1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file - Votre référence

Our file - Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Developing a Knowledge Based Decision Aid:
The Case of CASE Adoption

by

Rowland Phillips, B.Sc. Mathematics (Summa Cum Laude)

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the Degree of
Master of Management Studies

School of Business
Carleton University
Ottawa, Ontario
October 25, 1994

Copyright
1994, Rowland Phillips

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN
IRREVOCABLE NON-EXCLUSIVE
LICENCE ALLOWING THE NATIONAL
LIBRARY OF CANADA TO
REPRODUCE, LOAN, DISTRIBUTE OR
SELL COPIES OF HIS/HER THESIS BY
ANY MEANS AND IN ANY FORM OR
FORMAT, MAKING THIS THESIS
AVAILABLE TO INTERESTED
PERSONS.

L'AUTEUR A ACCORDE UNE LICENCE
IRREVOCABLE ET NON EXCLUSIVE
PERMETTANT A LA BIBLIOTHEQUE
NATIONALE DU CANADA DE
REPRODUIRE, PRETER, DISTRIBUER
OU VENDRE DES COPIES DE SA
THESE DE QUELQUE MANIERE ET
SOUS QUELQUE FORME QUE CE SOIT
POUR METTRE DES EXEMPLAIRES DE
CETTE THESE A LA DISPOSITION DES
PERSONNE INTERESSEES.

THE AUTHOR RETAINS OWNERSHIP
OF THE COPYRIGHT IN HIS/HER
THESIS. NEITHER THE THESIS NOR
SUBSTANTIAL EXTRACTS FROM IT
MAY BE PRINTED OR OTHERWISE
REPRODUCED WITHOUT HIS/HER
PERMISSION.

L'AUTEUR CONSERVE LA PROPRIETE
DU DROIT D'AUTEUR QUI PROTEGE
SA THESE. NI LA THESE NI DES
EXTRAITS SUBSTANTIELS DE CELLE-
CI NE DOIVENT ETRE IMPRIMES OU
AUTREMENT REPRODUITS SANS SON
AUTORISATION.

ISBN 0-315-98627-1

Canada

Name ROWLAND PHILLIPS

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

MANAGEMENT

SUBJECT TERM

0454

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0591
Asian 0305
Canadian (English) 0352
Canadian (French) 0355
English 0593
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Folklore 0358
Geography 0366
Gerontology 0351
History
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0615
International Law and Relations 0616
Public Administration 0617
Recreation 0814
Social Work 0452
Sociology
General 0626
Criminology and Penology 0627
Demography 0938
Ethnic and Racial Studies 0631
Individual and Family Studies 0628
Industrial and Labor Relations 0629
Public and Social Welfare 0630
Social Structure and Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture
General 0473
Agronomy 0285
Animal Culture and Nutrition 0475
Animal Pathology 0476
Food Science and Technology 0359
Forestry and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0817
Range Management 0777
Wood Technology 0746
Biology
General 0306
Anatomy 0287
Biostatistics 0308
Botany 0309
Cell 0379
Ecology 0329
Entomology 0353
Genetics 0369
Limnology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oceanography 0416
Physiology 0433
Radiation 0821
Veterinary Science 0778
Zoology 0472
Biophysics
General 0786
Medical 0760

EARTH SCIENCES

Biogeochimistry 0425
Geochemistry 0996

Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences
General 0566
Audiology 0300
Chemotherapy 0992
Dentistry 0567
Education 0350
Hospital Management 0769
Human Development 0758
Immunology 0982
Medicine and Surgery 0564
Mental Health 0347
Nursing 0569
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and Therapy 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences
Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0754
Mathematics 0405
Physics
General 0605
Acoustics 0986
Astronomy and Astrophysics 0606
Atmospheric Science 0608
Atomic 0748
Electronics and Electricity 0607
Elementary Particles and High Energy 0798
Fluid and Plasma 0759
Molecular 0609
Nuclear 0610
Optics 0752
Radiation 0756
Solid State 0611
Statistics 0463

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering
General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronics and Electrical 0544
Heat and Thermodynamics 0348
Hydraulic 0545
Industrial 0546
Marine 0547
Materials Science 0794
Mechanical 0548
Metallurgy 0743
Mining 0551
Nuclear 0552
Packaging 0549
Petroleum 0765
Sanitary and Municipal 0554
System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

The Undersigned recommend to the Director
School of Business acceptance of the thesis

"Developing a Knowledge Based Decision Aid: The Case
of CASE Adoption"

Submitted by

Rowland Phillips, B.Sc. Mathematics

In partial fulfilment of the requirements for the
degree of Masters in Management Studies

(Thesis Co-Supervisor)

(Thesis Co-Supervisor)

(Director, School of Business)

Carleton University
October 14, 1994

ABSTRACT

This research examines the CASE adoption decision process. A prototype of a knowledge based tool is developed to assist the adoption process.

The knowledge base is created through a combination of literature search and semi structured face-to-face interviews with individuals responsible for the adoption decision at organizations that have adopted CASE. These individuals serve as domain experts¹. The successful creation of the knowledge base, and the subsequent hybrid tool demonstrate that unstructured decision making, in general, and the CASE adoption decision, specifically, may be aided by a knowledge based approach.

¹A domain expert provides expertise which the knowledge engineer captures in a knowledge base.

ACKNOWLEDGEMENT

This thesis is dedicated to the memory of my father, the former Chief Justice of Jamaica, Sir Rowland Phillips, whose shoes I am still trying to fill.

I am forever indebted to my thesis committee of Professor Siva Pal, and Co- Supervisors, Professors Vinod Kumar and Uma Kumar who, often against my will, continually improved this document. Also, special thanks go to James Bowen who provided invaluable technical advice and criticism, and to Jean Blair who assisted in so many ways.

I would also like to thank those professionals who took time from their busy schedules and consented to speak with me, thus allowing the research to become a reality.

I must also thank my colleagues at the CARICOM Secretariat, particularly Mrs. Carol Collins, and my friends at CIDA, ACCC, and Universalia, especially David Grey, Louise Pesant, Margaux Beland, and Karen Christie who provided me this tremendous opportunity.

Neither can I overlook my family in Jamaica; Elinor, Hilary, and Geoffrey, who never waver in their confidence in my ability to achieve any goal, nor, my family in Guyana; Sherry, Kimberly and Randy, who made it more challenging and therefore more rewarding.

And finally, I must mention my mother, Lady Daphne Phillips, who instilled in me the desire to achieve academic excellence, a goal which I am still diligently pursuing.

TABLE OF CONTENTS

ACCEPTANCE.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES.....	vii
LIST OF APPENDICES.....	viii
 Chapter 1: <u>INTRODUCTION</u>	 1
1.1 Research Objectives.....	4
1.2 Research Approach	4
1.3 Organisation of the Thesis	5
 Chapter 2: <u>BACKGROUND</u>	 8
2.1 Computer Aided Software Engineering (CASE).....	8
2.2 Decision Making and Technology Adoption	12
2.2.1 Programmed Decision Making.....	16
2.2.2 Simon's Model of Non-Programmed Decision Making	17
2.2.3 Technology Adoption Decision	21
2.2.4 Rogers' Technology Adoption Model.....	23
2.2.5 Composite CASE Adoption Model	24
2.3 Expert Systems as Decision Aids	32
2.3.1 Decision Support Expert Systems.....	32
2.3.2 Expert Database Systems.....	39
2.3.3 Selection of ES Development Vehicle.....	43
 Chapter 3: <u>PROTOTYPE DEVELOPMENT - PHASE 1</u>	 45
3.1 Identification.....	48
3.1.1 Selection of Domain Experts.....	49
3.1.2 Preliminary Questionnaire Design.....	51
3.1.3 General Structure of Decision Process.....	52
3.2 Conceptualization.....	52
3.2.1 From Preliminary Questionnaire to Interview..	52
3.2.2 The Interviews.....	53
3.2.3 Modified Composite CASE Adoption Model and System Dependency Diagram.....	62

3.3	Formalization.....	68
3.3.1	Selection of Method of Representation.....	68
3.3.2	Selection of Shell.....	69
3.3.3	Resultant Rules.....	70
Chapter 4:	<u>PROTOTYPE DEVELOPMENT - Phase 2</u>	71
4.1	Implementation.....	71
4.1.1	The Knowledge Base.....	72
4.1.2	The Database Module.....	75
4.1.3	The Choice Module.....	76
4.2	Testing and Assessment.....	78
4.2.1	Assessment Criteria and Aims of the Prototype.....	81
4.2.2	Validation, Verification, and Testing.....	81
Chapter 5:	<u>DISCUSSION AND CONCLUSIONS</u>	85
5.1	Single vs. Multiple Experts.....	85
5.2	Composite Approach.....	86
5.3	The CASE Adoption Decision	88
5.4	Benefits of this Research.....	89
5.5	Limitations of this Research.....	90
5.6	Conclusions And Directions for Future Research	91
REFERENCES		95

LIST OF FIGURES

Figure	TITLE	PAGE
1	Research Approach.....	7
2	Architecture of an Integrated System Engineering Environment.....	11
3	Traditional and Modern Techniques of Decision Making.....	15
4	Composite CASE Adoption Model.....	27
5	Architecture of a Generic Expert System Shell.....	37
6	Rapid Prototyping (CASE Adoption)..	47
7	Modified CASE Adoption Model.....	66
8	System Dependency Diagram.....	67
9	Measures of Validation.....	80
10	Players in the Expert System Game..	114
 Sample Screens		
	Prototype Main Menu.....	122
	Knowledge Base Main Screen.....	123
	Hierarchical Model in Expert Choice	134

LIST OF APPENDICES

	Page
Appendix A: CASE: Tools and Environment.....	99
Appendix B: Expert Systems.....	108
Appendix C: Dependency Diagrams.....	115
Appendix D: Multicriteria Decision making and The Analytic Hierarchy Process.....	117
Appendix E: The Prototype.....	121
Appendix F: Preliminary Questionnaire	135
Appendix G: Interview Schedule.....	147

CHAPTER 1: INTRODUCTION

"Organizations increasingly rely on information technology (IT) both to perform their day-to-day operations and as a source of new products and services"

(Fichman, and Kemerer, 1993, p. 7).

The adoption of new technology, especially information technology is becoming vital to the survival of the firm. Historically, the popular focus has been on hardware developments, but, "in both the commercial and scientific markets, the rapid improvement in hardware capabilities (has) made software the critical path of product and systems development."²

Fichman and Kemerer (1993) maintain that the hardware/software imbalance has reached such proportions that it has been termed the "software crisis." And further, since Software production represents the single biggest obstacle to the successful use of IT in organizations; all precepts such as "using IT for strategic advantage", "reengineering the business", and "informating the work place" become mere slogans if the necessary software is not properly delivered on time.

In fact, a survey conducted by the Index Group Inc. identified "boosting software development productivity" as number 4 on its list of top ten 1991 management issues, and "improving software development quality" as number 7.

²HP CASE PRIMER An introduction to Computer Aided Software Engineering, Hewlett-Packard Company, 1989

Computer Aided Software Engineering, or CASE, is a software engineering process technology purported to solve or alleviate these problems. However, there is no free lunch, and there are significant costs associated with adopting this new technology. Robertson and Gatignon (1986) mention that high technology innovations are typically costly in monetary terms and have high switching costs. This is particularly true for CASE adoption since radical, difficult to reverse, changes to the software engineering process will occur. Also, there is uncertainty as to the consequences of adoption, as well as the realization of intended benefits.

Thus, firms are faced with the possibility of major benefits from adopting CASE, as well as disastrous results if CASE does not deliver. The situation is further compounded by a myriad of types of CASE and a plethora of vendors. As a result, a very complex, and important decision facing managers today is whether to adopt or not this new CASE information technology. Developing a tool to assist and thus improve the CASE adoption decision process is the focus of this research.

Decision making can be assisted by manual or computerized methods. Some manual methods are decision trees, decision charts, HIPO charts, and influence diagrams. Computerized methods incorporate most manual methods, and vary according to the type of decisions faced. Simon (1960, 1977) contended that decisions range from highly programmed to highly nonprogrammed. He maintained that highly programmed decisions

could be assisted by conventional programming methods, but that less structured decisions would be better aided by heuristic programs or expert systems as they are now known. The CASE adoption decision process has parts that are well structured, and parts that are not so well structured. For this reason among others, a hybrid³ prototype tool that incorporates expert system techniques as well as conventional programming techniques is developed. Decision support expert systems, and expert database systems are discussed more thoroughly in section 2.3.

