem and model its growth over time. They point out the concept of user innovation where the locus of innovation is the customer instead of company. Also, the recombination innovation is described as the construction of new ideas from existing ones. These concepts are well expressed by the complexity of mashup functionalities as the mashup platforms improve its capabilities. One valuable question remains in this research how to detect and measure the degree of imitation in the creation of mashups versus other growth mechanisms.

---

[5] Preferential attachment is also known as *Yule Process, Matthew effect, Rich gets richer, and Cumulative advantage*, Caldarelli (2009).

Observing statistics of biological taxa, Yule (1924) detected that the distribution of the number of species per genus follows a long-tailed form, and proposed a stochastic model to fit this data. Newman (2005) states that the Yule process is one of the most convincing mechanisms for generating power laws and describes it as follow. Yule model assumes that new species are added to the phylogenetic tree by splitting one species in two. It also assumes that the growth happens at constant rate proportional to k species in each m speciation events. In this process, a tree with n genera each with k species grows by adding m+1 new species on average, and the distribution of the sizes of genera when $n \to \infty$ is given by Eq. 1 that can be approximated to a beta function resulting in Eq. 2. In this model new species are added to the tree by splitting one species in two, thus m = 2 and $\gamma \propto 2.5$.

$$P(k) = \frac{m}{m+1}[(k-1) * P(k-1) - K * P(k)] \qquad \text{Eq. 1}$$

$$P(k) \propto (1+\frac{1}{m})B(k, 2+\frac{1}{m}) \text{ and } B(a,b) \propto a^{-b}$$

$$P(k) \propto k^{-\gamma} \qquad \text{Eq. 2}$$

where $\gamma = 2+1/m$.

In bioinformatics, the clustering method neighbor-joining is widely used to construct the phylogenetic tree from evolutionary distance data. The method, initially proposed by Saitou & Nei (1987), consists of interactively select a pair of taxa, creating a new node which represents the cluster of these taxa, and reducing the distance matrix by replacing both taxa by this node. The basic algorithm uses Eq. 3 to compute the new matrix Q based on a distance matrix d related to r taxon. An extension of this method is presented by Gascuel (1997), called BioNJ, which uses a simple first-order model of the variance and covariance of evolutionary distance estimates.

$$Q(i,j) = (r-2)d(i,j) - \sum_{k=1}^{r} d(i,k) - \sum_{k=1}^{r} d(j,k) \qquad \text{Eq. 3}$$

Diversification of clades is commonly modeled as a birth-and-death process where speciation occurs at a constant rate, λ, and extinction at a constant rate, μ, conveying to either exponentially increasing or decreasing diversity. Diversification rate is defined as the difference between speciation and extinction rates (Eq. 4). The relative extinction rate is given by the fraction of these rates (Eq. 5), Bailey (1964).

$$r = \lambda - \mu \qquad \text{Eq. 4}$$

$$\varepsilon = \frac{\mu}{\lambda} \qquad \text{Eq. 5}$$

The growth process can be reconstructed as a phylogenetic tree by tracing lineages that have given rise to at least one contemporary descendant, Nee et al. (1994). Following this approach, several methods have been proposed to estimate diversification rates, for instance maximum likelihood estimators and method-of-moments estimator described in Magallon & Sanderson (2000). Aldous (2001) discusses stochastic modelling for phylogenetic trees and calls for statistic descriptive modelling.

Because it is hard to quantify evolution rates when they vary widely between lineages Sanderson (2002) suggests a saturated model in which every lineage has a separate rate is combined with a roughness penalty that discourages rates from varying too much across a phylogeny. The semi-parametric method based on penalized likelihood is used to smoothly transform the tree by lambda parameter - a trade-off between individual rate for each branch and minimized change rates in contiguous branches.

Rabosky & Lovette (2008) present a modeling framework for speciation and extinction rates that vary continuously through time. They affirm that a simple way to model the growth of a phylogenetic tree through time is to "split" the tree into a collection of daughter branches, with each branch originating at some time t, and surviving to the present day (time T). Considering $\lambda(t)$ and $\mu(t)$ time-varying speciation and extinction rates, respectively, the model assumes that declining net diversification rates through time could result from three general processes:

(1) SPVAR (time-varying speciation only), declining speciation through time but constant extinction (a three-parameter model: $\lambda_0, k,$ and μ_0);

$$r(t) = \lambda_0 \exp(-kt) - \mu_0 \quad \text{Eq. 6}$$

(2) EXVAR (time-varying extinction only) increasing extinction through time, but constant speciation (three parameters: $\lambda_0, \mu_0,$ and z)

$$r(t) = \lambda_0 - \mu_0[1 - \exp(-zt)] \quad \text{Eq. 7}$$

(3) BOTHVAR (both speciation and extinction vary through time).declining speciation rates and increasing extinction rates through time (four parameters: $\lambda_0, k, \mu_0,$ and z).

$$r(t) = \lambda_0 \exp(-kt) - \mu_0[1 - \exp(-zt)] \quad \text{Eq. 8}$$

Paradis (2006)'s book describes many of these methods implemented in the R-packages: ape, ade4 and apTreeshape. Additional R-packages are used in this research such as geiser, laser, igraph, and R basic packages.

2.4. Copying Motivations

Copying arises naturally in different circumstances. For instance, in the context of citations, an author may refer fundamental related work through few familiar references, Price (1965) and Krapivsky & Redner (2005). All similar works, in a specific subject area, mention the primary research which has high incidence of references. As another example, web page authors create links to pages exhibiting relative commonality, Kleinberg (1999) and Kumar (2000). Again, similar pages, in a particular domain, connect to essential pages which have high occurrence of links.

Analogous copying process is deployed in opportunistic programming where developers interleave foraging for examples, learning and writing code, Brandt et al. (2009). Following the process ideation, programmers exploit existing code to develop their own program, or even use it as component of their system. This approach allows developers to

integrate functionality quickly, write certain preferred parts of the code over others, limit resources, and diminish the development costs, Haefliger et al. (2008). The copying and pasting strategy can be extend to "Frankensteining hardware and software artifacts by joining them with duct tape and glue code", Hartmann et al. (2008)

Assuming that the combination strategy depends on the knowledge of individual programmers and the particular situation, Ye (2001), end-user developers rely on information foraging strategies to learn and develop their applications. Using online information in familiar websites such as mashups directory and social networks, and using mashup development tools such as Yahoo Pipes, web-active end-users develop mashup application by trial and error, Zang & Rosson (2009). This way, similar mashup structures are developed in several application areas giving preference to well-established APIs.

3. RESEARCH DESCRIPTION

This section presents how to answer the research question: how does evolution happen in the mashup ecosystem despite copying mechanism? The inquiry is about how copying affects in terms of growth, innovation and diversity in the mashup ecosystem. The literature gives many venues to pursue as described in the theoretical framework. From this, two first research outcomes are described; one presents a model to compute copying factor from the network growth model and the other suggests a framework to estimate evolution parameters from a phylogenetic tree in terms of innovation, growth and diversity. Furthermore, this section describes the dataset, computational methods, and evaluation methods.

3.1. Theoretical Framework

The initial goal is to seek for techniques and methods to map and evaluate ecosystem evolution through growth and innovation processes. Some insights emerge from the literature review. From previous works in the technology management area, the mashup ecosystem growth is modeled by the API node degree distribution that follows a power law distribution. It suggests that APIs are selected by their popularity describing the preferential attachment phenomenon in which the more links a node has, the more likely it is to be selected as target of a link by nodes added to the network. However, this model does not explain effects in innovation or diversity of the ecosystem. Crossing study area boundaries and instantiating the bionomic model (Figure 1), it is inferred that mashup development is the central process that defines the ecosystem evolution. It contains essential information of the system when combining different services from providers and offering different services to customers, called user recombinant innovation. Although mashup development depends on existing services it defines the technological information change that explains the evolution of the ecosystem. Thus, diversity evolves from two sources of change: 1) internal or genotypic diversity - the variation is derived by the API recombination, and 2) external or phenotypic diversity - the difference is the

usability of the mashup application. Figure 5 illustrates the main concepts of this research. Mashup development, market and ecosystem evolution are the main streams of the literature review. And, growth, innovation, diversity, and evolution are the explored aspect under effect of copying.

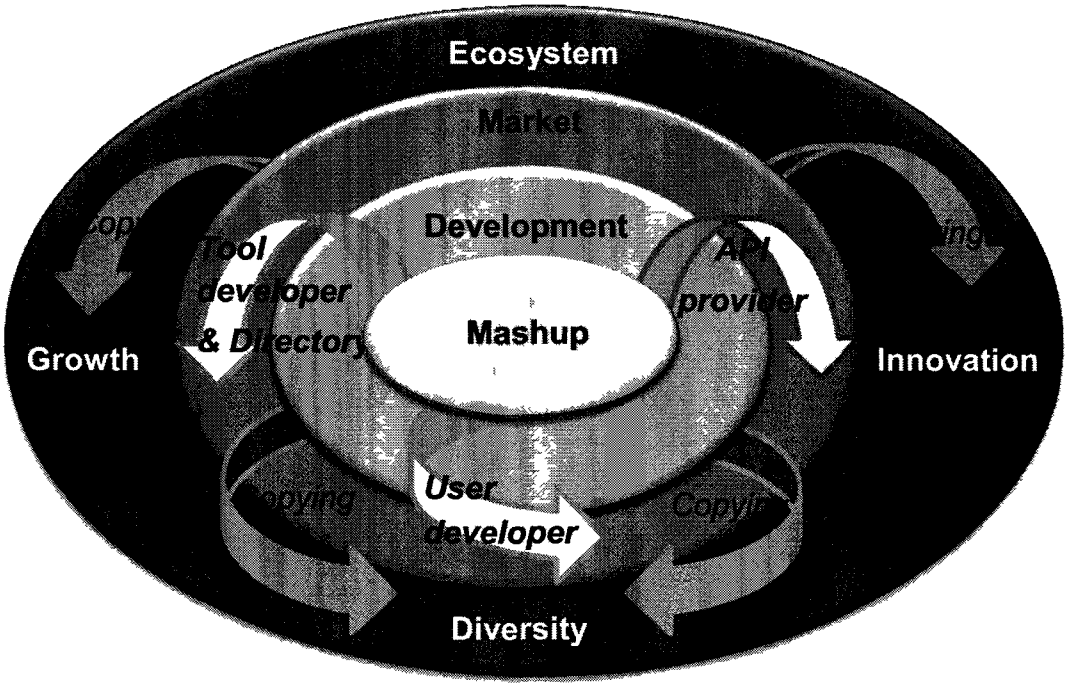


Figure 5 - Theoretical framework

3.2. Evolution Models

3.2.1. Network Growth Model

Mashup ecosystem is mapped as a bipartite graph, $G=(M\cup A,E)$ where two sets of nodes M (mashups) and A (APIs) are associated to the set E (edges or links between the nodes). Mashup nodes $m\in M$ are only connected to API nodes $a\in A$. If a mashup m combines the Google Maps (a_{gm}) and Flickr (a_f) APIs, the ecosystem graph will contain the edges (m,a_{gm}) and (m,a_f). The total number of node in the graph is $N=|M\cup A|$.

Growth network happens when APIs and mashups are added continually to the network. At each timestep t a new API is added to the network with probability p. With probability $1 - p$, an existing mashup m_s is selected from the set of mashups M_{t-1} at timestep *t-1*. This mashup provides the template of APIs for a new mashup to be added to the network. For each API in the template, the API is either copied to the new mashup or substituted by a random API. With probability α, the API is copied from the template. With probability $1 - \alpha$, a new API is chosen at random from the set of APIs A_{t-1} at timestep t - 1. An α close to 1 implies that most APIs are copied from the template, whereas an α close to 0 means that most APIs are chosen at random. From this rational the first proposition outcomes:

Proposition 1 – The mashup ecosystem grows by copying and the proportion of copied APIs in a mashup defines the copying factor.

Figure 6 depicts an example where two APIs (nodes 1 and 2) and one mashup (node 3) are created at first. The solid lines indicate that these links were randomly selected. Next, assume that mashup (node 4) is a full copy of the node 3, so the dashed lines indicate copied links. And, node 5 is a partial copy of the nodes 3 and 4, since it is not connected to node 2 and connected to node 6.

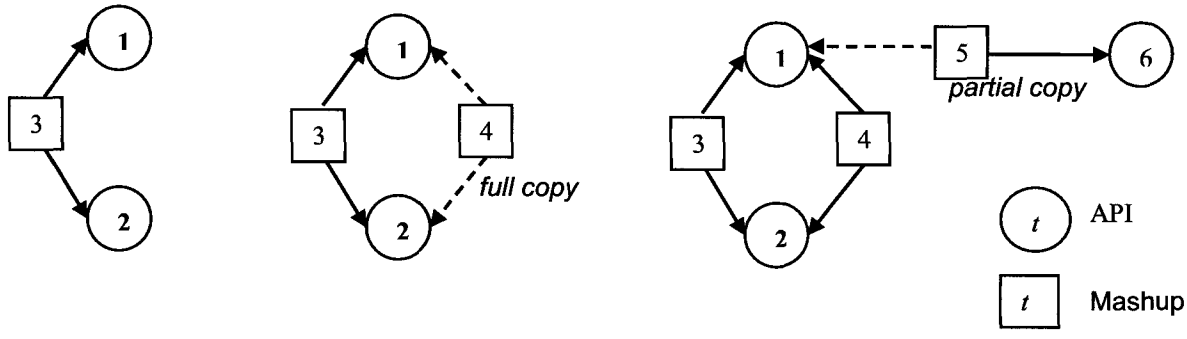


Figure 6 - Example of full and partial copy in a network.

3.2.2. Phylogenetic Tree Framework

Previously, mashup ecosystem was mapped as a bipartite graph, $G = (M \cup A, E)$ where two sets of nodes M (mashups) and A (APIs) are associated to the set E (edges or links between the nodes). Now, a projection of G is taken to represent the affiliations among mashups.

3.2.2.1 Innovation

$P = (MxM, S)$ is an unipartite graph where M is the set of nodes (mashups) and S is the similarity set. If two mashups $m_i, m_j \in M$ use at least one same API, such as Google Maps (a_{gm}), then they have a similarity relationship defined by Eq. 9. This is a well-known similarity metric, called Jaccard Index, which computes the proportion of equal elements in relation to the total elements of the two sets, Jaccard (1901).

$$S(m_i, m_j) = \frac{|m_i \cap m_j|}{|m_i \cup m_j|} \qquad \text{Eq. 9}$$

The tree construction is based on evolutionary distances between mashups, as suggested by Saitou & Nei (1987). In this case, the distance matrix d related to n mashups is defined by Eq. 10.

$$d(m_i, m_j) = 1 - S(m_i, m_j) \qquad \text{Eq. 10}$$

The tree estimation is defined as a birth-death process. Figure 7(a) shows the birth of the species or mashup combinations (m_1, m_2) and $((m_1, m_2), m_n)$ before time t. This means that m_2 is the closest distance to m_1, and m_n the closest distance to m_1 and m_2. In other words, the mashups m_1, m_2 and m_n are similar because they use one or more same APIs. This group of mashups is also referred as a cluster or clade. Some species do not survive after time t and become extinct to the present time T as showed in the Figure 7(b).