Two decision making models; Simon's model of unstructured decision making, and Rogers' adoption model are discussed in some detail, and combined to form a composite model that provides a theoretical framework of the CASE adoption decision process, and a starting point for the prototype development.

The background for the research endeavour, will include knowledge from several disciplines. First, some description of what exactly is CASE is required to focus the discussion, next, some fundamental concepts of Decision Making theory are presented. Then, Technology Adoption , as a specific type of decision making, is introduced to establish the perspective of the research. And finally, since an ES prototype tool is to be developed, some Artificial Intelligence concepts are mentioned. The research, therefore, will involve three areas:

³In strict AI parlance hybrid refers to a mixed representation method, here, hybrid simply means mixed or combined.

Decision Making and Technology Adoption, Computer Aided Software Engineering (CASE), and Artificial Intelligence (Expert Systems).

1.1 Research Objectives

The first objective of this research, is to analyze the decision making process of technology adoption by the firm, and formally represent this knowledge in a knowledge base. A second objective is to investigate the realism by actually creating a prototype tool that can assist CASE technology adoption decisions.

Thus, two certain research opportunities addressed by this research are:

- 1) Eliciting and formalizing the technology adoption process in order to create a knowledge base that can assist the adoption/rejection decision process by a firm with regards to CASE technology, and;

- 2) Developing a rule-based, demonstration prototype hybrid expert system, which incorporates this knowledge base, to assist a firm in the CASE technology adoption decision.

1.2 Research Approach

The research is divided into two phases. The first phase involves the creation of a knowledge base which accurately represents the CASE adoption decision process. The CASE adoption literature, along with the composite CASE adoption

model provides a theoretical framework and a starting point for the knowledge elicitation process. Using a preliminary questionnaire initially, and then a semi structured interview schedule, the concepts involved in the decision process, and the relationships between concepts are elicited from practitioners at a sample of firms that have adopted CASE. These concepts are then formally represented in a number of IF-THEN production rules forming a knowledge base. This comprises the identification, conceptualization, and formalization stages of the ES development process as shown in Figure 1.

The second phase involves the creation of a demonstration prototype ES, using the knowledge base captured in Phase 1. This will comprise the ES developmental stages of implementation as well as testing and assessment.

1.3 Organisation of the Thesis

Four chapters comprise the remainder of this thesis. The following chapter explains some concepts of CASE tools, and outlines the underlying general theory of decision making as well as the specific technology adoption decision process which together provide a general theoretical framework. The chapter concludes with a brief treatment of expert systems as decision aids. Chapter 3 then depicts Phase 1 of the system developmental process in detail. Chapter 4 describes the activities performed in Phase 2 of the development process.

The final chapter, chapter 5, presents a brief discussion of various difficulties encountered during this research, mentions some of the benefits and limitations of this particular research, describes avenues for related future research, and draws conclusions.

Research Approach

Figure 1

CHAPTER 2: BACKGROUND

This chapter discusses the underlying concepts involved in this research as well as a review of the literature. First, CASE is defined, and the complexity and importance of the adoption decision is supported. Then, the decision making process is examined in greater detail, suggesting the use of expert systems as an aid. The chapter is comprised of three sections: CASE, Decision Making and Technology Adoption, and Expert Systems as Decision Aids.

Each section presents, albeit briefly, those theoretical concepts in the particular domain that are considered necessary for a general understanding of the area. These concepts are explained within the context of the particular research, and opinions by researchers in the area are alluded to.

For the sake of simplicity and clarity, some material is presented later in the appendices, and where even more detail may be required, references are provided.

2.1 Computer Aided Software Engineering (CASE)

Computer Aided Software Engineering, or CASE, encompasses all the computer-based tools that assist in automating software development and maintenance. In other words, CASE is software that is used to develop software.

Any discussion on software development should begin with

the traditional concept of a Systems Development Life Cycle (SDLC). Systems Development consists of four phases; Analysis, Design, Construction, and Maintenance. CASE tools attempt to automate the development cycle, and therefore address these four phases. CASE tools are often referred to as front end or back end tools. Front end tools will provide support for the analysis and design phases, and back end tools will assist construction and maintenance. Another common terminology is upper CASE and lower CASE, where upper CASE tools address analysis, and lower CASE tools facilitate design and code generation.

Holloway (1987) proposed three basic categories of CASE tools; Analyst Work-bench tools, System Generators, and Life Cycle tools. However two additional types of tools can also be identified; Integrated Project Support Environment, (IPSE) tools, open and closed, and Strategic Planning tools. A schematic representation of an integrated systems engineering environment is shown in Figure 2. These categories of tools are treated in more detail in Appendix A.

At present however, according to Braithwaite (1990) there are over 2000 CASE products in the marketplace, spanning the categories mentioned above, and this number is growing rapidly. Moreover, as CASE tools become more sophisticated and therefore more powerful, these categories are no longer so distinct. In fact, some practitioners refer to any automated productivity aid as CASE. Thus, the proliferation of CASE

tools and their continuing improvement have resulted in many varying perceptions of CASE, but for the purposes of this research, the traditional view of computer aided software engineering, and the categories mentioned above are used.

The purported benefits of CASE are varied, including productivity increases, increased control of the development process, quality improvements, and employee satisfaction, among others. However, the technology is not cheap. There are several different costs associated with the adoption of CASE, such as training costs, restructuring costs, and of course the high cost of the technology itself. As a result, the adoption decision is not straightforward, involving several factors, some of which are uncertain. Thus, CASE is an important emerging technology, but the adoption decision is complex, and fraught with risk, especially in the event that the purported benefits are not realized due to lack of organizational readiness, or perhaps no genuine need for CASE, or simply the selection of an inappropriate tool. Consequently, a closer look at decision making in general, and the CASE adoption decision, in particular, is warranted.

Architecture of an Integrated System Engineering Environment

Figure 2

Source: Prototyping (Vonk, 1990)

2.2 Decision Making and Technology Adoption

This section introduces two established models of decision making; Simon's model of non-programmed decision making, and Rogers' model of technology adoption. A discussion of programmed decision making is first presented, so as to identify where the CASE adoption decision falls on the continuum of decision making. Some specific attributes of the technology adoption decision are mentioned, and a composite CASE adoption model tailored specifically for the CASE adoption decision process is detailed.

Event versus Process

As early as 1910, John Dewey⁴ described three stages in problem solving: 1) What is the problem? (Problem Definition) 2) What are the alternatives? (Possible Solutions) and 3) Which alternative is best? (Selection or Choice). However, it was not until much later that Herbert Simon (1960, 1977) building on these concepts established his famous three stage decision making model: Intelligence, Design, and Choice. Prior to this, Dewey notwithstanding, decision making had been considered an event, but Simon convincingly proposed that it was a process. "Generally speaking, intelligence activity precedes design, and design activity precedes choice. The cycle of phases is, however, far more complex than this sequence suggests. Each phase in making a particular decision is itself a complex decision making process ... There are

⁴John Dewey, How We Think, D.C. Heath & Co., 1910, chap. 8.

wheels within wheels within wheels" (Simon, 1960, p. 3).

Simon distinguished between types of decisions. He maintained that decisions range from highly programmed to highly nonprogrammed. He described programmed decisions as "repetitive and routine, to the extent that a definite procedure has been worked out for handling them, so that they don't have to be treated *de novo* each time they occur." These decisions can be modelled by conventional computer systems, and are used extensively in the relatively exact domains of Operational Research and Management Science.

Nonprogrammed decisions are "novel, unstructured and unusually consequential. There is no cut-and-dried method for handling the problem because it hasn't arisen before, or because its precise nature and structure are elusive and complex" (Simon, 1960, p. 6). These decisions pose difficulties for conventional systems, and contemporary efforts to automate or assist these decisions have incorporated many concepts from the realm of Artificial Intelligence, such as heuristic programming, fuzzy logic, and restructurable modelling. Simon developed a framework that illustrates how the different types of decisions are treated. This framework is shown in Figure 3.

The technology adoption decision, which is the focus of this research, will fall somewhere between these two extremes. There are aspects of the decision such as selection of a tool that will fall on the fringe of programmed decision making,

and will lend itself to conventional methods, and there will be aspects of the adoption decision such as organizational readiness, and need for CASE that will fall in the nonprogrammed category and will be handled better by heuristic programs (expert systems). This argument is further developed in section 2.3 and a hybrid solution is recommended in section 2.3.2.

Traditional and Modern Techniques of Decision Making

	DECISION MAKING TECHNIQUES	
	Traditional	Modern
Programmed: Routine, repetitive decisions Organization develops specific processes for handling them	1. Habit 2. Clerical routine: Standard operating procedures 3. Organizational structure Common expectations A system of subgoals Well defined information channels	1. Operations Research: Mathematical analysis models Computer simulation 2. Electronic data processing
Nonprogrammed: One-shot, ill structured, novel policy decisions Handled by general problem solving processes	1. Judgement, intuition creativity 2. Rules of thumb 3. Selection and training of executives	Heuristic problem solving techniques applied to: a) training human decision makers b) constructing heuristic computer programs

Figure 3

Source: The New Science of Management Decision (Simon, 1977)

2.2.1 Programmed Decision Making

This type of decision making is relatively well understood. Some of the mathematical tools used are linear programming, dynamic programming, game theory, and probability theory. The situations modeled by these tools are myriad. Linear programming has been used to model refinery operations, dynamic programming can be used to model inventory or production planning situations, game theory has been used to represent marketing problems, and probability models have been used in a variety of circumstances (Simon, 1960).

The general procedures for using these tools are also relatively well understood:

- 1) Construct a mathematical model which accurately represents the management situation.
- 2) Define a criterion function which can be used to compare the relative merits of various possible courses of action.
- 3) Obtain empirical estimates of the numerical parameters in the model that specify the particular situation to which it is to be applied.
- 4) Determine which course of action maximizes or minimizes the criterion function.

Clearly, there are limitations to the situations which can be handled in this fashion. The situation should be structured enough that a mathematical model will closely represent it, and there must be some way of estimating the

parameters. Also, the size of the problem must be such that economically feasible solutions can be obtained.

Nevertheless, once the type of decision situation falls within the boundaries delineated above, it is widely accepted that conventional programs can be used either to automate or to assist these decisions.

2.2.2 Simon's Model of NonProgrammed Decision Making

Simon identified three stages of decision making: Intelligence, Design, and Choice (Simon, 1960, 1977).

The Intelligence stage is the first phase of the decision making process, where the environment is searched for conditions calling for decision. Intelligence is used in the military sense of the word referring to the gathering of relevant information. Basically, there are two stimuli which may trigger the decision making process: problem detection or opportunity seeking.

Whatever the stimulus - problem detection or opportunity seeking - the intelligence stage encompasses "collection, classification, processing, and presentation of the data necessary to prepare the ground for the later stages of the decision making process" (Ahituv and Neumann, 1990, p. 37). In terms of CASE adoption, this will correspond to identifying a need for CASE. Firms considering adopting CASE will have either productivity or control problems, or be seeking a competitive advantage, or a cost saving opportunity.

The second stage is design, which involves "inventing,

developing, and analyzing possible courses of action." This covers a wide range of activities, but generally, it means determining a finite set of alternatives worthy of consideration. Simon introduces a concept called "satisficing" which basically means that the search for alternatives is never exhaustive, and is concluded once aspiration levels are achieved, or constraints are exceeded. If no solutions are generated, then aspiration levels are relaxed, and the intelligence phase is revisited. This shows that the process is iterative. The concept of "satisficing" applies to CASE adoption in that no firm can afford in terms of time or money to consider all possible tools or solutions. In fact, Bouldin (1990) claims that due to the plethora of options, a search time of longer than two months will lead to confusion and inaction (non-adoption).

The third stage, Choice, involves selecting a particular choice of action from those identified in the design stage. Simon pointed out that many wrongly interpret this phase as the decision process rather than the culmination of the process. In reality, according to Ahituv and Neumann (1990), there are several difficulties that make Choice a complex stage. Some of these are:-

- 1) Multipreference - In most cases, the outcome is not measured by a single variable, but by multiple variables. There are always trade-offs. For example, customer satisfaction versus cost of

quality control. Sometimes the trade-offs are not easily measurable.

- 2) Uncertainty - In many cases, outcomes are uncertain, and one has to assign probabilities to various results. Is it better to select an alternative with low risk and mediocre returns, or one with high risk and high returns, such as CASE ? It depends on whether the decision maker is risk averse or risk seeking.
- 3) Conflicting Interests - Within any organization, there will be different interests or goals held by various decision makers. It is important to determine who makes the final decision. In the CASE adoption decision process, the presence of a project champion and his/her position in the organization is crucial.
- 4) Group Decision Making - Similarly, is the decision made by a group or an individual? Group decision making adds a level of complexity. The CASE adoption decision is likely to be made by committee, but any attempt to automate group decision making would be beyond the scope of this research.

Clearly Choice is not a trivial stage, and it may happen that no alternatives are considered viable, and the decision process begun anew. Indeed, the decision whether to adopt or

not is complex, and aiding this process is an aim of the research.

In effect then, Simon's model of nonprogrammed decision making is a general model of decision making that can be applied to almost any unstructured decision making situation including the CASE adoption decision. However, since Simon's model is not very descriptive, there are other decision making models more suited to the technology adoption decision process, which is a particular case of decision making and has its own concerns. Such a model is Rogers' Technology adoption model which is presented following a brief introduction to technology adoption in general. For a thorough treatment of other decision making models, see Saren (1984), Loo (1987), or Gordon et al., (1975)

2.2.3 Technology Adoption

Technology adoption is the process by which new products, processes, and services that utilise technology are acquired. In this particular context, "technology" and "innovation" are used interchangeably.

Robertson (1967, 1971) delineated three types of innovations: the continuous, the dynamically continuous, and the discontinuous. A continuous innovation causes little disruption in behavioral patterns and involves the introduction of a modified product or process. A dynamically continuous innovation causes some disruption in behavioral patterns, but does not alter them substantially; it may involve the creation of a new product or the modification of an existing one. A discontinuous innovation is a new product that requires the establishment of new behavioral patterns. In this research, the term "innovation" is synonymous with discontinuous innovation. CASE will fall in this category, as its adoption will require significant changes to existing conventional software engineering practices, with corresponding changes in existing behavioral patterns.

Rogers (1962, 1971) defines 5 attributes of innovations by which innovations can be described and compared. These are: 1) Relative Advantage 2) Compatibility 3) Complexity 4) Trialability, and 5) Observability.

Relative advantage is the degree to which an innovation is perceived as being better than the idea it supersedes.

Compatibility is the degree to which an innovation is perceived as consistent with the existing values, past experiences, and needs of the adopters. Clearly a discontinuous innovation will have low compatibility.

Complexity is the degree to which an innovation is perceived as relatively difficult to understand and use. This is of course a very relative or subjective concept, but not one that should be overlooked. Obviously, if a product or process is too difficult to use it will not be widely adopted.

Trialability is the degree to which an innovation may be experimented with on a limited basis. With expensive products, or comprehensive process innovations, it may not be possible to adopt on a trial basis. Nevertheless, many vendors are willing to lend CASE on a trial basis.

And finally, Observability is the extent to which the results of an innovation are visible to others. In cases where the benefits of the innovation are somewhat intangible like software quality, a value for observability will be difficult to determine.

These attributes or criteria will be used to assist in the final selection of a CASE product. This final selection or choice, is treated in more detail in the Implementation subsection 4.1.3., and is accomplished using *Expert Choice* and the Analytic Hierarchy Process.

Having clarified what is meant by an innovation, and introduced ways of comparing and differentiating between

innovations, it is time to consider some of the factors specific to the adoption decision.

2.2.4 Rogers' Technology⁵ Adoption Model

Rogers (1962) describes adoption of a new idea as a "bundle of related events flowing through time, in short, a process". Further, he maintains that the "adoption process is one type of decision making. It is a special type of decision making, however, with some attributes not found in other kinds of situations."

Rogers described the adoption decision as a process composed of five stages: 1) awareness, 2) interest, 3) evaluation, 4) trial and 5) adoption.

At the awareness stage, the individual is exposed to the innovation but lacks complete information about it.

At the interest stage, the individual becomes interested in the new idea and seeks additional information about it.

At the evaluation stage, the individual mentally applies the innovation to his/her present and anticipated future situation and then decides whether or not to try it.

At the trial stage, the individual uses the innovation on a small scale in order to determine its utility in his/her own situation. (with a process innovation, this stage may or may not be possible)

Finally, at the adoption stage the individual decides to

⁵Technology and Innovation here have the same meaning and are interchangeable.

continue the full use of the innovation.

Rogers concedes "that there is not complete agreement as to the number of stages in the adoption process, although there is general consensus on the existence of stages, and that adoption is seldom an "impulse" decision."

Simon's model and Rogers' model, while providing a general framework for the adoption decision are predominantly models of individual decision making. The CASE adoption decision process is unlikely to involve only a single decision maker, except at very small firms, and there are some organizational aspects such as authorization and lobbying that these models ignore. Also, there are some specific factors to the CASE adoption decision such as organizational readiness, and need for CASE that are not explicitly described by either of these models. For these reasons, a composite model specifically designed for CASE adoption is developed, by using Simon's model as an outline, and combining and adapting the stages in Rogers' model to the specific CASE adoption situation. This model serves as a theoretical framework of the CASE adoption decision process.

2.2.5 Composite CASE Adoption Model

Simon's model is a generic model that can be applied to any decision making situation, and therefore it is not very descriptive or specific. Rogers' model is more specific in that it describes the adoption process, which is a special type of decision making situation, but it is still a general

model in that it applies to adoption of any technology.

Thus, a composite model of CASE adoption that incorporates some features of the preceding models is presented. The model is developed by using Simon's model as an outline, then combining some of the stages in Rogers' model and tailoring it specifically for CASE adoption, using the CASE literature. For example, the awareness and interest stages of Rogers' model would correspond to the identification stage in Simon's model and to the needs recognition stage in the composite model.

The stages identified in this fashion are: Needs Recognition, Solutions Search, Solutions Evaluation, Organizational Readiness, Lobbying, and Authorization. A schematic representation of this model is shown in Figure 4.

Needs Recognition - The needs recognition stage is driven both by internal demands and external pressures. Common internal demands are requests for modernization, and use of new technologies by staff, and external pressures would include the need, for example, to achieve competitive advantage. McLure (1989) identified the major reasons for CASE adoption as:-

- 1) The organization has large backlogs of software systems waiting to be developed;
- 2) The organization spends the majority of its software personnel effort maintaining existing software systems;

- 3) The organization needs better discipline and control over the software development process;
- 4) The organization wants to increase software productivity substantially; and,
- 5) The organization wants to use state-of-the art software technologies.

Solutions Search - The next step after identifying a need, is to search for ways to satisfy this need. Loo (1987) identifies five types of search techniques. These are: Passive, Organizational, Professional, Trap and Active.

Passive search is the waiting for unsolicited suppliers to provide solutions.

Organizational search involves the use of past experience to identify suitable solutions or suppliers.

Professional search is the gathering of information by hiring consultants, attending conferences or trade shows, and reading professional magazines.

Trap search is to let potential suppliers know that the firm is looking for a certain type of solution. And finally, Active search means the direct seeking of solutions by site visit of users or suppliers until a suitable system is found.

Composite CASE Adoption Model

Figure 4

Solutions Evaluation - Next, the alternatives identified by the search stage must be evaluated. These solutions are compared using Rogers' attributes of complexity, comparability, etc..

Organizational Readiness - A related stage is assessing organizational readiness to adopt the proposed solutions. The major considerations in determining if the organization is "ready" for CASE are: whether adequate resources exist, in terms of trained staff and money, whether the software development process is sufficiently advanced, and whether managerial support exists in the form of a project champion, or a steering committee (Mclure,1989; Bouldin, 1990).

Humphrey (1988) identifies five levels of maturity through which the software development process at an organization may evolve. These are: 1) the initial level (anarchy), 2) the repeatable level (tribal folklore), 3) the defined level, 4) the managed level, and 5) the optimized level.

The characteristics of the initial level are adhoc procedures, little formalization, and tools are informally applied to process. This reflects an organization where anarchy prevails. That is, where programmers consider themselves creative artists not subject to rules. The best way to determine if an organisation is at level 1 is to observe procedures in a crisis. If in a crisis procedures are abandoned, and basically the organisation reverts to coding

and testing, then it is still at the initial level.

The repeatable level reveals a stable process with a repeatable level of statistical control. The activities required to move to the defined level are the creation of a process group, the establishment of software development process architecture, and the introduction of software engineering methods and technologies.

The defined level is depicted by a stable foundation for major and continuing progress. The activities required to move to the managed level are: establishing a process database, gathering and maintaining process data, and assessing the relative quality of each product and informing management.

The managed level shows substantial quality improvements, and comprehensive process measurement.

The ultimate level, the optimized level shows major quality and quantity improvements. Automatic gathering of process data is supported, and data is used to analyze and modify the process. (D.C.I., 1990, p. H-5)

Humphrey does not recommend implementing new software development technologies, such as CASE, unless the organisation is at least at level 3. This is a very conservative, or cynical view, as although maximum benefits may be derived with higher levels of sophistication, some benefits of CASE are still realized at less than optimum conditions.

Lobbying - Once the solutions are identified, and it is established that they are feasible for the organization, the political process of generating support for solutions begins within the firm. The "best" solution may not necessarily be selected.

Authorization - The decision process normally may be carried out at a level below the top executive. However, the recommended solution still will normally have to be approved by top management especially in the case of a process innovation.

Implementation - And finally, the approved solution will be implemented. Often there is a lengthy time lag between approval and implementation, and if the environment changes, sometimes implementation does not occur, and the process begins anew.

This composite CASE adoption model as depicted in Figure 4 serves as a theoretical framework, and a platform for prototype development. Once it is supported by practitioners in the field, it will be used as an outline in creating a knowledge base that can be captured in a prototype tool to assist the Case adoption decision.

It should be noted that the composite CASE adoption model is similar in many ways to Loo's general advanced manufacturing model (Loo, 1987), and could even be considered a specific instance of this general model, as CASE could be perceived as a software manufacturing technology. There are

also similarities to Mintzberg, Raisinighani, and Theoret's general model of the strategic decision process (Gordon et al., 1975), although this particular model attempts to capture the complexities of the political process in a changing environment and is thus more complex. However, the emphasis in this research is not a comparative analysis of various decision making models, but an attempt to identify the areas of the CASE adoption decision making process which can be automated, or assisted by a computerized tool.

2.3 Expert Systems as Decision Aids

Once the nature of the CASE adoption decision process has been determined, the next step is to decide how best this decision process may be aided. There are various decision making aids, both manual and computerized. Some examples of manual aids are decision trees, decision charts, HIPO charts, and influence diagrams, among others. In this research, the tool envisaged is a computerised tool. Also, because of the unstructured nature of parts of the adoption decision process, such as the identification of a need for CASE, in accordance with Simon (1977), the modern approach is the use of heuristic programming techniques, or expert systems, as they are currently termed.

This idea is developed further in the following subsection on decision support expert systems. Also, since there are some parts of the CASE adoption decision process, such as tool selection, that are fairly structured, a hybrid tool is suggested. This is treated in the subsection on expert database systems. The section concludes with a brief discussion of the development vehicle to be used in constructing such an expert system.

2.3.1 Decision Support Expert Systems

The British Computer Society defines an expert system, ES, as "the embodiment within a computer of a knowledge-based component from an expert skill in such a form that the machine can offer intelligent advice or take an intelligent decision

about a processing function. A desirable additional characteristic, which many would regard as fundamental, is the capability of the system on demand to justify its own line of reasoning in a manner directly intelligible to the enquirer" (Simons, 1985).

In other words, ES, represent an attempt to computerise certain processes, especially the decision making processes, that until recently were considered much too unstructured for machines. The five major types of expert systems are knowledge based information systems, decision support systems, consultation systems, problem solving systems, and coaching systems (Greenwell, 1988).