Because the past information is not available the tree is constructed only with lineages (bold lines) that have some descendants at the present - extant species in Figure 7(c) which is converted to Figure 7(d).

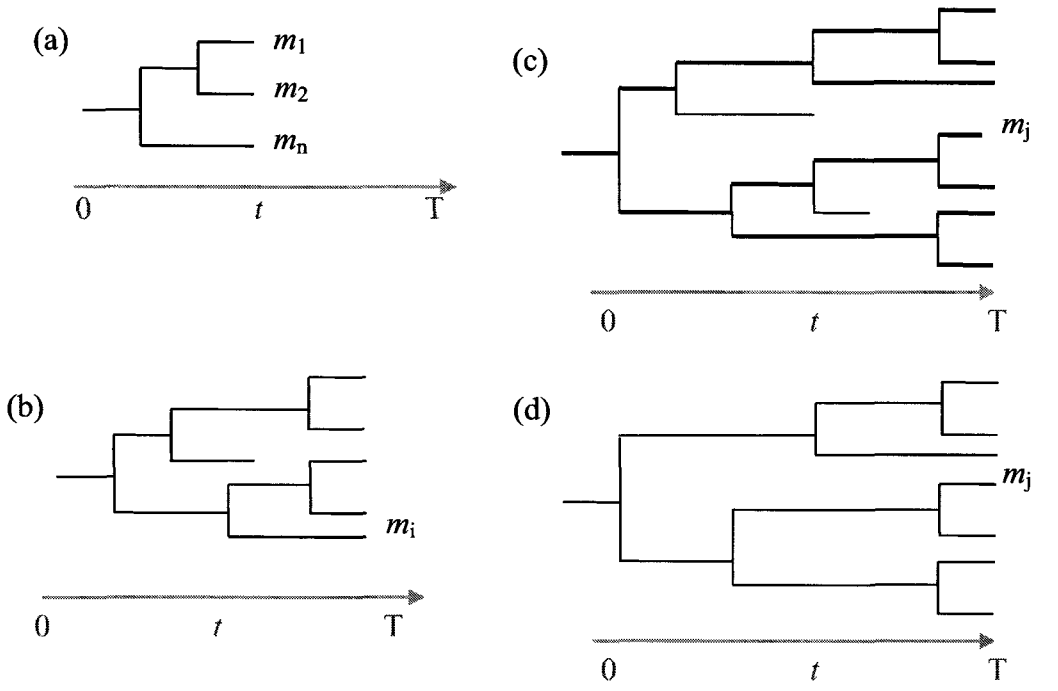


Figure 7 – Phylogenetic tree evolution based on birth-death process of combining APIs

The BioNJ method proposed by Gascuel (1997) is selected to estimate the phylogenetic tree structure because it computes variance and covariance of evolutionary distance to estimate the distance among mashup species. Calibration procedures are necessary to properly deploy evolution rate estimation. Two transform functions are suggested: one resolving multichotomies and another estimating node ages. For the latter, the parameterized method proposed by Sanderson (2002) is used to quantify evolution rates since they vary widely between lineages. So, the lambda parameter can be optimized to calibrate the tree structure to fit appropriately the dataset.

Consider that each node represents the reconstructed time of a mashup species. In other words, the node age is when the combination of APIs happens. According to recombinant

innovation, each combination is understood as one step in the innovation process. Pace of innovation is defined as the distance from the node to the tip of a parameterized phylogenetic tree. And, the innovation pace distribution is represented by the branching-time distribution. This rational describes the following proposition.

Proposition 2 - Phylogenetic tree reconstructs innovation processes, and it can be modeled by the scale-invariant branching-time distribution.

3.2.2.2 Growth

The mashup tree growth can be modelled according to Yule's process, equations Eq. 1 and Eq. 2. It assumes that the growth happens at constant rate proportional to k species in each m speciation events. In this process, a mashup phylogenetic tree with n clades, each with k species grows by adding m+1 new species in average, and the distribution of the sizes of clades when $n \to \infty$ is a function with scaling invariant. In other words, the tree grow follows the power law $P(k) \propto K^{-\gamma}$ with $\gamma = 2+1/m$.

Once again, the growth model show evidences of preferential attachment phenomenon. Therefore, large clades have more probability to add new species than small clades. It also raises another question about copying the mashup species. Since this model is based on evolutionary process, it assumes copying as a natural mechanism of growth. Nonetheless, identical copying may affect innovation and growth process. Moreover, the phylogenetic tree growth follows a random walk to stabilize its structure, thus the number of lineages vary from time to time. This reasoning defines the second proposition in terms of growth based on the number of lineages.

Proposition 3 - Phylogenetic tree growth describes the ecosystem growth, and it can be modeled by the scale-invariant distribution of the number of lineages.

3.2.2.3 Diversity

The tree structure grows dynamically, and it is modeled as a birth-and-death process where speciation occurs at a constant rate, λ, and extinction at a constant rate, μ, conveying to either exponentially increasing or decreasing diversity. The diversification model defined in equations Eq. 4 and Eq. 5 does not consider that the rates vary over time and they might not be constant. Variant and parameterized models defined in equations Eq. 6, Eq. 7 and Eq. 8 seem appropriated to describe the heterogeneity of the mashup ecosystem over the time. A speciation event happen when the new species kept the traits from its ancestor; otherwise it is an extinction event which give rise to distinct clades. Diversity is estimated by the number of diversity events happening through a balance between speciation and extinction rates. The likelihood of diversity events increases when the extinction fraction is maximized, so the diversification rate is minimized, Eq. 11 and Eq. 12.

$$\min r = \lambda - \mu \qquad \text{Eq. 11}$$

$$\max \varepsilon = \frac{\mu}{\lambda} \qquad \text{Eq. 12}$$

Proposition 4 - A diversity event is a balance between speciation and extinction rates, and it can be estimated by minimizing diversification rates and maximizing extinction fraction.

Figure 8 illustrates a hypothetical example where 7 mashups use a combination of 3 API - m_1 (a_1, a_2, a_3), m_2 (a_1, a_2), m_3 (a_1, a_3), m_4 (a_2, a_3), m_5 (a_1), m_6 (a_2), m_7 (a_3). The matrix D is result of the divergence metric that is dependent to the similarity metric. From the distance matrix D the BioNJ method constructs the tree which is showed in (a). The node n_0 has bases in (a_1, a_3), and it generates two branches from n_1 and n_3. Likewise, n_1 has bases in (a_1, a_2, a_3) and generates the mashups m_2 and m_5. Observe that, either m_2 or m_5 has a_3 API, so this trait was extinct when m_2 and m_5 were created. Also, a_1 is extinct in the clade generated by n_3. This demonstrates the role of extinction in creating diversified clades. Furthermore, Figure 8(b) depicts the tree structure after the penalized likelihood method is applied (Chronopl). Note that the branch length are normalized and they allow

to compute the age of the nodes, for instance the n_3 age is 1 – 0.52, where 1 is the total time of the reconstruction.

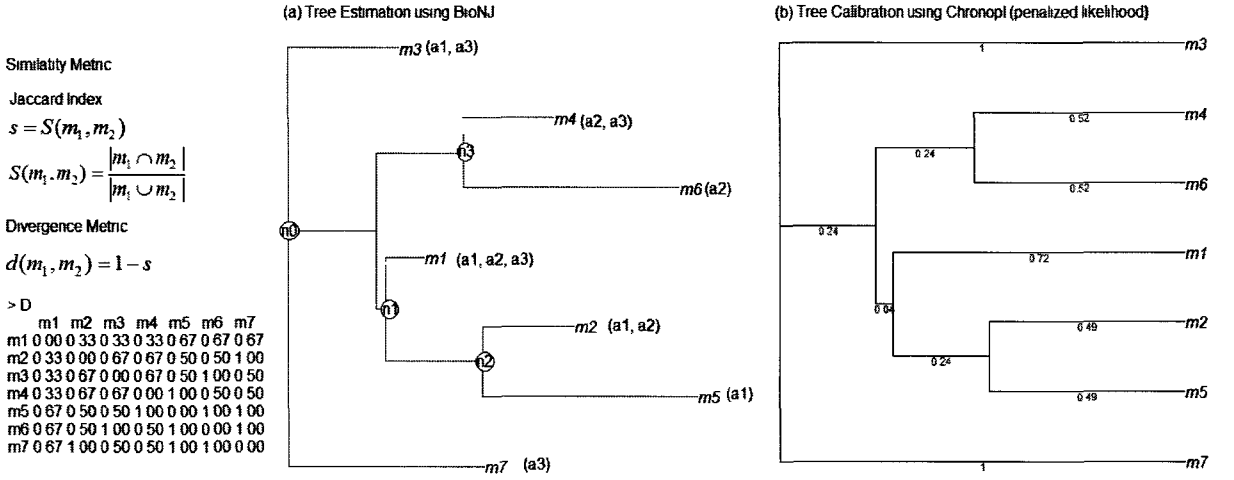


Figure 8- A hypothetical example of tree estimation using BioNJ and Chronopl methods.

3.3. Data Acquisition

Mashup technology has opened many frontiers mainly in the enterprise software business. This research take as sample the mashups and APIs listed in the ProgrammableWeb (PW) directory, an Alcatel-Lucent company since June 2010. PW assured the access to their open API[6] to this research, which is thankfully appreciated.

A web scrapper application extracts the data from the PW website and storage it to be managed accordingly to the experiment. Since September 2005 until the last update in November 2010, the dataset counts:

2364 APIs	5376 Mashups

[6] ProgrammableWeb API http://api.programmableweb.com/

Several information fields are captured, but at the moment only API names and tag names are intended to be used to describe the combination of each mashup. For example:

mashup_id, API_1, API_2, ..., API_N
2005-11-20T12:01:39Z, Upcoming.org, Flickr, AmazoneCommerce, del.icio.us,
mashup_id, Tag_1, Tag _2, ..., Tag_M
2009-07-06T00:35:03Z,news,messaging,photo,microbloggin

3.4. Research Method

Following the approach of exploratory data analysis this research takes methods from the network theory to network science including social network analysis – SNA. These two techniques are complementary and offer a variety of metrics to evaluate dynamic networks. However, some aspects of evolutionary innovation of the mashup ecosystem are only captured through the phylogenetic analysis[7] borrowed from bioinformatics. Figure 9 shows the main steps of the research and the computational methods used in the data analysis: scrapping, storage, alignment, filtering, and statistical analysis. Code is available upon request, and main functions are in the Appendix 2 and Appendix 3.

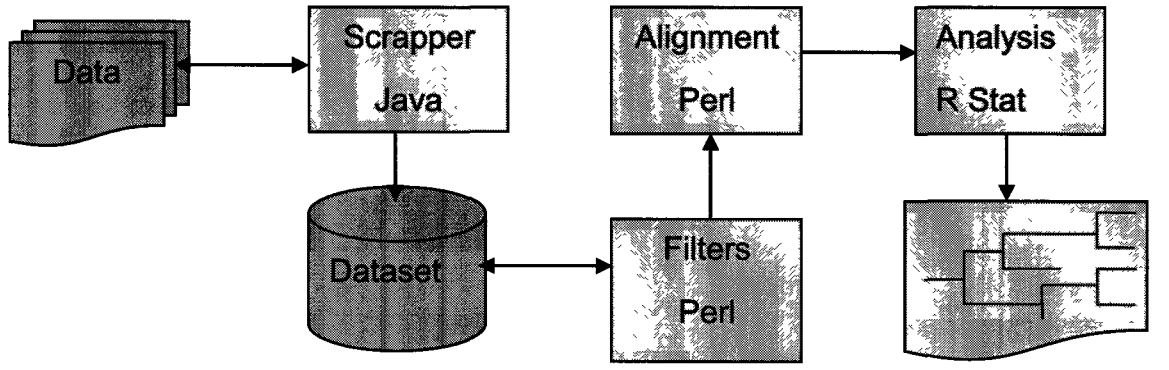


Figure 9 - Computational methods in the data analysis.

[7] Phylogenetic Analysis is the study of evolutionary relationships among organisms.

3.5. Research design

The following experiments are established to test the proposed models and emphasize the propositions presented in section 3.2.

3.5.1. Estimating Copying

To estimate the amount of copying that underline the mashup ecosystem, a simulation model of the evolution is created. This experiment simulates the growth model suggested in section 3.2.1 and calibrates the parameters according to actual data. The initial assumption is that each mashup has the same number (m) of APIs. Two parameters are defined; the ratio of APIs to mashups, $r = |A|/|M|$ which is obtained from the data on the ProgrammableWeb, and the proportion p of APIs among network nodes given by $p = |A|/|M \cup A|$. The copying factor is estimated by fitting the least square and power law methods to the mashup degree distribution. The least-squares approximation minimizes the sum of squared error which is the difference between the degree value observed and the fitted value provided by the simulation. The next method selects the copying factor that corresponds to the closest to the scaling exponent estimated by fitting the Zipf distribution to the actual degree distribution.

3.5.2. Estimating Phylogenetic Tree

The phylogenetic tree construction relies on several factors. Paradis (2006) discuses distance matrices and maximum likelihood methods. In addition to, other factors related to data format, clustering algorithm and transform functions are observed.

1) The modular format of mashups make it easy to represent is as a set of APIs. However, the order of APIs in a mashup, or even the order in the dataset, seems to affect the tree structures. Data pre-processing is implemented in Perl. It lists the

mashups in chronological order by its publication date. Also, it sorts the APIs by its utilization date in the dataset for each mashup.

2) Several similarity metrics from R-package ADE4 are tested and the Jaccard Index is the one fits better the dataset. The distance metric is taken as the complementary value of the similarity. Due the large size of the matrices, the similarity and distance metrics are implemented in Perl.

3) The clustering method BioNJ available in the R-package ape{bionj} adapts well the estimates obtained from the aligned mashups.

4) Certain functions require only dicthotomies in the tree structure, so the function ape{multi2di} eliminate all multichotomies.

5) Optimization of lambda parameter in the penalized likelihood method using function ape{chronopl}.

3.5.2.1 Time-Windows

Clade is a group of mashups within a specific radiation. In other words, clade or cluster is a group of mashups that share similar characteristics or descend from the same ancestor. To show different aspects over time, the dataset is fragmented in several time-windows with 500 mashups (approximately 6 months). Discrete and accumulative approaches are used to identify clades over the time. Discrete windows, $w_1,\ldots, w_{10}$, represent distinct subsets of the data. And, an accumulative window represent the total of all previous one, $w_{n+1}= w_n+500$. Thus, the sample w_7 has 3500 mashups.