"Decision support expert systems take the advisory role of the expert. The system assists the user in the decision making processes by asking questions and interpreting the answers, and has a powerful role as an organisational aid to decision making" (Greenwell, 1988). The prototype created in this research acts as a simple decision support expert system, as it will, on receiving information, advise whether to adopt or not. [See Appendix B for other types of ES]

Some examples of decision support expert systems are FireCode, which assists designers to check if their designs meet safety standards, ADVISER, which is a manufacturing diagnostic expert system, and ASAA, which is a joint ABS/Fujitsu prototype system designed to assist in the checking of statistical output. For a thorough treatment of

these systems and many others, see Quinlan (1990).

Expert systems can also be classified by their sophistication and size. Waterman (1986) defines five stages of evolution of an expert system: Demonstration prototype, Research prototype, Field prototype, Production model, and Commercial system.

A demonstration prototype is a small demonstration program that solves a portion of the problem undertaken, suggesting that the approach is viable and system development is achievable. A typical rule-based demonstration prototype will contain 50-100 rules and take one to three months to develop. The decision aid created in this project is a demonstration prototype. [See Appendix B for details on larger systems]

A decision support expert system, like any expert system, is composed of a knowledge base, an inference engine, working memory, a user interface, and often a knowledge acquisition facility which allows a domain expert to impart his expertise directly. A schematic representation of these components is shown in Figure 5.

A knowledge base holds the knowledge about a particular topic in the form of Facts (e.g. John is a man) and Relationships (e.g. John is the father of Bob).

An inference engine uses the knowledge in the knowledge base to infer new knowledge by questioning the user and interpreting the appropriate rules of relationship.

The working memory is a data area for storing intermediate results generated by problem solving.

The user interface encompasses all the various methods of communication between the user and the computer system.

Knowledge acquisition, or KA, is the collection and analysis of information from one or more domain experts and any other sources leading to the production of a number of documents which form the basis of a functioning knowledge base. A substantial subset of knowledge acquisition is knowledge elicitation. Knowledge elicitation, or KE, is that area of knowledge acquisition which deals with acquiring information directly from domain experts.

In effect then, a decision support expert system is developed by accepting information from a domain expert, storing and representing this information in a manner accessible to the computer, and then utilizing this information via the inference engine to produce either advice, diagnosis, or other output depending on the system. These concepts are explained more thoroughly in Appendix B.

A major obstacle to the development of all expert systems, including decision support expert systems, is the problem of knowledge elicitation. It is an intricate task to extract from an "expert" the thought processes and steps used in solving a problem within his domain of expertise. Especially so, when in the case of managers and decision making, the decision process is often not clear. Various

knowledge elicitation techniques grapple with this elusive problem. There have even been recently, attempts to automate knowledge elicitation using induction, repertory grids, and documentation systems (Hoffman, 1990). In this research, knowledge elicitation is to be effected by the semi structured interview. This is treated in more detail in the conceptualization stage of prototype development.

Architecture of a Generic Expert System Shell

Figure 5

Source: Knowledge Engineering for Information Systems
Beynon-Davies, 1993, p. 36

Also, as Turban (1990) mentions, it is important to understand the different concerns or emphasis between knowledge engineering and conventional programming. Turban (1990) identifies several specific differences between knowledge based systems and conventional systems. These are:-

- 1) The knowledge base is clearly separated from the processing (inference) mechanism, whereas in conventional systems data and processing are combined in one sequential program. This has implications for maintenance as well as ease of creation.
- 2) Explanation is a part of most ES, whereas conventional systems do not usually explain why input data are needed, or how conclusions were drawn. This is especially important in this instance, as for management to have confidence in a decision aid, they would need to know how the system is arriving at its conclusions.
- 3) An ES can operate with only a few rules (as the first prototype), while conventional systems operate only when completed. This is important, because significant benefits can still be obtained from an early version prototype. With conventional programming, an incomplete system is of little value.
- 4) Expert Systems can operate with incomplete or

uncertain information, while conventional systems need complete information to operate. This is particularly important, as the decision process is not well understood, and the necessary information is rarely complete, yet successful decisions have to be made.

- 5) ES can easily deal with qualitative data, while conventional systems are suited to quantitative data. The adoption decision is based mainly on qualitative information.

For all of the reasons mentioned above, particularly the ability of an ES to handle qualitative data, an expert system tool is advocated. However, as is shown in the following subsection, a hybrid tool that is knowledge based, but enhanced by conventional concepts such as database management is ideal. The hybrid, expert system, demonstration prototype envisaged will assist the user in the technology adoption, decision making process by asking questions and interpreting the answers, and by taking the advisory role of the expert will have an important role as an organisational aid to decision making.

2.3.2 Expert Database Systems

"The term 'expert database system' has no precise meaning either for the academic computing world or for the computing industry. This is largely because the term is used to describe a gradually emerging, and constantly moving research

endeavour, rather than a developed software construct" (Beynon-Davies, 1991, p 3). Nevertheless, Beynon-Davies describes four different categories of expert database systems: 1) Expert System shells and Database management systems 2) Database systems and Expert systems 3) Knowledge engineering and Software engineering, and 4) Artificial Intelligence and Information systems development.

The demonstration prototype developed in this research falls in the first category which is the simplest and most limited area. This particular expert database domain can be described as that area which covers the connection of existing software tools such as database management systems with expert system environments to solve problems at the fringes of conventional information processing, which is exactly where the CASE adoption decision process falls.

Basically, this categorization can be considered as evolutionary stages. Initially, an expert system shell will be used to create a knowledge base which will interact in some elementary fashion with a database system. As the system develops into a commercial application, the database system grows and the expert system provides intelligent inferencing. Generally, some programming in Lisp or Prolog will be needed to facilitate the interface. The next stage occurs as expert system concepts are applied to the database system, and the system no longer resembles database software but is a truly integrated knowledge based system. The final stage is when

advanced Artificial Intelligence concepts are combined with information system engineering concepts to form 'intelligent' CASE tools. Should the tool created in this research continue to evolve, these are the stages through which it would move, and provides avenues for future research.

As mentioned, the focus in this research is creating a knowledge base that can assist the adoption decision process. However, to be practical and useful, the prototype should also be able to interface with a CASE product database so that a specific CASE product could be recommended. For this reason, a product database was incorporated into the prototype forming a hybrid or composite system, in fact, an expert database system. Although the interface is extremely rudimentary, the utility of this concept is clearly demonstrated and unaffected by the lack of sophistication. At this point, somewhat belatedly, some database concepts should be discussed.

Database Systems

A database may be defined as a "collection of structured data shareable between different parts of a organization's information system" (Beynon-Davies, 1991, p 7). Some characteristics of a database system are:-

- 1) Program-data independence - the application should not be affected by changes in the storage structure and access strategy of data (Date, 1986).
- 2) Data integration - A database should not have any unnecessarily duplicated or unused data (Howe,

1983).

- 3) Data integrity - When maintaining data, no inconsistencies should arise in the database.
- 4) Separate logical and physical views of data - The concept here is to separate the business view of data from its physical representation on some device.

A database management system (DBMS) is a pool of shared facilities used to access and maintain a database. A DBMS acts as the interface between end-users, application programs and the database. It allocates storage, provides security and handles all the traditional demands of file processing (Beynon-Davies, 1991).

There are primarily three types of databases, and therefore three database models: hierarchical, network, and relational.

The hierarchical model is a direct extension of file processing methods. Data is organized hierarchically in relationships of ownership (Beynon-Davies, 1991, p 9).

The network model extends the hierarchy concept into a network. Each entity or record within the database is joined to other relevant entities by a system of pointers. The major problem is that it is not easy for programmer or end user to 'navigate' through the database to extract information (Bachmann, 1973).

And finally, the relational model which is the most

popular, and is the one used in this research, organizes data in one uniform representation. Everything in a relational database is represented in the form of two-dimensional tables related together by common attributes (Beynon-Davies, 1991, p 9). The relational model has attained superiority because of its simplicity, and because of difficulties with the other two models. For example, the hierarchical model forces similar data to be stored in separate areas causing data redundancy, and inefficiency, and the network model which is an extension of the hierarchical model and thus more complex requires more expertise on the part of the user to extract information. For these reasons, a relational database is preferred.

Having decided on using a relational database, and combining it with a knowledge base to form an expert database system, the question remains how to develop the expert system.

2.3.3 Selection of ES Development Vehicle

There are two major alternatives in developing an expert system: One way is to develop a system from scratch using a programming language such as Lisp or Prolog; the other way is to build a system with an ES building tool called a shell. Using a programming language requires an ES developer to construct not only a knowledge base, but also an inference engine and an explanation facility. On the other hand, an expert system shell has a knowledge base editor, an inference engine, an explanation facility, debugging aids and other utility facilities which are already built into the system.

The major task of an ES developer who uses an ES shell is to retrieve and document the knowledge from human experts and then store them in a knowledge base via a knowledge based editor. Thus, using an ES shell saves the developer significant time and effort (Kim and Yoon, 1992).

Beynon-Davies(1991) identifies three advantages of using shells. These are rapid prototyping, imposed structure, and reduced necessary skill level.

One of the most difficult phases of expert system development is the construction of the first prototype system. Shells allow developers to rapidly prototype a solution to a problem. Also, shells impose prior structure, thus enabling developers to concentrate on substantive content other than form. And finally, shells help to reduce the levels of skill needed by developers, by effectively supplying some of the required expertise. In many instances, the knowledge engineer simply doesn't have the expertise to develop a system in a programming language.

For all of the above reasons, the developmental vehicle used in this research is the expert system shell. The development approach is the rapid prototyping approach which is detailed in the next two chapters.

Chapter 3: PROTOTYPE DEVELOPMENT - PHASE 1

"One must learn by doing the thing, for though you think you know it - you have no certainty, until you try"

(Sophocles, 400 B.C.)

Having decided on the development vehicle, an expert system shell, in the previous chapter, the next decision is the development approach. The approach utilised in this research is the rapid prototyping approach. The emphasis is on developing a rapid prototype of a system which then undergoes a number of improving versions (Beynon-Davies, 1993). In abstract terms, the method has been described as a five-stage process: Identification, Conceptualization, Formalization, Implementation and Testing (Hayes-Roth, et al., 1983). This is depicted in Figure 6. It must be noted, that these stages are highly interdependent and overlapping. For example, it will be difficult to say exactly when identification stops and conceptualization starts.

The prototype development, as outlined in the research approach described in section 1.2, is divided into two phases.

Phase 1 includes the identification, conceptualization, and formalization stages. Phase 2 consists of the implementation and testing stages. The first research objective is the creation of a knowledge base that captures the salient points of the CASE adoption/rejection decision process. This task involves all the activities in Phase 1, up to and including

the formalization stage, and is detailed in this chapter. The second objective is utilizing this knowledge base to build a prototype tool as a decision aid. This encompasses all the activities in Phase 2, and is treated in chapter 4. The successful construction of the prototype implicitly demonstrates the achievement of the two research objectives.

The chapter is divided into three sections, each section representing a development stage. The final subsection in each section contains the final output, or results, of that stage.

Rapid Prototyping (Case Adoption Decision Aid)

Figure 6

Source: Adapted from Hayes-Roth et al. (1983 p. 139)

3.1 Identification

Before developing an expert system, it is important to understand and describe with as much precision as possible the problem the system is intended to solve. This is the first stage of knowledge elicitation or acquisition. Identification involves eliciting the generalities of the knowledge domain (Beynon-Davies, 1993).

In this research, this step involved a review of the CASE adoption literature. A composite CASE adoption model which was presented earlier in section 2.2.5, provided a framework, and a starting point, for discussion with the domain experts. A sample of 7 firms was identified, and the domain expert(s) within the firms were chosen. A preliminary questionnaire (probe) was then presented at two of the sites, to the domain expert(s) to generate a first pass knowledge base, which outlined the adoption decision process. Also, at this stage the assessment criteria for the envisaged prototype tool were established. The criteria included both performance and utility measures. Some aims of the tool were also defined during this stage. For the sake of cohesion, these criteria are presented in the implementation section. Semi structured interviews were conducted at the remaining 5 sites, and this is detailed in section 3.2.2.

The final output of the identification stage is the general structure of the CASE adoption decision process. In effect, the composite CASE adoption model was supported, and

thus, represents the general structure of the CASE adoption decision.

3.1.1 Selection of Domain Experts

This was a two stage process: first, a sample of firms was selected, and second, the domain expert(s) within the firms were chosen. Companies were selected randomly from the Directory of High-Technology Firms in the Ottawa-Carleton Area (1992). It proved very difficult to identify firms that had adopted CASE and were willing to participate in the research, as no directory of CASE users is apparently available, and moreover, CASE vendors are reluctant to disclose information about clients. Of the 11 firms contacted, the first two to respond positively were selected to provide input for the identification stage, and the rest were utilised during the conceptualization and testing and assessment stages. Five companies were willing to grant interviews, and one domain expert served to test and assess the system. Three companies contributed in terms of contacts and material provided. Information about the participating sites is available on request.

Sample

Because knowledge elicitation requires continuing interaction, and because of cost and time constraints, the sample of firms was taken mainly from the Ottawa-Carleton region, with Montreal and Toronto being the outer limits. Due to the nature of knowledge, and since information is gained

both from what firms did, and also from what they now realize they should have done, it is possible to generate a comprehensive knowledge base with input from just a few firms. In fact, Waterman (1986) maintains that short interaction with many experts is less beneficial than in depth discussions with a few experts. Of course, the validity of the knowledge base elicited is affected by the quality of the experts. The generalizability of the results of the research will depend on how rigorously the knowledge base is tested and refined with inputs from other domain experts.

Domain Experts

Waterman (1986, p. 192) maintains that "Without a knowledgeable and cooperative domain expert, the system building effort is doomed to failure." He insisted that competency in the relevant area was the most important criterion for selection. This served to underline the importance of selecting the correct person(s) within each firm. The selected persons should, not only, have participated in the decision process, but also, have been aware of all the stages and factors that were involved in the adoption decision. If a suitable domain expert was not found at a site, then that site was removed from the sample. This proved to be the case at three sites. However, all three sites contributed in terms of reference material and contacts provided. In some instances, Firms adopt CASE after receiving advice from consultants. In one such situation it was deemed appropriate

to treat the consultant as a domain expert also.

At the firms contacted, there was little opportunity to select domain experts, since this part of the process was largely out of the control of the researcher, as the firms provided the domain expert. However, in all cases the persons interviewed were very knowledgeable, and were responsible for the adoption of CASE. This threat to external validity was reduced by combining the literature and the practitioner viewpoint thereby limiting any dependence on either. This concept is discussed further in section 5.2.

3.1.2 Preliminary Questionnaire Design [Probe]

As mentioned, a probe, or semi-structured, interview schedule with several open ended questions was administered at two firms that have adopted the CASE technology. The responses yielded a first-pass knowledge base. This knowledge base itself provided the structure for the structured questionnaire that was used to create or elicit the information for the final knowledge base.

The items on this probe attempted to determine if the composite CASE adoption model was accurate. The domain experts were asked to identify the steps in the adoption decision process, and generally describe the factors to be considered in the adoption of CASE. Although not divided into specific sections, the probe elicited demographic information, and then questions related to the various stages of the composite CASE adoption model were grouped together. The

output of the probe helped form the interview schedule which is detailed in section 3.2.2. The preliminary questionnaire is presented in Appendix F.

3.1.3 General Structure of Decision Process

The general purpose of the probe was to determine if there was significant difference between the CASE adoption literature and the way practitioners felt the adoption decision process should actually be carried out. Although the probe was only administered at two sites, it was supported with no dissension, so the composite CASE adoption model, as given in subsection 2.2.5, was used as the starting point, or platform, for further development.

3.2 Conceptualization

Once the general structure has been finalized in the identification stage, the specifics of the decision environment must be elicited. An attempt must be made to represent the knowledge domain in the form of concepts and the relationships between concepts. The output from this stage is a system dependency diagram.

3.2.1 From Preliminary Questionnaire to Interview

While the Probe served to support, the Composite CASE adoption model, the responses generated also posed some concerns. The way the items on the Probe were presented, allowed the respondents to give details of their own situation specific CASE adoption, without revealing the generic

parameters that need to be considered for any CASE adoption.

In other words, the responses were site specific, and although yielding a framework for the decision process, did not greatly assist the formulation of IF-THEN rules. The interview schedule then, needed a different approach. The items on this schedule were designed in such a way as not to elicit the specific value of an attribute, but to identify which attributes must be considered, and to decide which values or range of values of an attribute were desired.

For example, the prototype system will prompt for an input value (instance) for cost of the CASE tool, however, the system will need to know what cost is prohibitive. Now, while this value will vary from firm to firm, there should be a consensus as to what percentage of sales can cost afford to be, and still be acceptable, or perhaps a fixed amount based on the number of employees. This is the type of information that was elicited by the interview schedule which is presented in Appendix G.

It is important to understand this distinction, as the respondents are being treated as domain experts rather than simple statistical data points, because, knowledge elicitation can be comprehensive and beneficial with even a single respondent.

3.2.2 The Interviews

The interview schedule was administered at 5 firms that have adopted CASE. The interview schedule was self

administered in the classical structured interview manner (Greenwell, 1988). The knowledge engineer was interested in a complete description of the decision situation, such as, when the decision was made, what the values of particular parameters were, and what the likely outcomes of alternative decisions would have been.

Data

The semi structured interview schedule is divided into 5 sections: Company Profile, CASE Need, Organizational Readiness, Product Profile, and General.

Company Profile

Questions 1-7 comprised the company profile, and provided important background information on the firm, such as size, industry, and age.

The respondents were mainly software developers whose main clients are large firms, or government. The firms ranged in size from 50 employees to over 300, and all had been established for over 5 years. The person interviewed was usually the software programs manager.

Need for CASE

Q8 - When asked who introduced CASE, 80% responded that CASE technology was brought to the company's attention by the software engineer. In one instance it was introduced by the general manager. It seems that it is not that important who introduced CASE, but all agreed that the level of the project champion who is often the same person that introduced the

technology is important for successful adoption.

Q9 - The main reasons for CASE adoption varied across the respondents, from productivity increases, to required documentation procedures. However, there was no significant departure from the reasons found in the literature. All respondents agreed that there needed to be a cost/benefit justification for CASE adoption as in the final analysis, CASE is a business decision.

Q10 - When asked to identify the steps in the CASE adoption process, all the respondents agreed with the stages in the model, but two respondents identified additional steps that were incorporated into the modified CASE adoption model. One step was criteria selection and evaluation. The criteria that were used to differentiate between tools are themselves evaluated to ensure that valid criteria are being used. Clearly, if improper criteria are being used, the chance of selecting a suitable tool is lowered. The second additional step was a cost justification stage. Although somewhat contained in the need for CASE stage, the experts agreed that at some point management will need to be convinced of a net dollar savings in terms of the adoption of CASE. Dollars saved in less maintenance, less programming time, and greater productivity, must exceed dollars spent in acquiring and using CASE. The length of the payback period will vary from firm to firm, but if a net savings is not projected then CASE will not be adopted.

Q11 - When asked to give reasons for CASE adoption, all the experts agreed that a lack of control, poor productivity, desire to use new technology, and desire to achieve competitive advantage were valid reasons to adopt CASE, but 80% disagreed that reduced maintenance effort was a valid reason for CASE adoption, and none of the experts considered a software backlog as a good reason for adoption. Probable reasons are; one, that it is difficult for CASE to reduce maintenance on existing software if software was not developed using CASE, and two, that a software backlog may exist for reasons other than lack of productivity.

Q12 - When asked to decide between alternatives to CASE, the experts agreed that this was an invalid question as the choices offered are development methodologies rather than alternatives to CASE. However the question proved useful as it established the credibility of the experts. Interestingly, the experts did not perceive any viable alternatives to CASE.

Organizational Readiness

Q13 - When asked to identify conditions necessary for adoption, all the experts agreed that staff must be competent in structured analysis and design techniques and that a project champion must exist. However, 40% did not think a steering committee was absolutely necessary, and 20% felt that the training period need not be as long as 3 months.

Q14 - When asked to identify the level of sophistication necessary in the software development process, the experts

were divided on this issue. 60% felt that some benefits of CASE could be derived even at level 2, while 40% felt level 3 was necessary. All agreed that the chances of successful adoption increased at level 3 and above.

Q15 - The experts agreed that there was no magic number of programmers beyond which CASE became mandatory, as they maintained that CASE could be useful with even one programmer. They noted, however, that what would increase with the number of programmers is the level of sophistication of the CASE tool, and the need for a central repository.

Product Profile

Q16 - This question attempted to identify the selection criteria and their importance. 80% identified the need for a central repository feature as the most important, while one expert chose user interface as the most important. Their ranking of the other criteria varied widely, reflecting the different perspectives and differing needs. This highlighted the necessity of allowing users to select their own criteria for selection, and also possibly to decide their own weights for each criteria, as each firm has different needs.

Q17 - This question attempted to elicit how best to search the CASE marketplace. Again, responses varied widely depending on situation and preference, and the only agreement was that passive search was the worst method. All other methods were considered adequate, with one expert maintaining

that once 3 reliable vendors were identified, then products could be evaluated.

Q18 - When asked how the final choice was made, the experts agreed that generally, engineering made a recommendation to management who then made the final decision. Clearly, a tool will only assist in this process. While management will consider the recommendation of the system, and perhaps even encourage an in-depth trial or assessment of the recommended product, they will not slavishly adopt simply on the recommendation of the system.

Q19 - This question to determine the relative importance of Rogers' criteria, was added to the questionnaire after some interviews had already been conducted. As a result only one expert responded. The weights given by this expert will be used in the choice module of the prototype to accomplish the final selection of a suitable tool.

General

Q20 - All the experts agreed that top management participation is necessary for CASE adoption to be successful.

Q21 - All the experts agreed that a team consisting of engineers and managers was the best approach, and the project champion should be an important member of this team. Eventually, senior management will still have to ratify the decision of this team.

Q22 - 80% of the experts used some sort of capital budgeting method in a rudimentary fashion. One expert had

just finished developing a cost justification system. This emphasizes that to be commercially effective, a decision aid would need a cost justification module, as in the final analysis, technology adoption is still a business decision firmly focused on the bottom line.

Q23 - When asked if any software could have assisted the adoption decision, various suggestions were offered such as a cost justification tool, a process modelling tool that identified where CASE was needed most, and one expert was interested in developing a knowledge based tool to be placed on the internet providing a CASE adoption advisory service. Another expert thought that an expert system which could identify the type of CASE tool required when provided organizational parameters and software needs, would be useful.

Q23 - Strangely enough, none of the experts had considered assessing the adoption decision formally, although all were familiar with problems encountered during the adoption decision process. This is probably due to the fact that they would have to evaluate their own decisions which is often uncomfortable.

Q24 - When asked what they would do differently in retrospect, with regards to the CASE adoption process, some interesting thoughts were offered. One expert noted that it was important to ascertain if development schedules were flexible, as CASE adoption will increase analysis time at the beginning of a project considerably, but reduce implementation

and maintenance time later. However, if this is not understood by management, the project may fail simply due to inflexible schedules and an inability to meet certain preset deadlines.

Another expert mentioned that if CASE can be used on many projects, then justification is easier and volume discounts with site licenses can be obtained making adoption cheaper.

Another mentioned that experience had taught him that vendor viability, and maintenance and user support were very important selection criteria.

Also, one expert claimed that in retrospect, he would have raised the visibility of the project earlier, and kept senior management more frequently informed.

Summary

The respondents felt that the concept of a knowledge based tool was good, and that a tool could prove useful if certain features were available, such as cost justification, and criteria selection and evaluation. The composite adoption model was supported, although additional steps were suggested in the area of criteria selection and evaluation, and cost justification. These were included in the modified CASE adoption model.

A main theme stressed by all respondents was that ultimately the CASE adoption decision is a business decision, therefore the "need for CASE" section should justify in terms of dollars the benefits of adopting CASE. This hints at a

capital budgeting sub-module, or a specific cost/savings justification module. In fact, one of the respondents is in the process of developing just such a software. Unfortunately, due to time constraints, to incorporate such a module is beyond the scope of this research, but it will be duly noted as one area in which the prototype could be improved en route to becoming a viable commercial product. A segment will be added to the knowledge base to represent this module in an elementary fashion.