Table 2 – Dataset time-windows showing the period of each discrete-window and size of each accumulative window.

w	1	2	3	4	5	6	7	8	9	10
period	05/09-06/02	06/03-06/10	06/10-07/02	07/02-07/06	07/06-07/11	07/11-08/05	08/05-08/11	08/12-09/05	09/05-09/12	09/12-10/07
size	500	1000	1500	2000	2500	3000	3500	4000	4500	5000

Two sliding window are tested to define the size of the data set - discrete and accumulative. The first subset only considers distinct time-window as specified in the Table 2. Each subset contains initially 500 mashups. The second data subset considers accumulative time-windows. That is, the first window contains 500 mashups and the second contains 1000, initially. In both cases, each subset is filtered to cover only the top-5 APIs (the most used and representing more than 80% of the dataset) eliminating noise in the tree and diminishing the computational processing.

3.5.2.2 Branching Times

The function chronopl{ape} estimates the node ages of a tree using a semi-parametric method based on penalized likelihood. The branch-lengths of the input tree are interpreted as mean numbers of substitutions, and they need to be greater than zero. It returns a tree where branch-lengths represent the node ages. A cross-validation technique is used to optimize the lambda parameter. Lambda → 0 means a saturated model with one distinct rate for each branch, and lambda → +inf means, a clocklike model with the same rate for all branches.

Having the tree properly calibrated, the function branching.times{ape} computes the distribution of the node ages that are normalized between 0 and 1. It is expected to have few old nodes and several young nodes, what may give raise a power law distribution. To test it, the function power.law.fit{igraph} fits a power-law distribution to the data, branching-times with maximum likelihood methods as recommended by Newman (2005). It returns the estimate value for the scale exponent. Clauset & Newman (2009) implement several functions in R commands to approximate power law distribution to data. For instance, loglogrsq estimates R^2 of the linear approximation to the double log of the distribution.

3.5.2.3 Number of Lineages

The mltt.plot{ape} function depicts the number of lineages observed on a tree which describes changes in the number of species through time. Using this plot to visualize the tree growth this experiment compares the growth between population (all mashups) and species (only the first published mashup as representative of its species). It represents 68% less mashups of the top-5 API subset. The goal here is to evaluate the role of identical species, or copy of the mashup blueprint.

3.5.3. Estimating Diversity

In bioinformatics, diversification rate is the difference between speciation and extinction rates (Eq. 4). It may cause confusion with diversity that at some level are related. For now, diversification rate is the parameter that indicates rise of distinct species. And the first step is to choose the diversification model that best fit the dataset. After the diversification rate is computed in different levels of the tree.

3.5.3.1 Diversification Model

This experiment intends to evaluate two diversification models namely birth-and-death and time-varying by fitting them in two phylogenetic trees, one representing the whole population of the mashups and other demonstrating only the unique mashups.

The functions colless.test and sackin.test{apTreeshape} test the tree balance on the Yule or PDA hypothesis based on the Colless or the Sackin statistic. The test consists of simulating *n* trees from a Yule process (n=500 by default) and comparing their indexes with the tested tree. In many cases the tested tree is less balanced than ones in the Yule model, so it is convenient to test this property in the subtrees using subtree.test{ apTreeshape}.

Birth and Death Model

Birth–death processes provide a simple way to model diversification. Because not all speciation and extinction events are observed through time the estimation of probabilities can be calculated fitting the birth-death model to the existing data. The birthdeath{ape} function fits by maximum likelihood a birth-death model to the branching-times computed from a phylogenetic tree. In addition, an extension of the Yule basic model allows computing multi speciation rates by using the function Yule-N-rate{laser}.

Time-varying Speciation and Extinction Models

Considering the framework for speciation and extinction rates (Eq. 6, Eq. 7, and Eq. 8) that vary continuously through time, the BOTHVAR model is more appropriated for the environment in study. The function fitBOTHVAR{laser} takes the branching-times derived from phylogenetic data, and use the parameters: x - net diversification rate; lam0 - initial speciation rate; k parameter of the exponential change in speciation rate; mu0 - the final extinction rate; and z- parameter of the exponential change in extinction rate.

3.5.3.2 Evolution Rates

Two other experiments are established to estimate evolution rates of the mashup species. The first experiment examines the ecosystem diversity by computing evolution rates for each time-window with incremental size and for selected clades involving the top APIs.

The second experiment analyses niche diversity by computing evolution rates for phylogenetic trees formed around the top APIs. A specific dataset is created for each niche including only mashup species. The objective in this experiment is to analyse complementary assets to the APY keystone.

4. RESULTS AND DISCUSSION

This section presents the results of the experiments established previously in section 3.5 and discusses the outcomes in order to answer the research question, how mashup ecosystem evolves under copying effects, and how it affects growth, innovation and diversity in the environment.

4.1. Results

4.1.1. Copying Factor

To test the model of the network growth by copying (3.2.1), the simulation model in section 3.5.1 optimizes network parameter by calibrating the simulated network with the actual data network. A detailed description of this experiment has been published in Weiss & Sari (2010). As example of the network simulation for a given combination of input m=2 and α=0.75, where *m* is the number of APIs in a mashup and α is the copying factor, Figure 10 depicts the network structure after 100, 500, and 2500 time-steps.

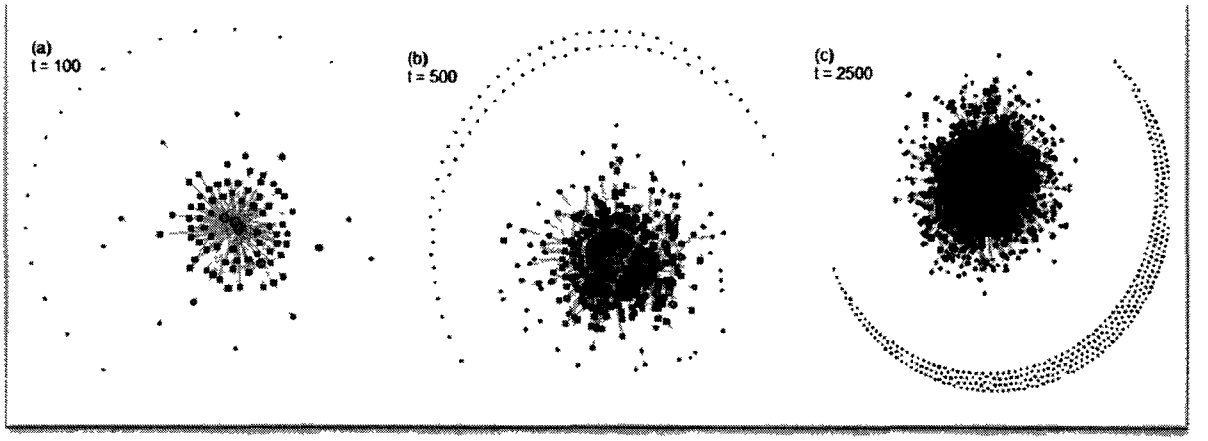


Figure 10 - snapshots of the simulated network after 100, 500, and 2500 time-steps, where red square represents mashup and green circle represents API.

To estimate the copying factor, the simulation is repeated for different values of α and *m* = 2. Figure 11 depicts two fitting models namely (a) the sum of square error where the best fit occurs for α=0.798, and the power law coefficient for the simulated network is 2.089, and (b) the power law fit error where the best fit occurs for α=0,855, and the power law coefficient for this simulated network is 1.986 with a 95% confidence interval of (1,983; 1,990). Figure 11 (c) and (d) compares ranked Zipf plot of the actual data distribution with the simulated data distribution. The slope for each plot (c) and (d) respective is -1.089 and -0,986.

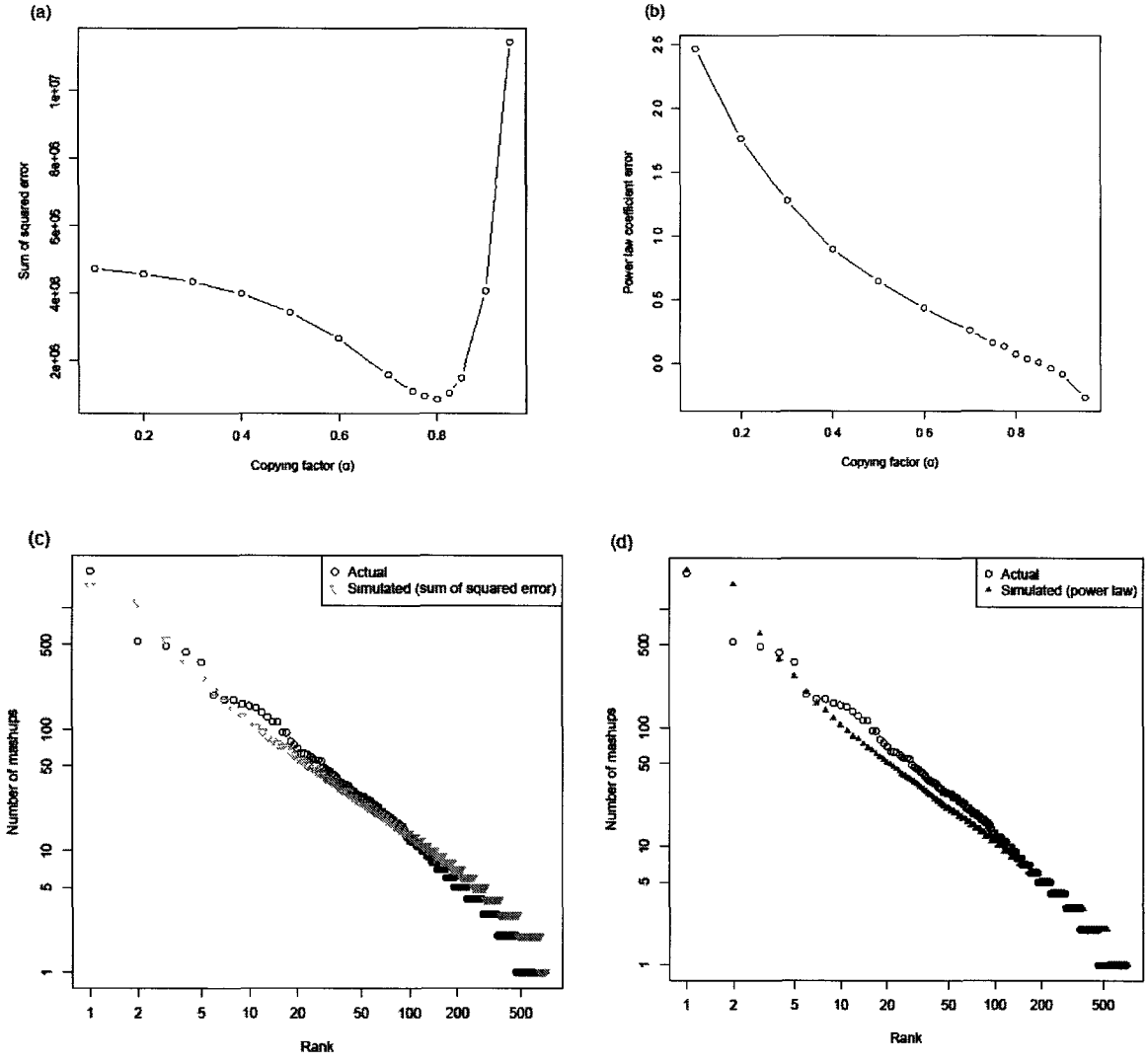


Figure 11 - Copying factor estimation by fitting the sum of square errors distribution (a) and power law distribution (b) to mashup degree distribution.

Table 3 – Comparison of the fitting methods with the actual data

indicators	SSE	Power Law
the number of mashups for the top-ranked API	under	over
the number of mashups for the second-ranked API	over	over
the number of APIs used by at least one mashup (actual 703)	1020	859
the number of APIs that contribute 50% of APIs (actual 12)	16	5

4.1.2. Tree Parameters

The temporal characteristic is kept in the dataset. Mashups are listed in chronological order. It is one of the reasons to use the mashup publication date as identity for instance 2006-03-31T01:12:19Z become 20060331011219 mashup ID. Each set of APIs – a mashup, is also chronologically sorted. Because API publication date is not available, the date of its first use is adopted as its identity.

To optimize the lambda parameter in the ape{chronopl} function the cross-validation function need to be minimized where "D2" correspond the effect of take out one lineage and l[i] the alternative lambdas, Equation 13 in R command.

cv[i] <- sum(attr(chronopl(phy, lambda = l[i]), "D2")). Equation 13

A large number is found to be appropriated to keep the tree balance. So, lambda=1e+05 is used in all transformations.

4.1.2.1 Time-Windows

Figure 12 shows GoogleMaps, Flickr, AmazoneCommerce, and YouTube as the most popular APIs; while YahooMaps, del.icio.us, 411Sync, eBay, and Last.fm appear only sporadically. And yet, Twitter, Facebook, and Twilio are the new ones. Phylogenetic trees of the windows 1, 5 and 10 are showed in the Figure 14, in which the node number and name of the top-5 API are identified. In (a) appears the strong presence of the

GoogleMaps clade, maybe half tree. The other half shows clades with complementary APIs to GoogleMaps – denoted by GoogleMaps+, and clades with APIs such as Flickr, del.icio.ous, Amanzon and YahooMaps that have descended by GoogleMaps+. Similar understanding occurs in (b) and (c); except that new clades appear as most used mashups.

The accumulative evaluation loses some API clades identified before such as 411Sync, Last.fm and Twilio. Figure 13 confirms GoogleMaps, Amazon, Flickr and YouTube as the most popular APIs, and Twitter as new clade. The tree structures also change over the time as depicted in the Figure 15. At each step, the tree estimation accommodates the new entrants by starting over the methods from the divergence matrix. What appears to be a solid area in the phylogenetic trees is the accumulation of combinations used for several mashups. In other words, combinations with similarity equal 1, or yet full copy of previous combination.