Another important observation was that depending on the aims of a CASE tool the criteria for selection will vary. While within a specific category such as analysis, or design, the criteria may not vary widely, there will be significant differences in the criteria for a code generator as opposed to a configuration management tool. This means that as part of the selection process the criteria themselves should be evaluated. This is further compounded by the fact that initially CASE tools addressed a particular phase of the life cycle like analysis or design, or maintenance, but currently, tools are becoming increasingly powerful and more difficult to slot into a category. In fact, one respondent felt that it would be sufficient for a tool simply to identify which type of CASE is needed. Since there are a limited number of products in any given category, it would be relatively simple to select a tool. To address this problem, in this research the conventional, even if somewhat outdated categories of CASE

are kept intact, and not every productivity aid is considered a CASE tool. Nonetheless, these concerns force a level of flexibility on the prototype if it is to assist in any practical manner. The prototype will have to allow users to formulate their own queries of the product database using varying criteria fitting their own situation.

The information gained from the interviews was examined, and used in the subsequent formalization stage to generate a comprehensive knowledge base that drove the construction of a prototype decision aid tool. This was an iterative process.

3.2.3 Modified Composite Case Adoption Model and System

Dependency Diagram

The criteria for the adoption decision were identified, and their relative importance to the decision determined. In effect, the composite CASE adoption model was supported, and in some ways augmented. Concepts such as conflict resolution, authorization, and lobbying were discussed, and their relationship to each other and the overall decision determined, however it was not feasible to include these steps in the prototype, as authorization and lobbying do not lend themselves easily to computerization. Conflict resolution is handled in a very rudimentary fashion utilizing the judgement of the knowledge engineer. If all the practitioners agreed but still conflicted with the literature, then the knowledge base reflected the practitioner position. For example, the experts agreed that passive search was not an effective search

technique so it was not included in the Knowledge base. If the experts disagreed, then the perspective of the literature was taken. For example, the experts disagreed on what software process level of sophistication was necessary, so level 3 the theoretical position was accepted. And, if the experts expressed positions that were not mentioned in the literature, then the knowledge engineer using his own judgement incorporated these views in the knowledge base. For example, the experts seemed to feel that there were no viable alternatives to CASE. An attempt was made to accomodate this view in solutions evaluation.

Arising from the interviews, a criteria selection and evaluation stage was added, but at this time, could only be incorporated in the database module in a somewhat simplistic fashion. This is an area that would require some attention for a commercial application. The user would have to be able to submit his own criteria which would then be evaluated by some scoring method. An adequate score would suggest that the user criteria are relevant and satisfactory. The modified CASE adoption model is shown in Figure 7. It is important to understand that not all facets of the model can be captured in the prototype.

System Dependency Diagram

The output from the conceptualization stage is the system dependency diagram. This diagram is depicted in Figure 8.

Dependency diagrams graphically model a knowledge based

system. Basically, a decision process is decomposed into segments or modules. The diagram shows the relationship between modules and segments, and even which IF-THEN rules are being used in a particular segment. This document "... serves as a complete reference outline for understanding, controlling, and coordinating a knowledge based system and its development" (Mockler, 1989).

While the composite CASE adoption model is an outline in a broader sense, the system dependency diagram is a much more detailed document, enabling the creation of IF-THEN rules in an ordered and systematic fashion. Although a knowledge base can be created without a dependency diagram, it would be akin to building a house without a blueprint. [Dependency diagrams are discussed in more detail in Appendix C]

As shown in Figure 8, the main modules are Need for CASE (NEED_CASE), Organizational Readiness (ORG_READY), Solutions Search (SOLUTION_SEARCH), and Number of Alternatives (NUM_ALT). This is in line with the composite CASE adoption model. Each module is then decomposed into submodules or segments. For example, NEED_CASE is decomposed into external factors (EXT_FACTOR), and internal factors (INT_FACTOR). External factors are further decomposed into competitive advantage (Comp_Adv), standards, and documentation. Internal factors are decomposed into production quality (Production_Quality), control, and employee need (Empl_Need). Then these segments are further decomposed until a

satisfactory level of granularity is obtained. The relationships between segments and modules are captured in the form of IF-THEN rules. These rules are extracted from the literature, or elicited from domain experts.

Modified CASE Adoption Model

Figure 7

System Dependency Diagram

Figure 8

3.3 Formalization

During this stage, the concepts and relationships among concepts determined in the conceptualization stage are formally represented as either rules, frames, or semantic nets. This means that the implementation vehicle has to have been decided on, and the method of representation, and a particular programming language or suitable shell chosen.

3.3.1 Selection of Method of Representation

As mentioned earlier in sub section 2.3.3, for a number of reasons, the development vehicle used in this research is the expert system shell.

The method of representation depends to a certain extent on the choice of the shell. Normally, the nature of the knowledge captured, and therefore the method of representation, should drive the selection of the shell, or even preclude using a shell. However, due to the constraints of the research resources, such as time, money, and programming expertise, the choice of a shell was driven by other criteria, such as ease of use, and availability. The selection of a particular shell is treated in subsection 3.3.2.

The three most common methods of representation are frames, semantic nets and rules. The most popular method, by far, and the one employed in this research is rules (Beynon-Davies, 1991 p. 41). Rules are preferred to frames and semantic nets mainly because of simplicity and ease of

implementation.⁶

Rules are a formal way of specifying a recommendation, directive or strategy expressed as IF premise THEN conclusion or IF condition THEN action (Waterman,1986). The major task of this research is to accurately capture the relevant portions of the CASE adoption decision process in a number of *IF-THEN* rules, in such a way as to assist the process.

3.3.2 Selection of a Shell

Kim and Yoon (1992) point out that selection of a shell for instructional or prototyping purposes will require a different set of criteria from that of selecting an ES shell for commercial applications. Using an analytical hierarchy approach, they evaluated four shells, short listed from approximately 40 shells. These were, *1st class*, *Level5 Object*, *Exsys Prof* and *VP-Expert*.

Two shells, *VP-Expert* and *Level5 Object* compared favourably on relevant criteria, with *Level5 Object* being a little more expensive. Since the knowledge engineer was more familiar with *VP-Expert*, which is simpler and easier to use, the knowledge base was created and modified using *VP-Expert*.

In a commercial system, the selection of a suitable shell is critical, as the level of sophistication desired has to be within the capabilities of the shell. However, in this particular research the emphasis is on the knowledge

⁶most inexpensive shells cannot handle mixed representation schemes

elicitation process, and demonstrating the feasibility of aiding the adoption decision process with a knowledge based tool. Thus, the focus is on creating an appropriate knowledge base, and the importance of the vehicle used to represent this knowledge base is reduced.

3.3.3 Resultant Rules

The output of the formalization stage is the resultant rule base. The rules are created in concert with the system dependency diagram as shown in Figure 8. This construction involved activities such as breaking down the decision logic, translating the logic into IF-THEN rules, debugging the rules, checking the syntax, and editing the rules. The complete rule base is given in Appendix E. A satisfactory rule base signifies the end of phase 1 of the research, and the achievement of the first research objective. It should be noted, that in many cases the rules were oversimplified in that choices were restricted to two options as opposed to three or four. For example, top management support was either high or low, with no intermediate value. This was done to restrict the number of rules, for as options increase, the number of rules required to represent all possibilities grows exponentially.

CHAPTER 4: PROTOTYPE DEVELOPMENT PHASE 2

This chapter contains the last two stages of development; implementation, and testing and assessment. At this point, the knowledge base has been created and captured in the form of IF-THEN rules signifying the end of Phase 1 of the research, and the achievement of the first research objective. The next phase of the research and the second research objective is to implement this rule base in a viable demonstration prototype tool.

4.1 Implementation

During this stage, the knowledge base is captured in a demonstration prototype system. This construction involved activities such as improving the interface, checking the output with varying inputs, modifying rules, debugging the rules, editing the rules, and refining the knowledge where necessary.

The final output of the implementation phase is a complete and functional "demonstration prototype" system together with system documentation. The system details are found in Appendix E.

The prototype was originally conceived as a rule based expert system. However, it became apparent that to be useful the tool should not only recommend whether to adopt CASE or not, but provide assistance in the final selection of a CASE

product. Thus, the prototype evolved to incorporate a Database module and a multicriteria selection module, thus becoming a very rudimentary expert database system.

The system consists of three modules, as mentioned:-

- (1) A knowledge based module implemented in *VP-Expert* which is a simple rule based expert system shell developed by Paperback Software Inc.
- (2) A Database module implemented in *DBASE IV* which is a relational Database Management program developed by Ashton Tate, and,
- (3) A multicriteria selection module implemented in *Expert Choice* which is a computerized manifestation of the Analytic Hierarchy Process, and is developed by Expert Choice Inc..

The interface is accomplished in a very simple manner, using Direct Access as a control program. Once the user exits a module, control is returned to Direct Access. In this manner, although this is not a truly integrated system, it is transparent to the user, and for demonstration purposes seems quite adequate. The remainder of this section discusses these three modules in more detail.

4.1.1 The knowledge Base

The knowledge base generally follows the composite adoption model which was used as a framework, and specifically is created from the system dependency diagram. The modified composite adoption model has three major sections that are

incorporated in the knowledge base. These are: needs recognition, solution search, and organizational readiness. Also, a cost/benefit section is added simplistically to represent the need for such a module. Basically, the system establishes that if there is a genuine need for CASE, and the solution search has been adequate, and the organization is ready to adopt CASE, then the recommendation is to adopt. If otherwise, depending on the situation, the recommendation will be wait or reject. The purpose of the knowledge elicitation is to determine what factors are important in deciding whether these conditions have been met, and if possible to elicit specific parameters for criteria. For example, one expert maintained that a 5% gain in competitive advantage due to CASE would justify CASE adoption.

Each subsection is further subdivided into factors that are derived from the literature and from the interviews with the domain experts. For instance, some of the factors that would determine organizational readiness are staff expertise, available resources, support for CASE, and the level of sophistication of the software development process. Some of these areas are further subdivided by indicators that would allow the knowledge engineer to decide whether a condition is being satisfied. In other words, to decide whether there is adequate staff expertise, it may be necessary to establish if staff is familiar with structured programming methods. This process continues until a suitable level of granularity is

obtained. At some point the knowledge engineer must call a halt to the decomposition process or else it would continue *ad infinitum*. In a demonstration prototype the levels of decomposition will be few, so that the system will appear to be trite, but the framework will be established that further decomposition may be accomplished easily if a more sophisticated system is desired. In a rule based system, the number of rules increases substantially as levels of decomposition are added.

System development varies slightly depending on the implementation vehicle. In this instance, *VP-Expert* simplified the development process. The tasks involved were mainly editing the rule base, and checking the logic, and syntax. A knowledge base created in *VP-Expert* is divided into three blocks; an actions block, a rules block, and a statements block. The actions block contains procedural directions for the consultation. The goal variable appears here, as well as display messages that contain the final results of the consultation. The rules block contains the IF-THEN rules representing the system logic, and the statements block contains messages that control user input and interaction with the system (Hicks and Lee, 1988). Also *VP-Expert* has several helpful features such as syntax checking, decision traces, and "what if" analysis. There is also the facility to enter certainty factors either during a consultation or in the rules. The entire knowledge base,

along with a trace of a consultation is presented in Appendix E.

The output of the knowledge base will be advice prompting the user either to adopt now, postpone the adoption decision, or reject the CASE technology. If the recommendation is to adopt, the user will access the database module to determine which tools are suitable.

4.1.2 The Database Module

Although the knowledge base is the focus of this research, it became evident that for the tool to be useful, it also should aid in the selection process which is an important step in the adoption decision process. To accomplish this, an elementary database was created using *DBASE IV* from Ashton Tate. At this point, it should be mentioned that *DBASE IV* was chosen simply because of familiarity and availability, and any recognized database program such as *Paradox* by Borland, or *Access* by Microsoft would serve equally well.

The database itself is small, containing approximately 50 CASE products out of an actual population of over 2000 products (Braithwaite, 1990). However, for the purposes of demonstration, this size of database should be more than sufficient.

Each product record contains fields that can be treated as criteria. For instance, a product record contains fields such as cost, platform, type, windows, etc.. Using the *DBASE IV* query feature, selected records can be retrieved from the

database. Thus, the user by designing his\her own query, and inputting parameters or values that are relevant to his decision environment may retrieve from the database the product records that fit his criteria. In other words, the user may select all CASE products under \$75,000, with a central repository feature, that run under windows, on a personal computer platform. These records will represent a shortlist of suitable CASE products given the correctness of the criteria. From these favoured products, a final selection will be made by a more rigorous examination of the selected products on a different set of criteria, taking into account the relative importance of each criteria. This is facilitated in the choice module.