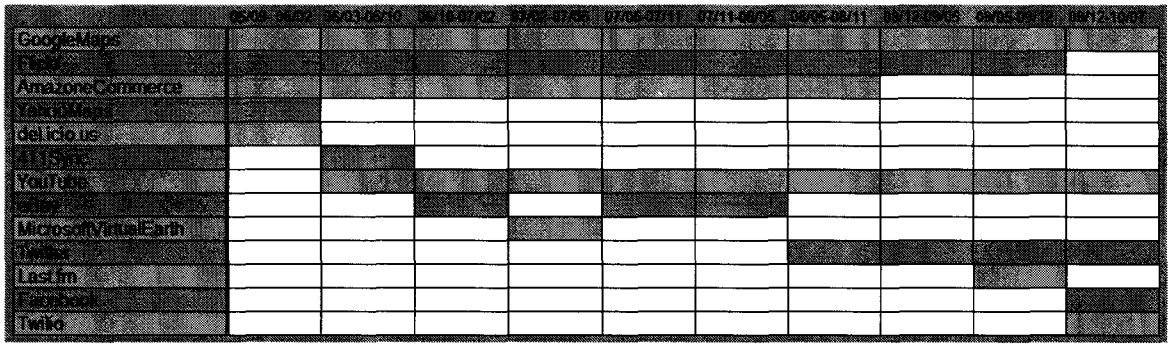


Figure 12- Chart of the top 5 APIs in the discrete time-windows

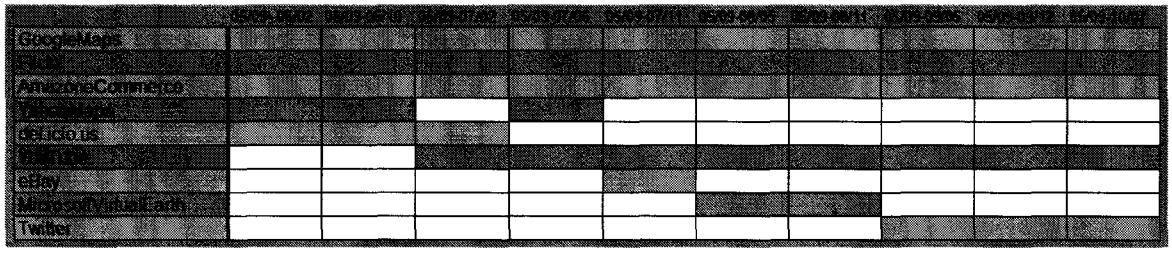


Figure 13 – Chart of the top 5 APIs in the accumulative time-windows

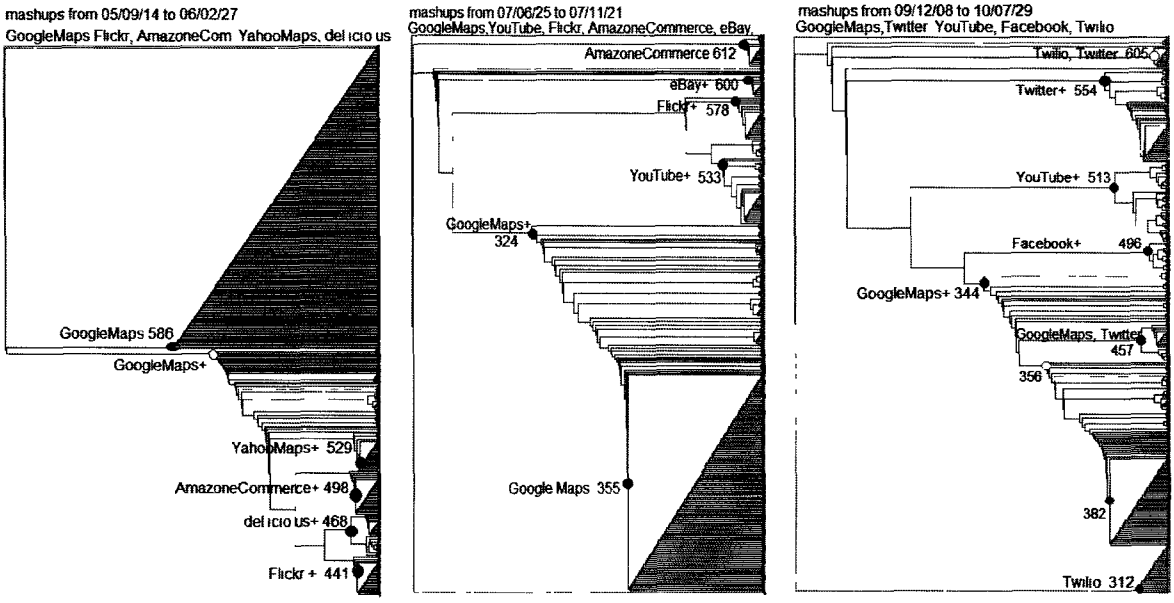


Figure 14- Phylogenetic trees of three time-windows with distinct data - beginning, middle, and end of the observed period: (a) w1 (05/09/14 to 06/02/27), (b) w5 (07/06/25 to 07/11/21), and (c) w10 (09/12/08 to 10/07/29). The red circle indicates the node number and name of the top-5 API.

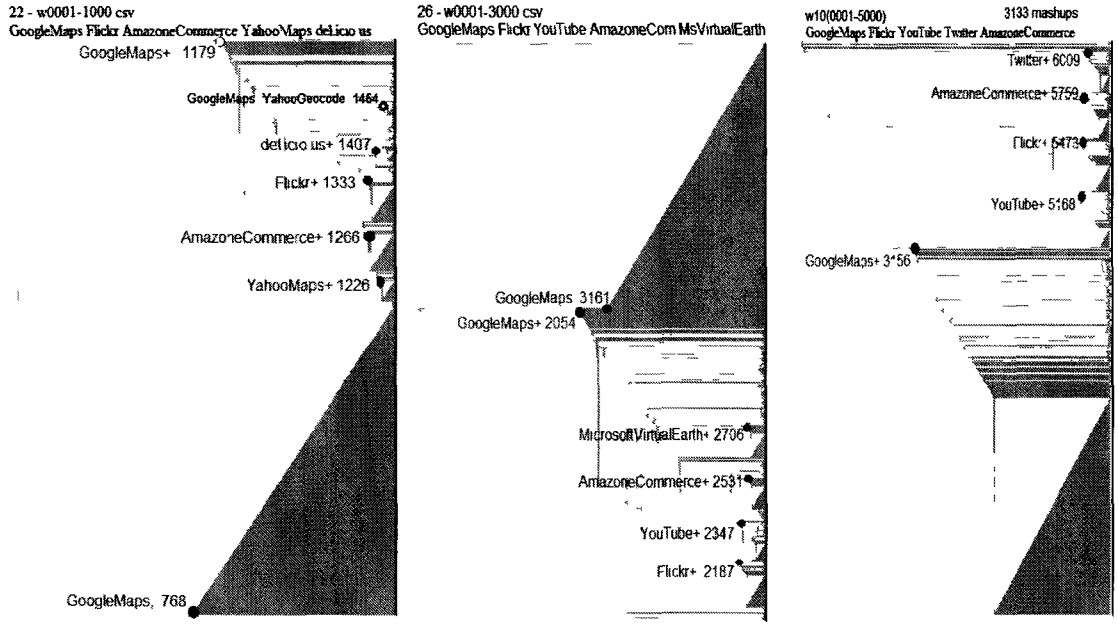


Figure 15 - Phylogenetic trees in three time-windows with accumulative data: (a) w2 (05/09/14 to 06/02/27), (b) w6 (05/09/14 to 07/11/21), and (c) w10 (05/09/14 to 10/07/29). The red circle indicates the node number and name of the top-5 API.

4.1.3. Innovation over Time

In the proposed phylogenetic tree framework, innovation process is reconstructed over time (3.2.2.1), and it can be modelled by the branching-times distribution (3.5.2.2). Figure 16 presents the branching-times distributions of the trees depicted in figures Figure 14 and Figure 15. In the first discrete time-window W1 the number of combinations increases only after some time (age <= 0.6), and then it rapidly increases linearly. In the middle of the period of analyse – W5, the number of old mashup species is larger than W1, and it increases almost linearly. Meanwhile, the branching-times distribution in the W10 tree changes at a slower rate. The accumulative approach (Figure 16b) describes the slow pace of the evolutionary innovation process. New species only appears after time < 0.6 for W2 and W6, and time < 0.4 for W10. Also, all tree distributions present linear intervals. Table 4 shows the power law fitting for all branching-times distribution with cutting-off equal to 0.4. The double log of each function is tested with linear model. W10 of the accumulative dataset has the best fitting since it has the largest loglogrsq.

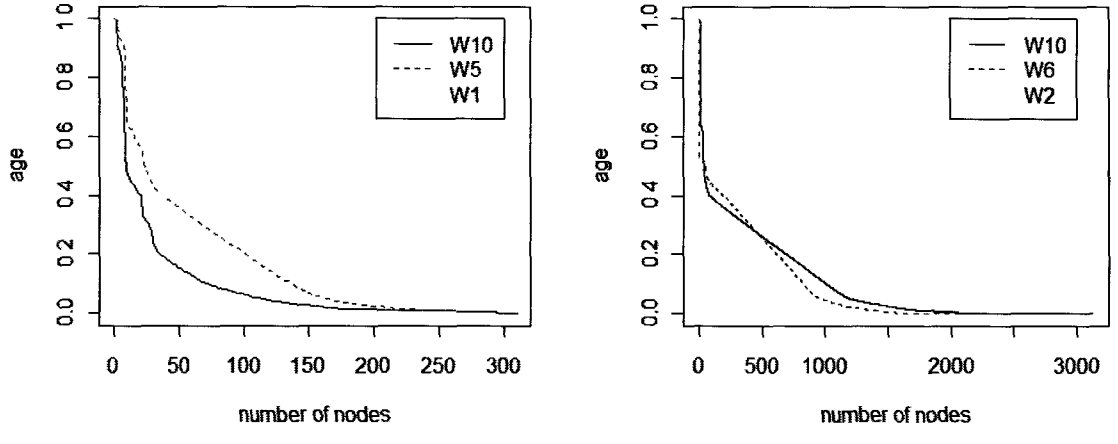


Figure 16 – Branching-times distribution of the phylogenetic trees discrete time-windows (w1, w5 and w10) and accumulative time-windows (w2, w6 and w10).

Table 4 – Power law fit to branching-times distributions

xmin=0.4	W1d	W5d	W10d	W2a	W6a	W10a
power.law.fit	2.233932	1.827962	1.840456	2.304773	2.333260	1.951463
loglogrsq	0.8485338	0.8719826	0.832979	0.8098579	0.8448398	0.9012946

4.1.4. Growth over Time

In the proposed phylogenetic tree framework, the ecosystem growth is reconstructed over time (3.2.2.2), and it can be modelled by the number of lineages distribution (3.5.2.3). Figure 17 depicts the number of lineages of the phylogenetic trees representing (a) all mashups of the population as individuals, and (b) only the first representative of the species. Each approach varies the time-windows (beginning, middle, and end of the period). Over the time the population not only increases the number of lineages, but also have early development when compared with the species curves. This difference is clear when comparing the log number of lineages over the reconstruction time in the population and species phylogenetic trees, Figure 18 where population tree is given by W10_pop and species tree is given by W10_spec.

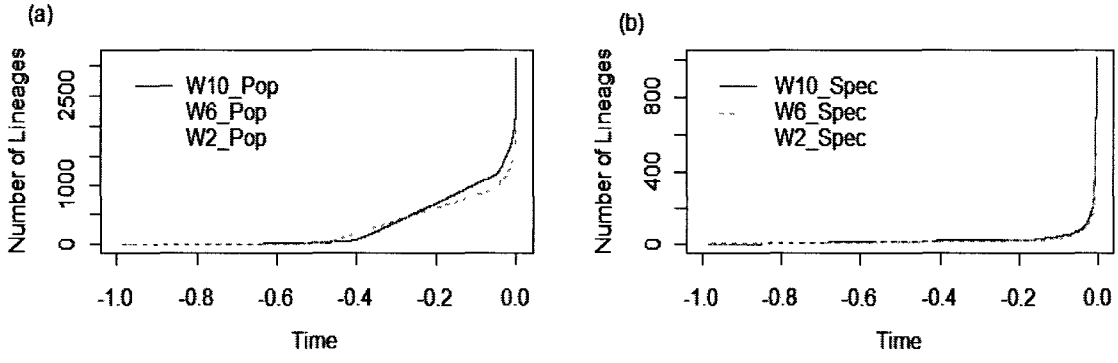


Figure 17 – Number of lineages of the phylogenetic trees including (a) mashup population, and (b) mashup species in three time-windows each.

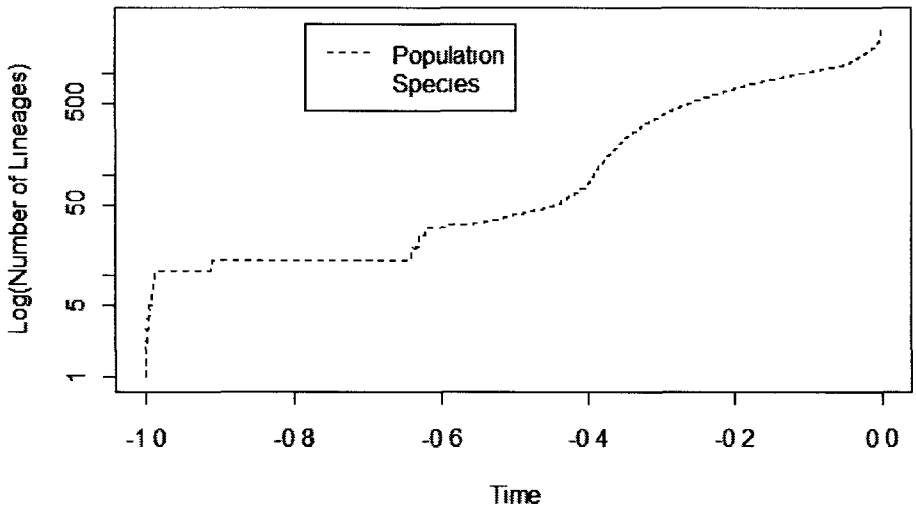


Figure 18 – Log number of lineages over the reconstruction time in the population and species phylogenetic trees.

4.1.5. Fitting Diversification Model

In the proposed phylogenetic tree framework, the diversity is the result of a birth-and-death process (3.2.2.3), so the first step is to find a diversification model that best fit the data (3.5.3.1). A Colless test indicates that the trees are unbalanced, since they are less balanced than the ones in the Yule model, results in the Table 5. However, the subtree test suggests that Yule model might fit in subtrees with specified size. The birth-and-death model estimates the parameterized diversification rate (b-d) and extinction fraction (d/b) as showed in the Table 7. Note that, these parameters for ctrPop are large than ones for ctrSpe. Furthermore, Table 8 presents the fitting results of the Yule-N-rate model which only estimates speciation rate. Table 9 describes the fitting results of the time-varying model with both variables – speciation and extinction. The best fitting is when the phylogenetic tree get the largest negative AIC[8]. Thus, the population tree has better

[8] Akaike's information criterion – AIC is a measure of the goodness of fit of an estimated statistical model, Akaike (1974).

AIC than species tree. Also, observe in Table 8 that the ctrPop's second speciation rate is high (r2 = 128.1935) and compare with the elevation in the second half of the ltt distribution in Figure 18. Moreover, note in Table 9 that the difference between initial speciation rate ($lam0 =73.25) and initial extinction rate ($mu0=54.73) is large.

Table 5 – Colless test for population and species phylogenetic trees.

tree	CIs is "less"	p.value	meaning
Population	953725	1	100% of the trees generated by Yule model have CIs and SIs less than the tested tree. Thus the tested tree is less balanced than predicted by the Yule model
Species	24246		

Table 6 – Subtree test for population and species phylogenetic trees.

Test of the Yule hypothesis based on the minimum number of subtrees of size	tree	size	stat	p.value
	Population	23	0.003	0.91
	Species	8	0.027	0.97

Table 7 - Fitting birth-and-death model to population and species phylogenetic trees.

Phylogenetic tree	ctrPop	ctrSpe
Number of tips:	3133	1014
Deviance:	-54580.75	-16886.31
Log-likelihood:	27290.37	8443.153
Parameter estimates: (b: speciation rate, d: extinction rate)	d / b = 0.9692115 StdE= 0.0033 b – d = 1.644086 StdE= 0.1672	d / b = 0.90637 StdEr= 0.0130 b - d = 4.58814 StdEr= 0.593
95% confidence intervals:	d / b: [0.9667974, 0.9714207] b - d: [1.533139, 1.762561]	d / b: [0.897469, 0.9144985] b - d: [4.216437, 4.989495]

Table 8 – Fitting Yule-N-Rate model to population and species phylogenetic trees.

tree	LH	st1	r1	r2	Aic
ctrPop	28054.34	0.003270954	6.950011	128.1935	-56102.69
ctrSpe	8657.589	0.04289973	4.374616	45.65266	-17309.18

Table 9 – Fitting Time-varying BOTHVAR to population and species phylogenetic trees.

tree	$LH	$aic	$lam0	$k	$mu0	$z
ctrPop	27404.18	-54800.36	73.25796	0.2914521	54.73573	53.10227
ctrSpe	9207.179	-18406.36	136.859	0.02185094	133.8999	42.41919

4.1.6. Ecosystem Diversity

In the proposed phylogenetic tree framework, the likelihood of diversity events increases when the extinction fraction is maximized, and so the diversification rate is minimized (3.2.2.3). The first experiment established in 3.5.3.2 computes evolution rates initially for the whole tree in each time-window and then for specific clades.

A new set of APIs takes place when examining mashup species (or unique mashups) as showed in the Figure 19. GoogleMaps, Flickr, and AmazoneCommerce remain constant over the time-windows, and YouTube and Twitter also remain constant since its appearance. Clades such as del.icio.us, YahooSearch and MicrosoftVirtualEarth happen sporadically. These fluctuations also can be seen in the tree structures exemplified in Figure 20. Figure 21 depicts evolutionary rates of the mashup species in 10 phylogenetic trees representing incremental dataset over time. Speciation and extinction rates are very close (left scaling axis), so its difference, diversification rate, is small (right scaling axis). Note that the u-shape of the diversification rate function describes a pattern of increasing and decreasing diversification. Analysing the whole dataset, including identical copies, the clusters are more visible in the shade areas of the tree, as showed in Figure 22. This affects the evolution rates which are depicted in Figure 24. Comparing with species evolution rates in Figure 21, the average of speciation and extinction rates decrease with negative diversification rate in the period of w2 to w6. From w7 the diversification rate is positive and greater than rates in the species tree.

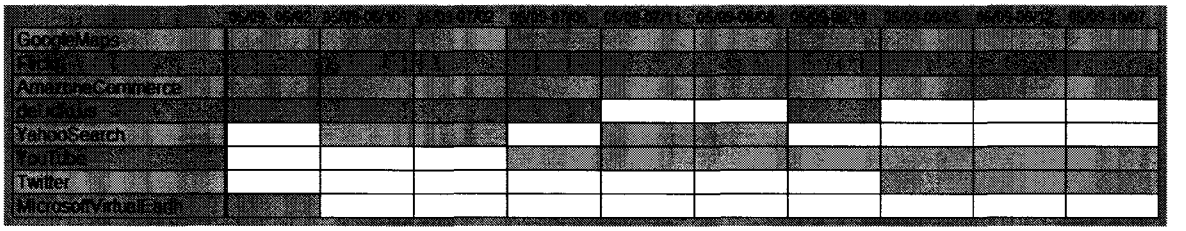


Figure 19 - Chart of the top-5 APIs for unique mashups in the accumulative time-windows

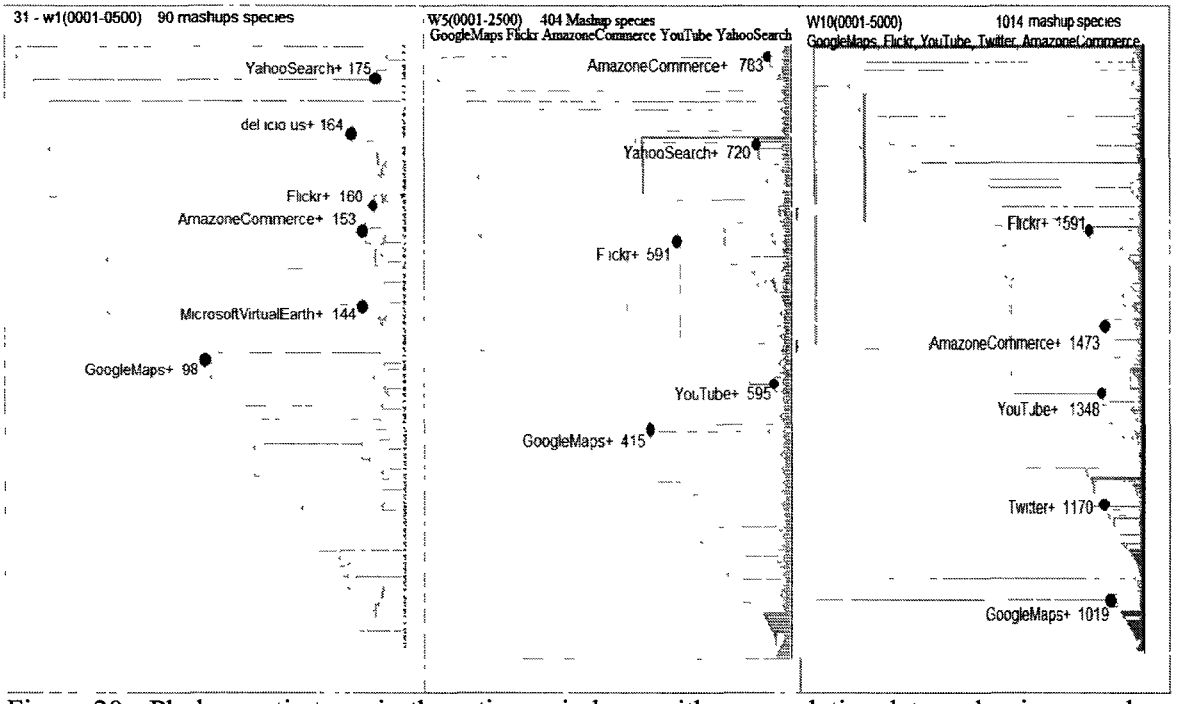


Figure 20 - Phylogenetic trees in three time-windows with accumulative data and unique mashup species: (a) w1 (05/09 to 06/02), (b) w6 (05/09 to 08/05), and (c) w10 (05/09 to 10/07). The circle indicates the node number and name of the top-5 API.

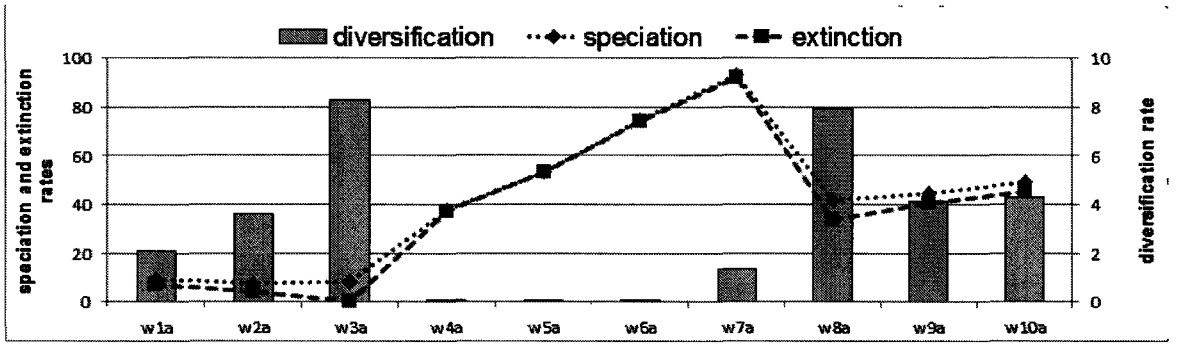


Figure 21 – Evolution rates of unique mashup species in each time-window phylogenetic tree with accumulative-size.

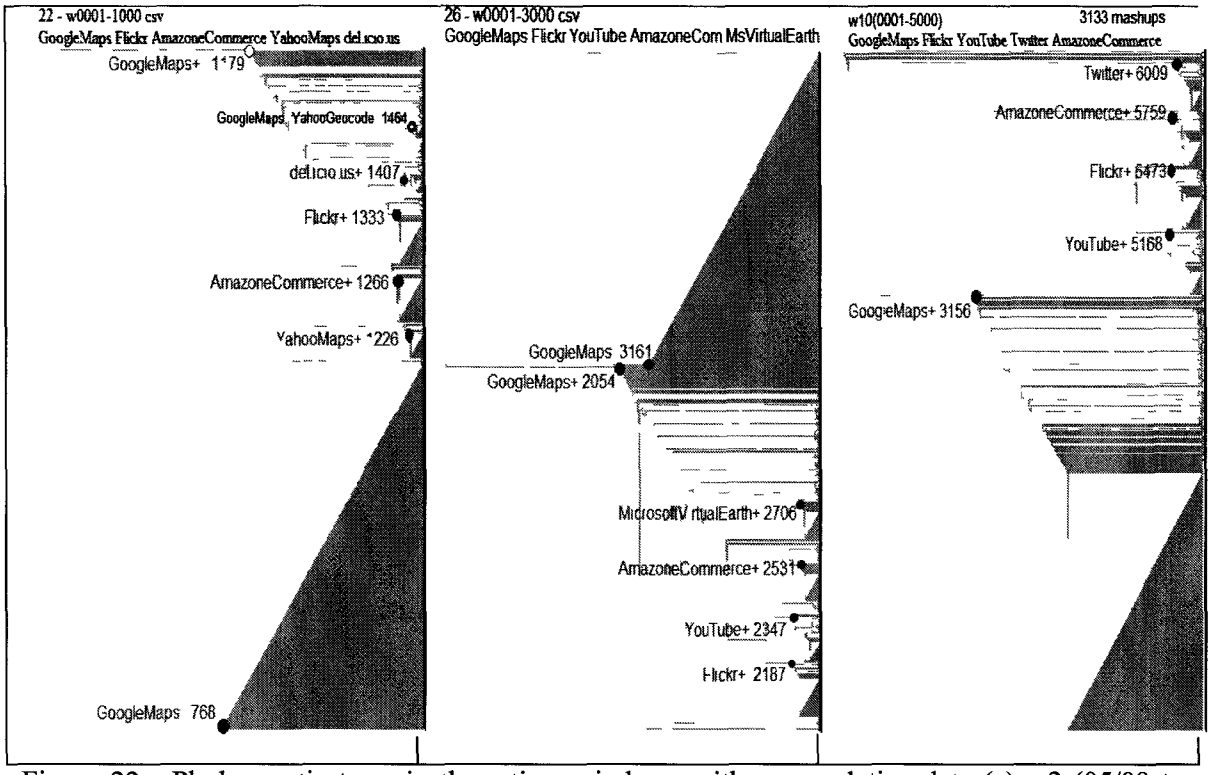


Figure 22 - Phylogenetic trees in three time-windows with accumulative data (a) w2 (05/09 to 06/10), (b) w6 (05/09 to 08/05), and (c) w10 (05/09 to 10/07). The circle indicates the node number and name of the top-5 API.

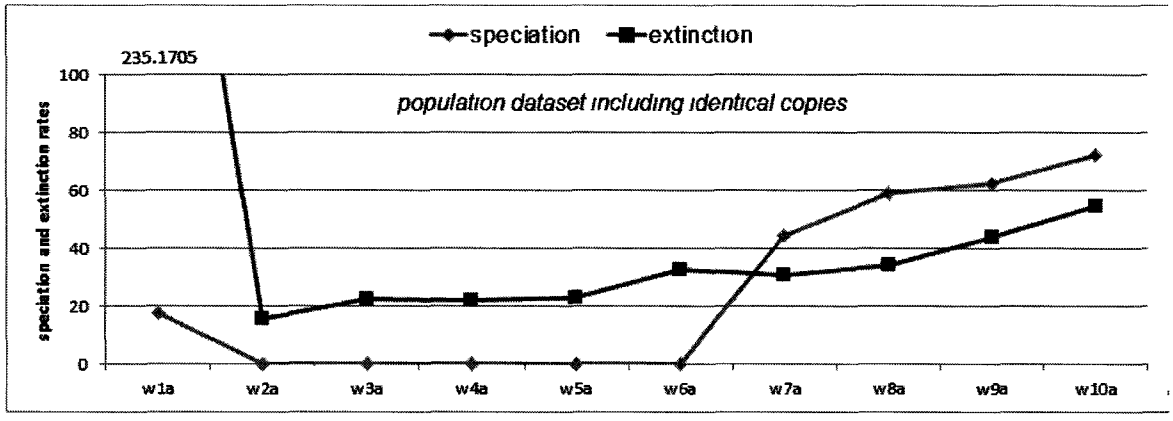


Figure 23 - Evolution rates of mashup (including identical copies) in each time-window phylogenetic tree with accumulative-size.

4.1.6.1 Clade Diversity

Large clades are created around each API technology, as showed in Figure 24. For example, Google Inc provides an interface to query locations on the earth for elevation data – GoogleMaps, and its clade includes complementary APIs such as Flickr, YahooGeocoding, YouTube and so on. Figure 25 displays evolution rates of mashup species in specific clades – GoogleMaps, Flickr, AmazoneCommerce, and YouTube, in the selected time-window phylogenetic tree. Two extreme points are identified. First, high diversification, that is high speciation and low extinction, is visible in GM10, Fk3, Ac3, Ac7, YT5. And second, low diversification, that is close speciation and extinction (<20), is noticeable in GM1-7, Fk5-7, Ac7, and YT7. Although each time-window phylogenetic tree has incremental number of mashups it does not mean that the clades follow this rule. For example, the clade size of the GoogleMaps increases through GM1 and GM8 and decreases in GM10. This variation is noticeable in the clade size of the Flickr and AmazoneCommerce.

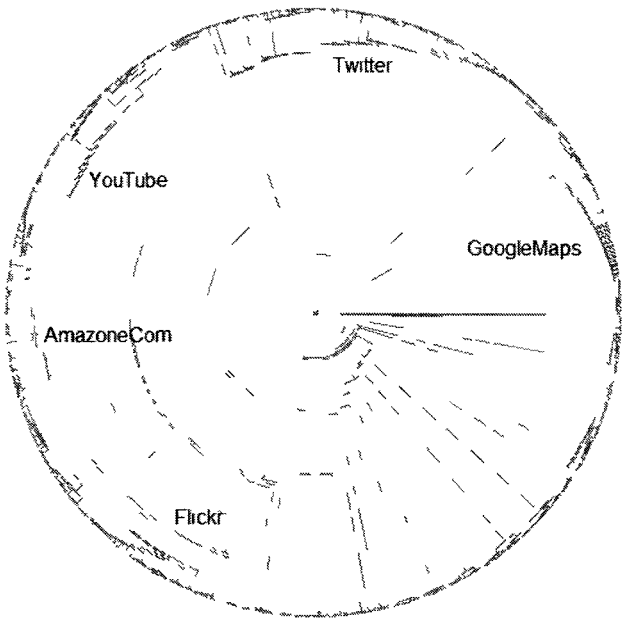


Figure 24 – Phylogenetic tree of the mashup ecosystem. The last snapshot showing the main niches namely GoogleMaps, Flickr, AmazonCom, YouTube, and Twitter.