The database system provides the usual database features such as adding records, deleting records, and editing and displaying selected records. The level of sophistication is kept simple, as the focus here, is not on developing a complex database system, but on demonstrating how a database system can enhance the utility of an expert system. System details are found in Appendix E.

4.1.3 The Choice Module

Once the decision to adopt has been made, and the product database manipulated to extract a shortlist of suitable tools, the final choice of a product should be made. In some ways, this is an optional module, as one could argue that by limiting the scope of the query, or increasing the rigidity of

the criteria the list of suitable tools could be reduced as desired. While this may be true, a database program will not provide a way of ranking criteria, or expressing preference between criteria. This is only handled rigorously by a multicriteria decision making process such as the Analytic Hierarchy Process (AHP). From this perspective the choice module is deemed necessary.

The choice module is implemented in *Expert Choice* which is a computerized version of the AHP, however, there are many other multicriteria packages such as *Electre III* and *Electre IV*. *Expert Choice* is chosen simply because of cost, availability and simplicity.

Following the principles of the AHP, a hierarchical model is created with the top level being the goal, which is selecting the 'best' CASE tool. The underlying levels are the criteria. In the model created, there was only one level of decomposition, containing the criteria identified by Rogers (1983) earlier. These are compatibility, complexity, trialability, relative advantage, and observability. Cost was also included. These criteria are then given a ranking or weight. These weights were based on the responses of the domain experts to the item concerning these criteria on the interview schedule. However, the user could build his\her own model with different criteria or different weights. The system will then by pairwise comparison, arrive at the 'best' CASE tool given the selection criteria. The model could have

more than one level of decomposition if desired.

Summary

In sum, the completed prototype allows the user to decide if CASE adoption is advisable at present. If so, then the system allows the user to identify a suitable CASE product that will match his\her organizational needs. Because the sample of CASE tools contained in the database is small, any tool recommended by the system is not necessarily the "best" tool available, but is the best of the tools entered in the database. A commercial system would not only have to have a complete database of tools, but also have procedures in place to update the database.

4.2 Testing and Assessment

In a commercial system, testing and assessment while never easy, are simpler than in a prototype building situation. The reason for this is that the requirements are (usually) more clearly defined in a commercial system. During prototype development, it is difficult to determine at what point the system is satisfactory, because the prototype can always be improved. As a result, system development is usually an ongoing iterative process involving knowledge refinement and subsequent testing and evaluation.

Waterman (1986) maintains that "evaluating an expert system is difficult because there may be no formal way to prove that a given answer is correct or the best possible

choice. The validity of the answer may depend on the persuasiveness of the argument given to support that answer."

Payne and McArthur (1990) identify three types of reviewers: The development team, experts and end users, and management. In this research, the development team was the knowledge engineer, the experts and end users were the domain experts mentioned earlier, and the examining committee could be considered management.

The knowledge engineer checked for implementation stability, and he decided when the system was ready for review. When the system ran without breaking, and manual output matched specified system output, and the user interface was deemed satisfactory, then the system was considered ready for review by the domain experts. Two sites that had nothing to do with the development of the system, and were therefore impartial, were chosen to review the system and assess its completeness. However, due to unforeseen circumstances only one expert was available.

The domain expert checked the validity of the knowledge base: domain facts, rules, and inferencing conclusions. He was satisfied with the interface in terms of its ability to inspect and update facts. The expert approved the recommendations provided by the system, and was satisfied with the performance (speed, efficiency) of the system. Figure 9 lists validation criteria, some of which were considered by the expert.

Measures of Validation

Measure (Criteria)	Description
Accuracy	How well the system reflects reality. How correct is the knowledge
Adaptability	Possibility for changes
Adequacy (or Completeness)	Amount of necessary Knowledge in Knowledge base
Breadth	How well the domain is covered
Depth	The degree of the detailed knowledge
Face Validity	How credible is the knowledge
Generality	Capacity to handle broad range of similar problems
Precision	Consistency of advice.
Realism	Similarity to reality
Reliability	The fraction of the ES predictions that are correct
Robustness	Sensitivity of conclusions to model structure
Sensitivity	Impact of changes in KB on quality of outputs
Technical and Operational Validity	Goodness of assumptions, context, constraints and conditions
Usefulness	How adequate is the KB
Validity	Capacity to provide correct advice

Figure 9

Source: Decision Support and Expert Systems (Turban, 1990, p. 476)

However, in the absence of normative measures, the attainment of standards specified at the identification stage is still a subjective decision.

4.2.1 Assessment Criteria and Aims of the Prototype

Hayes-Roth et al. (1983) mention 5 aspects of a system that need to be assessed to determine completeness: 1) Is the knowledge representation scheme adequate or does it need to be extended or modified? 2) Is the system coming up with right answers and for the right reasons? 3) Is the embedded knowledge consistent with the experts? 4) Is it easy for users to interact with the system?, and 5) What facilities and capabilities do users need?

Some general aims of the tool are: reduce subjectivity of the decision process, improve comprehensiveness of the decision (i.e., involve all factors), enhance soundness of the decision (i.e., reduce faulty logic), and base the decision on a verifiable knowledge base.

Once the knowledge engineer is satisfied that the prototype is operating correctly, and the aims have been achieved, then a domain expert must test the tool, and validate and verify that this in fact so.

4.2.2 Validation, Verification, and Testing

Validation refers to building the "right" system, that is, substantiating that a system performs with an acceptable level of accuracy. Throughout the development process it was necessary to refine the knowledge base. Any additional

knowledge gleaned from current literature or any other source was added. These changes to the knowledge base demanded periodic validation. Some measures of validation are shown in Figure 9.

Verification refers to building the system "right", that is, substantiating that the system correctly implements its specifications (Turban,1990).

Testing a knowledge based system is quite different from testing a conventional information system and may be quite esoteric. "The classical benchmarking approach does not generally apply to KBS, as different inputs are possible for different user queries and user defined information (Barstow, 1987).

In this research, assessment will simply involve a subjective evaluation by a domain expert of the knowledge base, using the criteria shown in Figure 9. Because of the prototypical nature of the tool, some categories such as robustness, sensitivity, and realism are not as meaningful as others. The following criteria were considered relevant for this particular prototype, and a brief discussion of the criteria from the perspective of CASE adoption ensues.

Accuracy - Since the Composite CASE adoption model was supported by the domain experts, and since the KB closely followed the model, the KB should be accurate. Moreover, if any specific parameter value such as for cost, or a length of time for an adequate solution search is deemed to be

inaccurate, then these can be quickly updated. Knowledge refinement of this nature would be an ongoing process.

Adaptability- Because of the use of a shell, and because the shell is simple and easy to use, it is very easy to modify the KB. A module can be added, or further decomposition effected if desired. The number of rules will increase rapidly with further decomposition, and as this happens, then modification and maintenance will become more difficult.

Adequacy - This is difficult to assess, especially since it is a prototypical tool, and since the CASE adoption decision process is not precisely defined in a normative fashion.

Breadth - If the Composite CASE adoption model is accepted, then the KB covers the domain. As the tool evolves into a field prototype, or a commercial system, the detail required would increase and also the coverage of the area. This would entail much longer and more extensive knowledge elicitation sessions with the domain experts.

Depth - This is limited, again due to the prototypical nature of the research and its objectives and constraints. As with breadth, as the system becomes more detailed, with more levels of sophistication, the knowledge elicitation process would have to be much more intensive. This knowledge base serves as a platform for further work of this nature.

Face Validity - This depends on the assessment of the domain experts and is one of the weaknesses of the research. However, because of the composite approach used, combining

both a theoretical approach, using the literature, and a practitioner based approach, using domain experts, the knowledge base should be credible even if the experts used are perceived as not being of top quality.

Generality - No attempt was made to apply the tool to other situations. This is an avenue of future research.

Precision - When given the same inputs, the KB produces the same advice or recommendation. This was verified by inputting the same test data repeatedly.

Usefulness - As mentioned, the KB serves as a platform for future development. By providing a framework of the adoption decision process, the groundwork is laid for a commercial tool. Such a tool could be of use to consultants advising on CASE adoption, or information systems departments advising top management, or service organizations such as CRIM or IDRC who could make such a tool available on the internet for wide public access.

Validity - It is very difficult to assess the capacity of the tool to provide correct advice. The domain expert simply used his experience to decide that the advice provided by the system was sound. Most business decisions are assessed in retrospect, and as such, some time would have to elapse to assess the recommendations made by the tool.

CHAPTER 5: DISCUSSION AND CONCLUSIONS

"It is difficult to predict, especially about the future"

(Chinese Proverb)

There were several issues that affected the research endeavour, that became evident during the research. In some cases it was possible to resolve or minimise the effects, in other cases it was only possible to identify the problem, and suggest possible solutions which could be attempted in future research projects. These issues ranged from the well known difficulties of knowledge elicitation and representation, to more abstract concerns such as whether the prototype should be normative or descriptive in approach. These concerns are discussed below, followed by the benefits and limitations of the research, some general conclusions of the research, and some directions for future research.

5.1 Single versus Multiple Experts

The interviewing process while yielding valuable knowledge with which to refine and improve the knowledge base also revealed a flaw in the research design. The idea of using multiple domain experts seemed sound in theory, and is in fact supported in the literature (Mittal et. al., 1984). However, this probably holds true for commercial projects where either the expert is being well rewarded for his/her time and effort, or is being quietly coerced. In the absence

of these motivating factors as occurs in a student project, it is very difficult to gain access to the expert for a sufficient length of time. Moreover, knowledge elicitation, to be done properly, requires ongoing visitation for the purposes of knowledge refinement. This proved very difficult, and highlighted the fact that in a project of this nature, one very accessible expert is ideal, with perhaps multiple experts being contacted for testing and assessment, although the same problem is likely to occur. In fact, if a suitable expert can be identified, then recent experience suggests that this will probably yield a more comprehensive knowledge base and a less frustrated knowledge engineer. Also, interviewing multiple experts proves inefficient, as 'familiarization' has to be done repeatedly.

However, the effects of this problem were reduced by adopting a composite approach.

5.2 Composite Approach

An expert system, or for that matter any software system, can be developed entirely from theoretical constructs. However, in the case of expert systems, where the knowledge of the expert is elicited and captured, there is somewhat of a dilemma. Often, books or theory, differ significantly from the way the expert approaches the problem in the field. The question arises as to how to decide which knowledge is correct. In deterministic situations where the outcomes can

2 of/de 2

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

**PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT**

PRECISIONSM RESOLUTION TARGETS

easily be evaluated as correct, or incorrect, the procedure is relatively simple. Given a problem or series of problems, one knowledge base, theoretical or practical, will perform better, that is, will yield a greater number of correct solutions or responses. However, in a non-deterministic arena where it is not clear how to evaluate solutions or responses the procedure is less evident. In this particular research, the underlying area of concern is decision making, and particularly nonprogrammed decision making. The prototype system will yield a recommendation, but how does one evaluate this recommendation as correct or incorrect. Decisions, especially managerial decisions are usually neither wholly correct nor wholly incorrect, and given incomplete information and an uncertain environment a 'correct' decision at time "t" may produce an unfortunate outcome at time "t+1". Thus, there are actually two related concerns: one, is to consider whether the prototype should be based on theoretical or practical knowledge, the other is to decide if in a dynamic situation, the success of a decision should be assessed on the outcome of the decision, or the soundness of the decision.

In this research, in order to address this problem, a composite approach was used. By using the theory, or literature, as a starting point, and incorporating practical knowledge by interviewing actual practitioners in the industry, a compromise was made. While not representing a definitive solution to the dilemma, it does provide a method

of minimising possible deviations in accuracy. By not depending completely on the experts provided by the firm, the problem of establishing the validity of the experts is also reduced.

The experts themselves, while revealing insights into the CASE adoption decision process also highlighted some problem areas.