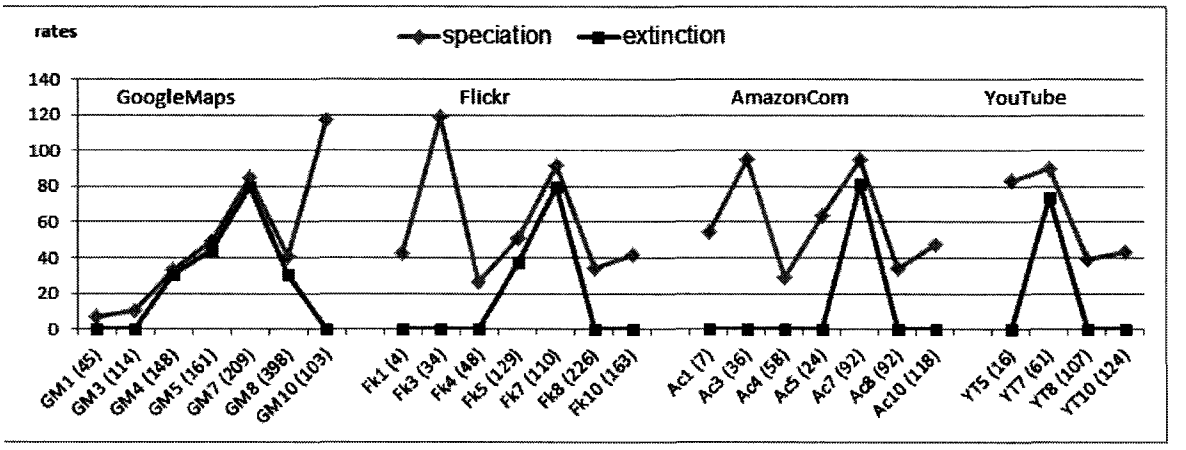


Figure 25 - Evolution rates of mashup species in selected clades of the selected time-window phylogenetic tree with accumulative-size (clade size in brackets).

4.1.7. Niche Diversity

In the proposed phylogenetic tree framework, the likelihood of diversity events increases when the extinction fraction is maximized, and so the diversification rate is minimized (section 3.2.2.3). The second experiment established in section 3.5.3.2 computes evolution rates for phylogenetic tree with one only top API.

This experiment describes these relations among APIs in the main niches of the mashup ecosystem as showed in Figure 26. Each tree is reconstructed from a data subset including only mashups that apply its API keystone. For example, the tree in Figure 26(a) represents mashup species using Flickr. All other APIs listed in the picture exemplify complementary assets in this niche. Each API name corresponds to the interface with high frequency in the clade. In some cases, this classification was not possible due to large number of different APIs.

Over reconstructed time, Figure 27 shows that the number of lineages is large for GoogleMaps niche followed by YouTube, Flickr and AmazonCom niches. The early combinations refer to APIs not yet defined in the major clades, for example Twitter in the GoogleMaps niche and Last.fm in the Flickr niche.

The results of the diversification analysis in these niches are described in Figure 28. Note that the graph includes three sequences representing speciation, extinction and diversification rates. Because speciation and extinction rate are very close one sequence overlaps another one; nonetheless the difference between them is shown in the diversification rate properly scaled in the second vertical axis.

Additional information is taken from Figure 26. Two distinct clades are formed, one in the bottom close to the API keystone, namely small radiation clade, and other above distant to the API keystone, namely large radiation. Moreover, small clades are formed in the top of tree, which are not defined yet. Table 10 – Clade radiation defining complementary relation between APIs. Table 10 describes these clades and computes the diversification rate for each clade in the API keystone radiation. Note that, the clades with low diversification rate are more heterogeneous than the other with high diversification rate. For instance, GoogleMaps, Flickr and YouTube present more diversity in the small radiation clades, while AmazonCom displays diversity in the large radiation clade. Also observe that, the niche around GoogleMaps has Google as the keystone, since it provides many other APIs such as Google Ajax and, GoogleSearch. And more interesting, Yahoo as Google's competitor has complementary role with YahooLocalSearch and YahooGeocoding in the GoogleMaps niche.

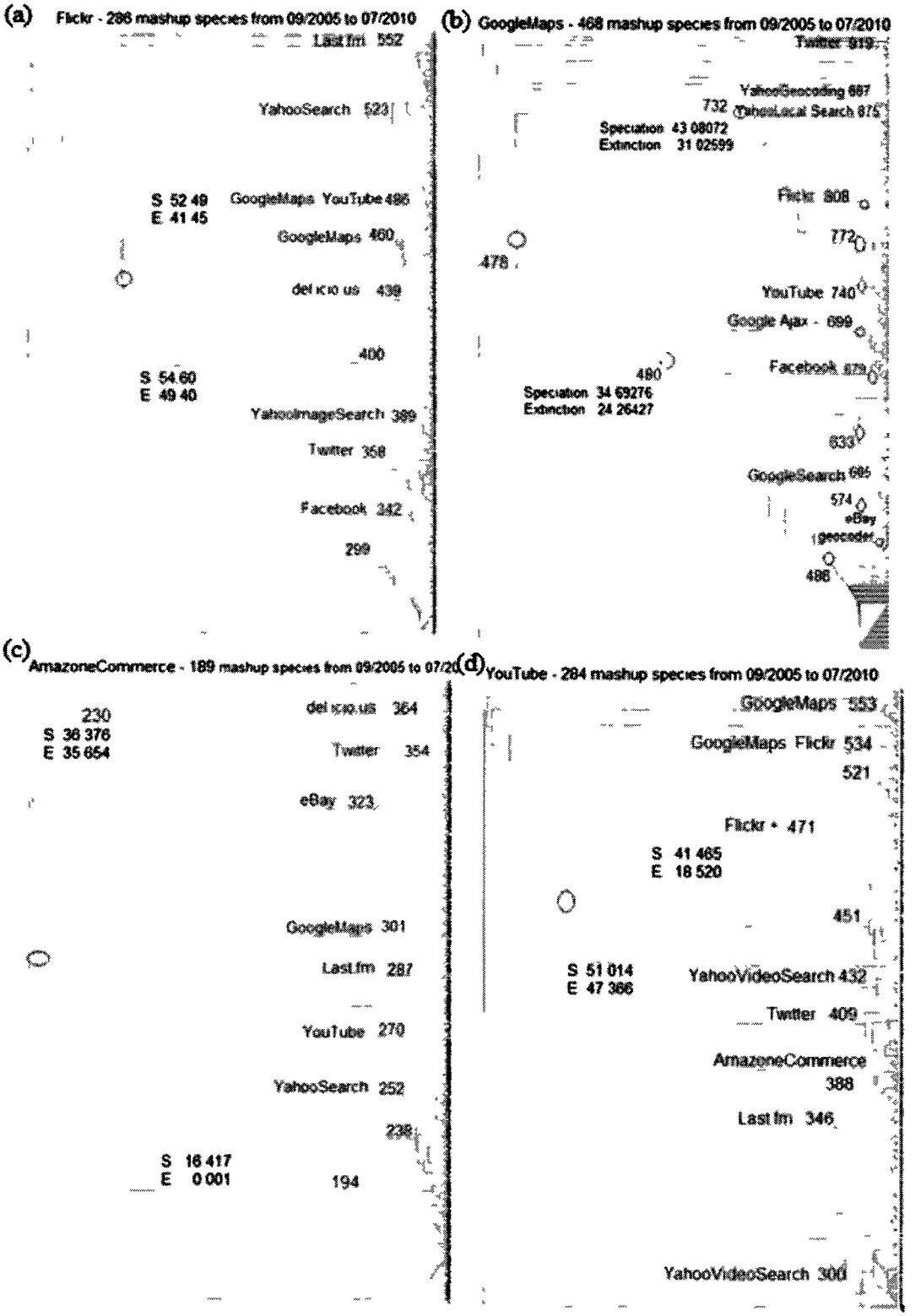


Figure 26 - Phylogenetic trees of the niches Flickr, GoogleMaps, AmazoneCommerce, and YouTube highlighting the most popular complementary APIs.

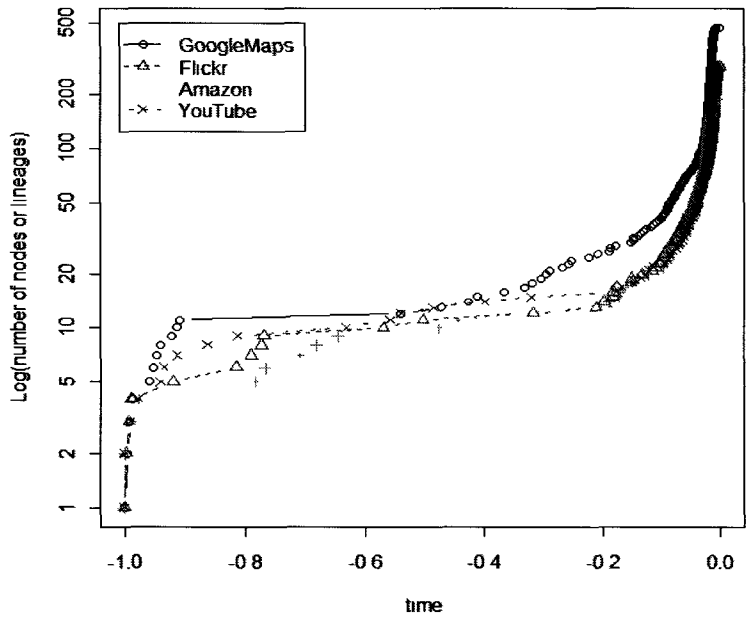


Figure 27- Age distribution and lineages distribution for Flickr, GoogleMaps, AmazonCom and YouTube niches.

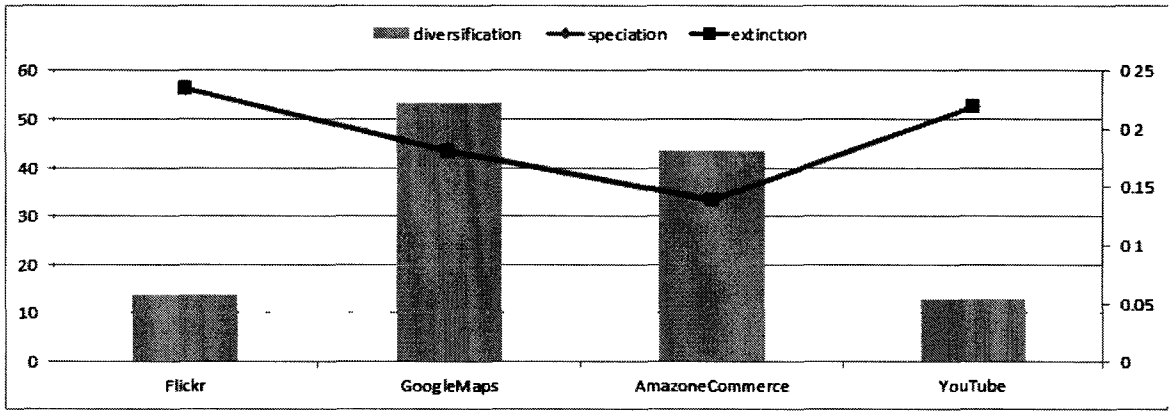


Figure 28 – Evolution rates – speciation, extinction and diversification, of the major niches Flick, GoogleMaps, AmazoneCommerce and YouTube.

Table 10 – Clade radiation defining complementary relation between APIs

Niche	*Small radiation*	*Large radiation*	*Not defined*
rate	r = 5.2	r = 11.04	
Flickr	Flickr	GoogleMaps	Last.fm
	Facebook	YouTube	YahooSearch
	Twitter		
	YahooImage		
	Del.icio.us		
Niche	*Small radiation*	*Large radiation*	*Not defined*
rate	r = 10.4285	r = 12.0547	
GoogleMaps	GoogleMaps	YahooGeocoding	Twitter
	Geocoder	YahooLocalSearch	YahooGeocoding
	eBay	Flickr	
	GoogleSearch	YouTube	
	Facebook		
	GoogleAjax--		
Niche	*Small radiation*	*Large radiation*	*Not defined*
rate	r = 16.416	r = 0.722	
AmazonCom	AmazonCom	del.icio.us	
		Twitter	
		eBay	
		GoogleMaps	
		LastFM	
		YouTube	
		YahooSearch	
Niche	*Small radiation*	*Large radiation*	*Not defined*
rate	r = 3.65	r = 22.94	
YouTube	YouTube	Flickr	GoogleMaps
	YahooVideo		
	Last.fm		
	AmazonCom		
	Twitter		

4.2. Discussion

4.2.1. Benefits of Copying

By using a copying process, end-user developers combine existing applications adding value to them. For instance, geographical coordinate maps by themselves do not mean much without additional data such as event location or statistical data. The resulting mashup application is really what brings innovation to their customer.

The process of constructing new ideas from existing ones shorts the learning curve by combining known elements in new ways, sharing of past experience across organizational boundaries, and the diversity of problem solving frames. According to recombinant innovation, end-user developers as knowledge brokers play collaboratively the role of the knowledge brokers to bridge between knowledge domains and reinterpret existing ideas in new contexts. Building on existing ideas is also the concept employed in open innovation. While an organization exploits its technology leveraging existing technological capabilities others organizations explore this technology, capturing and benefiting from external sources to enhance current technological developments.

The modular characteristic of mashups makes the recombination easy. The resulting performance depends on the ability of selecting interoperable modules and the diversity of available designs. Diversity of APIs implies high levels of dissimilarity among mashups and numerous combinatorial changes - the evolutionary innovation, that leads to ecosystem diversity. Copying is an important factor to API providers since it increases the network effects. Also, the recombinant approach helps API providers to identify complementary relations that reinforce their position in the innovation network.

4.2.2. Growth

The results in section 4.1.1 demonstrate the importance of copying in the network growth. Both fitting methods obtain the best fit for a high copying factor. This suggests

that most mashups are created by using an existing mashup as a template and modifying its design. In 79.8% to 85.5% of the cases, an API in the template mashup is copied to the new mashup; otherwise it is substituted by a different API. This indicates that copying plays a significant role in the evolution of the mashup ecosystem.

The indicators in Table 3 describe small discrepancies in the simulation results compared with the actual data. It can be fixed by increasing the number of APIs allowed to each mashup that now is restrict to two.

The results in section 4.1.4 show that the growth of the trees over the time (W2, W6, and W10) increases with identical copies in the population. Also, over reconstructed time, the number of lineages increases early (-0.4) in the population tree compared with the number of lineages (close to zero) in the species tree. In other words, population tree growth is variable with a large increment in the end (close to the present time), while species tree growth increase smoothly over time.

4.2.3. Innovation

The framework proposed in section 3.2.2 relies on similarity of organisms to describe the innovation in the ecosystem and its resulting diversity. Mapping the environment as phylogenetic tree allows reconstructing the evolution of the species from extant mashups. In addition, the research design describes other evolutionary aspect when taking data in progressive and accumulative number of mashups (section 3.5.2.1). The tree estimation performance depends not only on the clustering method, but also in pre and post considerations (section 4.1.2). In this research, the data is carefully structured chronologically, and the tree is chronologically calibrated to better fit the data.

The results in section 4.1.2 show that the set of top-5 APIs changes when the time-window approach changes, i.e. discrete or accumulative. However, there is a set of persistent interfaces such as GoogleMaps, Flickr, Amazon and YouTube. These APIs

become platform for many mashups, as it can be seen in the dark areas of Figure 14 and Figure 15. The triangle shape describes a set of mashup with high similarity, i.e. usually mashups using only one API. Over the time, the tree structure changes to fit the current dataset. Comparing these two pictures, Figure 15 highlights strong clades, but it loses small temporary clades such as YahooMaps, 411Sync, and Last.fm. Yet, it later detects the clades, for example YouTube appears in the second discrete window (Figure 12) and in the third accumulative window (Figure 13).

Evolutionary innovation is measured here by the pace of species appearance. An example is the W1 tree structure in Figure 13(a) and its branching-times distribution in Figure 16(a). These pictures describe the fact the ecosystem origin was centred in a few APIs and that it took some time to appear in the combinations of interfaces. Over time, the dynamic tree growth accommodates all species in a balanced structure, close to the ones in the Yule model. According to the Table 4, the W10a tree's branching-times distribution has better fit to the power law distribution, since its double log function approximates the linear function with scalar close to 1.

4.2.4. Diversity

4.2.4.1 Trees

Over time, the tree structure changes as new mashups are added, and it may increase or decrease the ecosystem diversity, i.e. the number of different clades. The experiment in section 3.5.3.1 tests two models, namely birth-and-death and time-varying evolution rates, by fitting them in two phylogenetic trees, one including the complete mashup population and the other only mashup species (unique mashups).

The test results in section 4.1.5 show that the population and species trees are less balanced than the ones in the Yule Model, and the subtrees seem to be more likely to have constant rates. The plot of the number of lineages over the reconstruction time in Figure 18 depicts the divergence in the second part of the distributions. The

diversifications models capture the discrepancy estimating speciation and extinctions rates with large variability. Comparing the three variations of the diversifications models in Table 7, Table 8 and Table 9, it seems reasonable to choose the one the computes both variables through time.

4.2.4.2 Clades

To identify clades, only unique mashup species are examined. The experiment in section 4.1.6 estimates evolution rates in each time-window tree, by progressively increasing the number of mashup species over the time. It keeps the same set of top APIs, GoogleMaps, Flick, AmazoneCommerce, and YouTube (Figure 19). These interfaces are easily identified in the adaptive structure of the phylogenetic trees over the time windows, as showed in Figure 20. The structure changes to better adapt new combinations of APIs, i.e. mashup species. In this process, clades are formed by speciation or extinction events. The main difference is whether or not the new species keep characteristic of its ancestor. Therefore, diversity is inferred to happen in the extremes of high and low diversification rate.

The tree diversification rate gives an indication of when the diversity happens, but it does not say where or which clade is more likely to grow by speciation or extinction of the traits, meaning APIs. The subtree analysis in section 4.1.7 presents the evolution rates in four major clades, as depicted in Figure 25. It might be acceptable to have high speciation in the first time-window trees, but some outliers need to be examined. For example, the last snapshot of GoogleMaps clade presents high speciation and low extinction. It could be the case when small clades move out of the large clade reducing the clade size. However, the small size of the GM10 is a mistake when selecting the root node that should be in the top of the tree, which it would be node number less than 1019.

4.2.4.3 Niches

Analysing the pace of innovation and growth of each niche over reconstructed time (Figure 27), it is inferred that the niches follow different innovation and growth processes. Branching-times and lineages distributions point GoogleMaps and YouTube as platform for many mashup species, while Flickr and AmazonCom cover specific domains.

Each niche has diversification rate close to zero meaning that the tree is growing through approximated rates of speciation and extinction. As much the extinction rate gets close to speciation rate as more diversity events happen. Because the construction process builds the tree left-right and bottom-up the API keystone normally appear in the bottom of the tree.

As the ecosystem structure the niche structures also evolve over time dynamically and adaptively. A snapshot of 2005 to 2010 period highlights four large niches, namely GoogleMaps, Flickr, AmazonCom and YouTube, and a fifth one just coming, i.e. the niche around the Twitter API. The identified diversity pattern reinforces the fact that API providers cooperate and compete in their own niches. Complementary assets with small radiation describe the close relationship of the APIs, for instance Flickr and YahooImageSearch. AmazonCom niche has evolved from mashups applying other APIs such as GoogleMaps and Flickr, but it does depend directly to them. Google assumes a position of dominator in the GoogleMaps niche not only for intense use of its API of maps, but also for the development of other APIs such GoogleSearch and GoogleAjax--.

5. FINAL CONSIDERATIONS

5.1. Summary of Contributions

This research presents an exploratory study and data analyses to answer the following research question: How does evolution happen in the mashup ecosystem, despite copying mechanism? This study offers several outcomes. First, a model of network growth by copying, based on social network analysis, is implemented as a simulation application and calibrated with actual data. Second a phylogenetic framework to estimate innovation, growth and diversity of the ecosystem is proposed, in which the following propositions are evaluated by applying phylogenetic analysis to the actual dataset:

Proposition 1 – The mashup ecosystem grows by copying and the proportion of copied APIs in a mashup defines the copying factor.

Proposition 2 - Phylogenetic tree reconstructs innovation processes, and it can be modeled by the scale-invariant branching-time distribution.

Proposition 3 - Phylogenetic tree growth describes the ecosystem growth, and it can be modeled by the scale-invariant distribution of the number of lineages.

Proposition 4 - A diversity event is a balance between speciation and extinction rates, and it can be estimated by minimizing diversification rates and maximizing extinction fraction.

The results are summarized as follow:

Modeling the mashup ecosystem as a network that connects mashups to API providers, and simulating this network growth by copying

The simulation of the network growth by copying, properly calibrated with actual data, provides the copying factor which indicates around 80% of API selection an API is

copied from an existent mashup. This high proportion reveals the significant role that copying plays in the evolution of the mashup ecosystem.

The mashup ecosystem evolution is reconstructed in a phylogenetic tree from the similarity of the mashups. The tree estimation is obtained by sorting and filtering data, and applying transforms to the tree structure. In particular, a large parameter is used in the Lambda transform to calibrate the tree structure to fit appropriately the dataset. It also transforms branch-lengths to branching-times specifying when the combination of APIs happened. Furthermore, the phylogenetic tree structures vary over time to adapt new or different mashups. Discrete time-windows detect early clade formation such as YouTube and Twitter, while accumulative time-windows highlight established clades such as GoogleMaps and Flickr.

Each branching-time is the age of the node representing the appearance of a new mashup species, called pace of innovation. The innovation process of the mashup ecosystem is modeled by the distribution of the ages, or simply branching-times distribution. This distribution follow the power law with scale exponent around 2, and the best fit happen when the dataset includes identical copies which converges earlier and faster than the distribution related to the mashup species subset. Therefore, it is inferred that copying increases the pace of innovation in the mashup ecosystem.

The mashup ecosystem grows as much as the number of APIs and mashup increases, and it can be seen in the phylogenetic tree growth. The number of lineages increases as much as the number of different API combinations increases. Therefore, the mashup ecosystem growth process is modeled by the distribution of the number of lineages. This distribution is derived from branching-times distribution, since in each reconstructed time the number of lineages is equal to the number of branches with that age. Thus, the results as similar to the branching-times distribution in terms of scale exponent and convergence. On consequence, it is inferred that copying is a mechanism of growth in the mashup ecosystem.

Because the number of lineages varies widely over reconstructed time, the changing proportion is different for each clade. Diversification models, commonly based on birth-and-death process, compute rates of traits kept in the new species, namely speciation, and proportions of traits lost in the new species, namely extinction. Time-varying diversification model fits better the dynamic and adaptive evolution process of the mashup ecosystem. In this model, both variables vary declining speciation rates and increasing extinction rates through time.

Diversification rate indicates diversity events, i.e. distinct clades, which result heterogeneity of the mashup ecosystem. Thus, diversity events happen when diversification rate is minimized. Ecosystem diversity evaluation points out a period of intense diversity (windows 4 to 7) in the occasion of YouTube appearance. It is also observed in the clade diversity evaluation which shows the diversification rate for four main clades, namely GoogleMaps, Flickr, AmazoneCommerce, and YouTube. Further investigation in the niches around these APIs describes an interesting pattern where two distinct clades are formed in each niche, one more and other less dependent on the API at the root of the tree. For instance, in the GoogleMaps niche, the clade with small radiation and small diversification rate presents more heterogeneous than the other clade with large radiation and great diversification rate. This emphasizes that small diversification rate indicates diversity in the environment. Additional insights can be taken from the niche diversity analyses in terms of the mashup blueprints and complementary assets, and it follows as recommendations to the stakeholders of the mashup ecosystem.

5.2. Recommendations

The proposed models in this research provide a valuable copying centric tool to evaluate the ecosystem, and different stakeholders can exploit it.

API providers can use this tool to sense and shape opportunities and threats. The mashup tree presents several new clades that might be opportunity for new developments. In established clades new entrants can easily be identified as complementary asset or a threat. Also, the provider can evaluate the API performance in the ecosystem. For instance, the model estimates network effects by computing clade size and counting identical copies of the mashup using the interface. Therefore, the proposed tree framework should be used as tool to identify dynamic capabilities which give essential support to API providers innovate and capture value from the innovation.

Directory providers have fundamental role in the mashup ecosystem by publishing the public APIs and mashups. Actual directories, such as ProgramableWeb, classify and display statistics of these applications. However, the combination of APIs and its popularity is not visible. Thus, directory providers should use the proposed tree framework to show the evolution and frequency of API combinations and the clade formation in specific domains.

Mashup development tools help user-developers to create their own application fast and easier by presenting friendly interface and listing available APIs. Nevertheless, the rationale behind the mashup design is yet a challenge. Hence, software developers should use the proposed tree framework to include templates of the mashup designs, and storage the new mashup.

End-users can use the proposed tree framework directly or indirectly to look at blueprints to select APIs that work well together. So, they can improve the mashup application by increasing the complexity and quality which brings more innovation to the ecosystem. Yet, they can use the same design to different domains creating new business opportunity.

5.3. Conclusions

To sum up, this research show the important role of natural copying mechanism which accelerates the innovation process, increases the growth, and drives diversity of the mashup ecosystem. The tree framework provides a copying centric tool which is an alternative approach to estimate and predict innovation pace and growth, and indicate diversity events. The tool recommends market strategies to API providers, publishing options to directory providers, development methodology to software developers, and guidelines to end-user developers.

5.4. Future Work

The simulation model for network growth by copying is limited to two APIs in each mashup. Thus, the model can be extended to increase the mashup size;

To estimate the phylogenetic tree a well know similarity metric was used and other basic metrics were tested. As suggestion, the similarity metric could include information about the API domain. For instance, a metric having not only API name, but also API classification as parameter of comparison.

Additional character information can be added as external trait, such as tag name, in order to cluster business areas. For example, the large GoogleMaps clade could be break down into travel_maps or event_maps, and so on.

Innovation and growth processes are modeled by the branching-times distribution which is scale-invariant. Thus, a simulation model can be created to predict the number of lineages (or paces of innovation) or the effects of identical copies in the mashup ecosystem.

Further investigation is required to estimate exponential change parameters for speciation and extinction rates in the time-varying diversification model in order to predict diversity in the mashup ecosystem.

One attractive work is to apply the framework to other domains. The only challenge is how to represent the character that defines the product or service. The methods remain the same with few parameter modifications. Modular character has structural advantage, for instance components of a software or hardware. In the open source software space, each component of the software could be represented by the sponsor organization. Yet, parts of a mobile phone could be symbolized by the supplier organization.

REFERENCES

Akaike, H. (1974). "A new look at the statistical model identification". *IEEE Transactions on Automatic Control* **19** (6): 716–723

Albert, R., Jeong, H., and Barabási, A.-L., 1999. Diameter of the world-wide web. *Nature* 401, 130-131.

Aldous, D. J. (2001). Stochastic Models and Descriptive Statistics for Phylogenetic Trees, from Yule to Today. *Statistical Science* 2001, Vol. 16, No. 1, 23–34.

Anderson, J. C., Narus, J. A., & van Rossum, W. (2006). Customer value propositions in business markets. Harvard Business Review, 84(3): 90-99.

Bailey, N. T. J. (1964). The elements of stochastic processes with applications to the natural sciences. Chapter 8 – Homogeneous Birth and Death Process. Wiley, New York.

Balasubramaniam, S., Lewis, G., Simanta, S., & Smith, D. (2008). Situated software: concepts, motivation, technology, and the future, *IEEE Software*, November/December 2008, 50-55

Barabási, A.-L. and Albert, R., (1999). Emergence of scaling in random networks. *Science* 286, 509-512.

Brandt, J., Guo, P.J., Lewenstein, J., and Klemmer, S.R. (2008). Opportunistic programming: how rapid ideation and prototyping occur in practice. WEUSE - *Proceedings of the 4th international workshop on End-user software,* publisher: ACM.

Brandt, J., Guo, P.J, Lewenstein, J., Dontcheva, M. and Klemmer, S.R.(2009). Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. CHI - *Proceedings of the 27th international conference on Human factors in computing systems,* publisher: ACM.

Brandenburger, A. and Nalebuff, B. (1996). Co-Opetition : A Revolution Mindset That Combines Competition and Cooperation. *Harper Collins Business*, 1997.

Brown, S. L., & Eisenhardt, K. M. (1995). Product development: Past research, present findings and future directions. Academy of Management Review, 20(2): 343-378.

Caldarelli, G. (2009), Scale-Free Networks, Oxford

Chakravorti, B. (2004). The new rules for bringing innovations to market. Harvard Business Review, 82(3): 58-67.

Chesbrough, H., (2003). Open Innovation: The New Imperative for Creating and Profiting from Technology. *Harvard Business School Press*, Boston, MA.

Chesbrough, H., Vanhaverbeke, W., & West, J., (2006). Open Innovation: Researching a New Paradigm. *Oxford University Press*, London.

Clauset, A. Shalizi, C.R. and Newman, M.E.J.(2009). "Power-law distributions in empirical data" *SIAM Review (*Society for Industrial and Applied Mathematics*)* **51**(4), 661-703 (2009). (arXiv:0706.1062)

de Nooy, W., Mrvar, A., & Batagelj, V. (2005), Exploratory Social Network Analysis with Pajek, Cambrdige

Eisenhardt, K M., & Tabrizi, B.N. (1995). Accelerating adaptive processes: Product innovation in the global computer industry. Administrative Science Quarterly, 40(1): 84-110.

Ethiraj, S.K., & Levinthal, D. (2004). Modularity and innovation in complex systems. Management Science, 50(2): 159-173.

Ferrier, W. J. (2001). Navigating the competitive landscape: the drivers and consequences of competitive aggressiveness. Academy of Management Journal, 44(4), 858-877.

Gans, J. S., & Stern, S. (2003). The product market and the market for "ideas": commercialization strategies for technology entrepreneurs. Research Policy, 32: 333-350.

Gamble, M. T., & Gamble, R. (2008). Monoliths to mashups, IEEE Software, November/December, 71-79.

Gascuel, O. (1997). BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular Biology and Evolution*, 14, 685-695.

Goldenberg, J., Lehmann, D. R., & Mazursky, D. (2001). The idea itself and the circumstances of its emergence as predictors of new product success. Management Science, 47(1): 69-85.

Haefliger, S., von Krogh, G., & Spaeth, S. (2008). Code reuse in open source software. Management Science, 54(1): 180-193.

Hartmann, B., Doorley, S. and Klemmer, S.R. (2008). Hacking, *Mashing, Gluing: Understanding Opportunistic Design*. IEEE - PERVASIVE computing, Jul/Sept 2008, pgs 46-54.

Hargadon, A. (2002). Brokering Knowledge: Linking Learning and Innovation, in: Research in Organizational Behavior, 24, 41-85.

Hoyer, V., & Fischer, M. (2008). Market overview of enterprise mashup tools, International Conference on Service Oriented Computing (ICSOC), LNCS 5364, Springer, 708-721.

Iansiti, M. and Levien, R., (2004). Strategy as Ecology. *Harvard Business Review* 82, no. 3 (March 2004).

Iyer, B., Lee, C-H., Venkatraman, N. (2006). Managing in a "Small World Ecosystem": Lessons from the Software Sector. California Management Review, 48(3): 28-47.

Iyer, B., & Davenport, T.H. (2008). Reverse Engineering Google's Innovation Machine, Harvard Busines Review, 86(4), 58-69.

Iyer B., Lee, C-H., Venkatraman, N., and Vesset, D. (2007). Monitoring Platform Emergence: Guidelines from Software Networks. Communications of the Association for Information Systems, Volume 19, 1-13

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 547–579. [http://en.wikipedia.org/wiki/Jaccard_index]

Krapivsky, P. L., and Redner, S., (2005). Network growth by copying, Phys. Rev. E 71 036118.

Kavanagh, E. (2010). Transforming Information Management with Enterprise Mashups. *Information Management* (1521-2912), 15212912, Jan/Feb 2010, Vol. 20, Issue 1.

Kleinberg, J.M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 604-632.

Kleinberg, J.M, Kumar, S.R., Raghavan, P., Rajagopalan, S., and Tompinks, A.S., 1999. The Web as a graph: Measurements, models and methods, *In Proceedings of the International Conference on Combinatorics and Computing*, no. 1627 in Lecture Notes in Computer Science, pp. 1-18

Krishnan, V.,and Ulrich, K. (2001). Product development decisions: A review of the literature. Management Science, 47(1): 1-21.

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tompinks, A.S. and Upfal, E., (2000). Stochastic models for the web graph. *In Proceedings of the 42st Annual IEEE Symposium on the Foundations of Computer Science*, pp 57-65.

Lee, S.-H., DeWester, D., & Park, S. (2008). Web 2.0 and opportunities for small business, Service Business, 2(4), 335-345.

Luksha, P. (2008). Niche construction: the process of opportunity creation in the environment, *Strategic Entrepreneurship Journal*, 2(4), 269-283

MacCormack, A., Verganti, R., & Iansiti, M. (2001). Developing Products on "Internet Time": The Anatomy of a Flexible Development Process. Management Science, 47(1): 133-150.

Magallon, S. & Sanderson, M. J. (2000). Absolute diversification rates in angiosperm clades. Evolution 55:1762-1780, 2001.

Nee, S., May, R. M. and Harvey, P. H. (1994). The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London*. Series B, Biological Science. 344:305-311.

Newman, M. E. J., (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46, 323. [arXiv:cond-mat/0412004v3 [cond-mat.stat-mech] 29 May 2006]

Miller, R., & Olleros, X. (2007). The dynamics of games of innovation. International Journal of Innovation Management, 11(1): 37-64.

Moore J. F. (1993). Predators and Prey: A New Ecology of Competition. Harward Business Review, May-June:75-86.

Moore J.F. (1996). The Death of Competition: Leadership and strategy in the age of business ecosystems, 297 pp. New York, NY, USA: Harper Business.

Moore, G. (2004). Darwin and the demon: innovating within established businesses. Harvard Business Review. 82(7/8). 86-92.

OConnor, G. & Ayers, A. A. (2005). Building a Radical Innovation Competency. Research Technology Management, *Industrial Research Institue, Inc*, Jan/Feb2005, Vol. 48 Issue 1, p23-31, 9p, 5

Paradis, E. (2006). Analysis of Phylogenetics and Evolution with R. *Springer Science LLC*, USA, 2006 ISBN-978-0-387-32914-7

Pirolli, P. and Card, S. K. (1999). Information Foraging. Psychological Review 106(4): 643-675.

Pisano, G., and Teece, D. J. (2007). How to capture value from innovation: shaping intellectual property and industry architecture. California Management Review, 50(1): 278-296.

Price, D.J. de S., (1965). Networks of scientific papers. Science 149, 510-515.

Rabosky, D. L. and Lovette, I. J. (2008). Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? Journal compilation 2008, The Society for the Study of Evolution, 62-8:1866-1875

Rothschild M. (1990). *Bionomics: The inevitability of capitalism*, 423 pp. New York, NY, USA: Henry Holt and Co.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology Evolution* (1987) 4(4): 406-425.

Sanderson, M. J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. *Molecular Biology and Evolution*, 19, 101-109.

Schumpeter, J. A. (1939). Business Cycles - A Theoretical, Historical and Statistical Analysis of the Capitalist Process. New York Toronto London : *McGraw-Hill Book Company*, 1939, 461 pp

Shenkar, O. (2010). Imitation Is More Valuable Than Innovation. *Harvard Business Review*, April 2010, pp 28-29.

Shuen, A. (2008). *Web 2.0 : A Strategy Guide*, 1st Edition, O'Reilly Media, Inc., April 17, 2008. Users Create Value, Chapter 1, 1-38 and Networks Multiply Effects, Chapter 2, 39-67.

Solé, R. V., Pastor-Satorras, R., Smith, E. and Kepler, T.B., (2002). A model of large-scale proteome evolution. *Advances in Complex Systems,* 5, 43-54.

Teece, D.J. (1986). Profiting from technological innovation: implications for integration, collaboration, licensing, and public policy. *Research Policy*, 15: 285-305.

Teece, D.J., Pisano, G. & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7): 509-533.

Teece, D.J., (2007). Explicating Dynamic Capabilities: The Nature and Microfoundations of (Sustainable) Enterprise Performance, Strategic Management Journal, 28: 1319–1350

Terwiesch, C. & Xu, Y (2008). Innovation contests, open innovation, and multiagent problem solving, *Management Science*, 54(9), 1529-1543.

Vanhaverbeke, W., Vrande, V. & Chesbrough, H. (2008). Understanding the advantages of open innovation practices in corporate venturing in terms of real options, *Creativity and Innovation Management*, 17(4), 251-258.

Vázquez, A., Flammini, A., Maritan, A. and Vespignani, A., (2003). Modeling of protein interaction reaction network. *Complexus* 1, 38-44.

Venkatraman, N. & Lee, C.H. (2004). Preferential linkage and network evolution: a conceptual model and empirical test in the u.s. video game sector, *Academy of Management Journal*, 47(6), 876-892.

von Hippel, E. (1986). Lead Users: A Source of Novel Product Concepts. *Management Science*, Jul86, Vol. 32 Issue 7, p791-805, 15p;

Watson, R. T., Wynn, D. and Boudreau, M. C. (2005). JBOSS: The evolution of professional open source software. MIS Quarterly Executive 4(3):329-341.

Watts, D. and Strogatz, S. (1998). Collective dynamics of 'small-world' networks, *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

Weiss, M., and Gangadharan, G.R. (2009). Modeling the mashup ecosystem: Structure and growth. R&D Management, 40(1), 40-49, 2010.

Weiss, M. and Sari, S., (2010). Evolution of the mashup ecosystem by copying. Mashups 2010 Proceedings, ACM. 4th International Workshop on Web APIs and Services Mashups. .....

Ye, Y. (2001). Supporting Component-Based Software Development with Active Component Repository Systems. Ph.D. Dissertation of the Department of Computer Science at University of Colorado.

Yu, S. and Woodard J. (2008). Innovation in the Programmable Web: Characterizing the Mashup Ecosystem. Second International Workshop on Web APIs and Services Mashups at ICSOC 2008.

Yu, J., Benatallah, B., Casati, F., & Daniel, F. (2008). Understanding mashup development, IEEE Internet Computing, September/October, 44-52.

Yule, G. U. (1924). A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis. *Philos. Trans. Roy. Soc. London Ser. B* 213 21–87.

Zang, N., Rosson, M.B., and Nasser, V. (2008). Mashups: Who? What? Why? CHI 2008 Proceedings · Works In Progress April 5-10, 2008 · Florence, Italy

Zang, N. (2008). Mashups for the web-active user. 2008 IEEE *Symposium on Visual Languages and Human-Centric Computing* (VL/HCC)

Zang, N. and Rosson, M.B. (2009). Web-Active Users Working with Data. CHI 2009 ~ Spotlight on Works in Progress ~ Session 2 April 4-9, 2009 ~ Boston, MA, USA

APPENDICES

Appendix 1 - R-Packages

Bortolussi, N. et al. 2009. Package 'apTreeshape' - Analyses of Phylogenetic Treeshape. 1.4-3, Date/Publication 2009-03-13 [http://cran.rproject.org/web/packages/apTreeshape/apTreeshape.pdf]

Chessel, D., Dufour, A-B., and Dray, S. 2009. Package 'ADE-4' Version 1.4-14 - Analysis of Ecological Data: Exploratory and Euclidean methods in Environmental sciences. Date/Publication 2009-12-09 [http://cran.r-project.org/web/packages/ade4]

Paradis, E. et al. 2010. Package 'ape' - Analyses of Phylogenetics and Evolution. Version 2.5-3, Date/Publication 2010-06-15 [http://cran.r-project.org/web/packages/ape/ape.pdf]

Harmon, L. et al. 2009. Package 'geiger' - Analysis of evolutionary diversification. Version 1.3-1, Date/Publication 2009-10-20 [http://cran.r-project.org/web/packages/geiger/geiger.pdf]

Rabosky, D. 2009. Package 'laser' - Likelihood Analysis of Speciation/Extinction Rates from Phylogenies. Version 2.3, Date/Publication 2009-08-30 [http://cran.r-project.org/web/packages/laser/laser.pdf]

Appendix 2 - R Scripts

The following R commands are examples of the main functions used in this research.

```
# ------------ Loading R-packages
library(graph, pos=4)
library(Matrix, pos=4)
library(ade4, pos=4)
library(ape, pos=4)
library(geiger, pos=4)
library(laser, pos=4)
library(apTreeshape, pos=4)
library(nnet, pos=4)
# ------------ Distance Matrix
 D <- as.matrix(read.table(file=dist_file, header = FALSE))
 M <- as.matrix(read.csv(file=id_file, header = FALSE))
      rownames(D)<- colnames(D) <- M
# ------------- Tree estimation
 tr <- bionj(D)
# -------------- Tree Transform
        dtr <- multi2di(tr)  # only dichotomies
        ctr <- chronopl(dtr, lambda=1e+05)  # Node Age  optimized lambda
        cbt <- branching.times(ctr);
# -------------- Number of lineages over time
```

```
mltt.plot(ctr1,ctr2, dcol = TRUE, dlty = TRUE, log="y")
# -------------Subtrees / Clades
plot(extract.clade(ctr,1099),"c",edge.color ="darkgrey", cex=.7)
ctrs <- as.treeshape(ctr)
colless.test(ctrs, model = "yule", alternative = "less", n.mc = 500)
subtree.test(ctrs, 23)
#------------------------ Birth & Death model
bd <- birthdeath(ctr)  # LASER
        div_rate1 <- bd$para[2] #speciation rate _S_ minus the extinction rate _E_
        ext_fraction <- bd$para[1]  #  extinction fraction _a_ = _E/S_
btv <- as.vector(branching.times(ctr)) ;
#------------------------ Time-Varying model
fitBoth <- fitBOTHVAR(branching.times(ctr), init=c(6)) #net diversity, lam0, k, mu0, z
        speciation <- fitBoth$lam0;    extinction <- fitBoth$mu0
#-----------------------------------------------------------------------------------------------------
```

Appendix 3 - Perl Scripts

The following Perl commands describe part of the code used in the data processing, in particular the Jaccard metric.

```
# -------------------extract of the main code
        my %data = ReadRawData("$datadir/$rawFile");
                %mashups = %data;
        my $hash_ref = \%data;          # only to print specific hash
        &DumpHashToFile($hash_ref, "$datadir/$dataFile");
        my %encodes = ReadApiIdFile("$datadir/$apiFile");
        my %chronos = SortAPIs();
        my %new_apis = ();
          %mashups = %chronos;
        # my %unique = Unique(); # no copy included
        #   %mashups = %unique;
        # my $N = 5;              # number of top APIs
        # my @top = FilterTopN($N);
        my @top = ("20050914152146");
        my %genus = mashupsUsingApis(@top);
          %mashups = %genus;
        my %dist = DistanceMatrix();
        DumpMatrixToFiles();
# -------------------compute the Jaccard Index
sub jaccard  {
        my ($i, $j) = @_;
        my @apis_i = split(" ", $mashups{$i});
        my @apis_j = split(" ", $mashups{$j});
        # print "i: @apis_i \n";
        # print "j: @apis_j \n";
```

```
        my %union =  ();
        foreach $a (@apis_i, @apis_j) { # put all items in hash table
                $union{$a}++
        }
        my @union = (keys %union);
        @union = grep /\S/, @union;
        my $union_size = @union;
        # print "Union Size: $union_size \n ";
        my %original = ();
    my @isect = ();
        map { $original{$_} = 1 } @apis_i;
        @isect = grep { $original{$_} } @apis_j;
        @isect = grep /\S/, @isect;
        my $isect_size = @isect;
        #print "Inter Size: $isect_size \n ";
        my $s = 1.0*($isect_size/$union_size);
        # print "similarity: $s \n";
        return $s;
}
#-------------------------------------------------------------------------------------------------------
```