5.3 The CASE Adoption Decision

A major portion of the difficulties associated with this research is attributable to the complexity of the CASE adoption decision process. Particularly in the area of determining a need for CASE, it is not clear if this knowledge exists in a normative fashion. For example, the CASE literature presents some criteria to be considered in deciding if CASE is necessary, and domain experts interviewed supported these in general. However, these criteria mainly address the present situation of the firm with regards to software productivity or capacity, yet technology adoption has implications not only for the present but for the future. Thus, management has to predict how CASE will impact on their future situation. This uncertainty compounded the elicitation process, and although the creation of a knowledge base provides structure for this ill-defined task it must not be perceived as a panacea for the problems facing the decision maker. This is not to detract from the utility of the

research, but simply to reiterate the fact that decision making is still a relatively vague area. In fact, many benefits are derived from the research.

5.4 Benefits of this Research

The nature of this research project is such that benefits both tangible and intangible were derived. The elicitation process proved more difficult than envisaged, yet the expertise captured, yielded a valuable snapshot of the essence of the adoption decision process.

Thus, an intangible benefit of the research is a greater insight into the adoption decision process, and a tangible benefit is the creation of a prototype that can serve as the logical design and a platform for the development of a computerized decision aid that can assist the CASE adoption decision process. The concept that unprogrammed decision making can be assisted by a knowledge based system, enhanced by a database management system should prove useful to decision makers in general. Also, the idea of incorporating multicriteria decision making techniques such as the analytic hierarchy process into the CASE adoption process should be particularly useful to CASE adopters, especially in the area of criteria selection and evaluation.

The specific beneficiaries of such an aid will be CASE consultants who may use the tool to provide better advice to their clients, CASE developers who may use the tool as a

marketing aid, or simply managers who are confronted with the CASE adoption decision.

5.5 Limitations of this Research

The main limitations of this research are in the areas of knowledge elicitation and knowledge representation.

Due to the small sample size, and the shortcomings of the elicitation process in terms of time and intensity, it is difficult to verify the accuracy or completeness of the knowledge base elicited. The decision process at the firms interviewed may not be representative of the process in general. Similarly, but even more significant, the persons selected as domain experts may not be genuinely qualified to act in this capacity, and may have provided information leading to an invalid knowledge base, and consequently a tool that could provide incorrect advice.

Also, as with any Expert System, the inference mechanism and knowledge representation method will impose constraints on how closely the knowledge base captured represents the expertise which is being elicited. In this instance, the shell selected, *VP-Expert*, and the rule based representation offer the best level of sophistication possible within the constraints faced.

Finally, due to time constraints, the prototype tool developed may not be as complete or as sophisticated as one could envisage. This will also offer opportunities for future

research.

5.6 Conclusions and Directions for Future Research

Conclusions are presented at three levels of abstraction. The first level is the practical mundane description of lessons learnt, and problems faced. The next level is the more formal discussion of the achievement of research objectives, and the implications of the results of the study, and the third level is the direction of future research suggested by the study.

A valuable lesson learnt during this research is that for an expert system project requiring knowledge elicitation, the most critical factor is the selection of domain experts, and, if possible, repeated sessions with one very accessible expert will prove superior to limited sessions with several experts. Time allowed, and interviewing expertise will largely decide how successful the elicitation process can be.

Another lesson learnt is that the selection of a suitable expert system shell with which the knowledge engineer is familiar will substantially reduce development time.

The successful creation of first the knowledge base, then a prototype tool signified the achievement of the research objectives. The research demonstrates that an effective decision aid must incorporate both theoretical and practical knowledge, and moreover, depending on the decision situation should include both knowledge based as well as conventional

programming techniques. In the specific case of CASE adoption, an effective decision aid tool must identify the need for CASE, justify the CASE adoption in terms of dollars saved, verify that the organization is ready to adopt CASE, confirm that an adequate search for solutions has been done, and assist in the selection of a particular tool. The successful creation of the prototype suggests that with a more thorough elicitation process, a hybrid decision aid of commercial value can be created. Further, the modified composite CASE adoption model created during this research is seen as a valid theoretical framework of the CASE adoption decision process.

An important conclusion of this research is that, not only may an automated tool assist the CASE adoption decision process, but also, that there are parts of the decision process which will be very difficult if not impossible to automate. Future research can explore how best to automate more of the decision process, but due to the uncertain and dynamic nature of the decision process there will still need to be a human component to the CASE adoption decision.

Future Research

There are several opportunities for future research, the most obvious being a simple extension of the present research. That is, using the present prototype as the logical design, complete the tool by sampling more firms that have adopted CASE, or even better, identifying a suitable site where

continuing and intensive knowledge elicitation can occur. By refining the knowledge base, in this manner, and creating a more comprehensive tool database, as well as creating more levels of criteria in the choice module, a more complete tool that will be better able to assist the technology adoption decision will be ensured. The tool will need a complete cost justification module. Such a module will provide capital budgeting analysis with varying time frames, and using existing statistics for maintenance time and development time per lines of code will provide quantitative estimates of monies saved due to CASE adoption. It may also be worthwhile to explore if the tool could be applied to similar technology adoption decisions.

Another avenue of future research is in the area of hybrid expert systems as the idea of combining expert systems with database concepts or object oriented techniques is in its infancy. Particularly interesting would be further exploring the concept of using an expert system as an intelligent front end. The idea of intelligent querying coupled with an object oriented database is worthy of study.

Yet another valid area of research highlighted by this research is the validation, verification, and testing of knowledge based systems. As knowledge based systems, especially hybrid systems become more prevalent, the testing and assessment of these systems becomes more complex and more important. For example, knowledge based systems can be

examined as a whole, or each component such as the knowledge base or inference engine can be examined separately. With tightly coupled systems where the knowledge base is providing intelligent querying of the database it becomes even more difficult to anticipate the output and test correctness, as the interaction between components increases.

References

Ahituv, N., and Neumann, S., Principles of Information Systems for Management, Wm.c.Brown, 1990.

Barstow, D., "Artificial Intelligence and Software Engineering", Proc. ICSE 9, Monterey, CA, 1987.

Bachmann, C. W., "The programmer as navigator" Communications of ACM, vol. 16, pp. 47-50.

Bouldin, B. M., Agents of Change: Managing the introduction of automated tools", Yourdan Press, 1989.

Beynon-Davies, P., Knowledge Engineering for Information Systems, McGraw-Hill, UK, 1993.

Beynon-Davies, P., Expert Database Systems A gentle introduction, McGraw-Hill, UK, 1991.

Bowen, J., A Proposal for Developing an Expert System from an Investigator's Decision Process, MMS Thesis Proposal, 1991.

Braithwaite, K.S., Applications Development Using CASE Tools, Academic Press, San Diego, California, 1990.

Date, C. J., An Introduction to Database Systems, Addison-Wesley, Reading, Mass., 1986, vol. 1, 4th ed..

Dewey, J., How We Think, D. C. Heath, New York, 1910, chapter 8.

Digital Consulting Inc., "CASE: The Next Generation", Summer, 1990.

Fichman, G. R., and Kemerer, F. C., "Adoption of Software Engineering Process Innovations: The Case of Object Orientation", Sloan Management Review, Winter, 1993, pp. 7-22.

Fisher, A. S., CASE: Using Software Development Tools, John Wiley & Sons, 1991, 2nd ed..

Floyd A. S., and Wakefield G. S., "A Knowledge Based Decision Aid for Research and Development Project Selection", Management of Technology III, Institute of Industrial Engineers, 1992, pp. 359-368.

Gordon, L. A., Miller, D. and Mintzberg H., Normative Models in Managerial Decision-Making, National Association of Accountants, New York, 1975.

Greenwell. M., Knowledge Engineering for Expert Systems, Ellis Horwood series in Expert Systems, Ellis Horwood Limited, Chichester, 1988.

Hanna, M., "Can Case bridge to object world? object technology debate continues as vendors search for middle ground", Software magazine, July, 1993, pp. 41-46.

Harmon, P., and King, D., Expert Systems-Artificial Intelligence in Business, John Wiley and Sons, New York, 1985.

Hayes-Roth, F. ,Waterman, D. A.,and Lenat, D. B., Building Expert Systems, Addison-Wesley, 1983.

Hicks, R., and Lee, R., VP-Expert for Business Applications, Holden-Day Inc., 1988.

Hoffman R. R., "A Survey of Methods for Eliciting the Knowledge of Experts", Readings in Knowledge Acquisition current practices and trends, Ellis Horwood series in Artificial Intelligence, Ellis Horwood, 1990, pp. 7-20.

Hoffman, R. R., "Methodological preliminaries to the development of an Expert System for aerial photo interpretation." Report ETL-0342. U.S. Army Corps of Engineers, Engineer Topographic Laboratories, FT. Belvoir, VA, 1984.

Holloway, N., "Categories of CASE Tools", ICP Business Review, December, 1987.

Holloway, S., The Future of Data Dictionaries, Gower Technical Press, 1989.

Howe, D. R., Data Analysis for Data Base Design, Edward Arnold, London, 1983.

HP CASE PRIMER: An introduction to Computer Aided Software Engineering, Hewlett-Packard Company, 1989.

Humphrey, W., Managing the Software Process, Addison-Wesley, 1989.

Kim, S. C. and Yoon, Y., "Selection of a good expert system shell for instructional purposes in business", Information and Management, 1992, pp. 249-262.

Loo, S. C. K., An Examination Of The Decision Making Process in AMT Investment Decisions, M.M.S. Thesis, 1987.

Mclure, C., CASE is Software Automation, Prentice-Hall, 1987.

Meilleur, J., "Client/Server Development Tools for Windows", Paper presented at Comdex Canada, 1993.

Mittal, S., and Dym, C.L., "Knowledge Acquisition from Multiple Experts", IEEE Congress Knowledge Acquisition, 1984, pp. 75-81.

Mockler, R. J., Knowledge based Systems for Strategic Planning Prentice Hall, 1989.

Payne, C. E., and McArthur, R. C., Developing Expert Systems A Knowledge Engineer's Handbook for Rules & Objects, John Wiley & Sons, 1990.

Quinlan, J. R., Applications of Expert Systems, - vol. 1, Addison-Wesley, 1987.

Quinlan, J. R., Applications of Expert Systems, - vol. 2, Addison-Wesley, 1989.

Richards, N. L., "AD/Cycle - A New Unofficial Standard", CASE on Trial, edited by Kathy Spurr and Paul Layzell, John Wiley & Sons, 1990, pp. 147-161.

Robertson, T. S. and Gatignon H., "Competitive Effects on Technology Diffusion", Journal of Marketing, vol. 50, July, 1986, pp. 1-12.

Robertson, T. S., Innovation Behaviour and Communication, Holt, New York, 1971.

Rogers, E., Diffusion of Innovations, The Free Press, 1983, 3rd ed..

Rogers, E., and Shoemaker F. F., Communication of Innovations: A Cross-Cultural Approach, The Free Press, 1971.

Saren, M. A., "A classification and review of models of the inter-firm innovation process", R & D Management, vol. 14, no. 1, 1984.

Simon, H. A., The New Science of Management Decisions, Prentice Hall, Englewood Cliffs, New Jersey, 1977, Revised ed., (1960).

Simons, G. L., Expert Systems and Micros, NCC Publications, England, 1986.

Turban, E., Decision Support and Expert Systems, Macmillan, New York, 1990, 2nd ed..

Vonk, R., Prototyping, the effective use of CASE technology, Prentice Hall, London, 1990.

Waterman, D., A Guide to Expert Systems, Addison-Wesley, 1986.

Williams, R., Using CASE Tools in Systems Development their Scope and Value, Gower, 1990.

APPENDIX A:

CASE:Tools and Environment

Originally, software tools supported narrowly defined tasks, but increasingly, sets of tools closely attuned to one another are becoming available. These sets of tools are referred to as workbenches, and depending on the activities supported they will be known as "analyst, designer, programmers, or project management". Analyst workbenches here include analyst and designer workbenches, as the boundary between these two tools is vague.

Analyst Workbenches

As the title suggests, analyst workbenches(AWB) aim primarily at supporting the system developer during the early or analysis phases of a development project. According to Vonk (1990) the necessary tools are as follows; a data dictionary, a diagram editor, an intelligent text editor, prototyping facilities, analysis facilities, reporting facilities, facilities to share meta-data, and a user interface.

Data Dictionary

An analysis work bench (AWB) must allow the developer to store data produced during the development process. This central store of data about data, or meta-data is generally called a "data dictionary." AWB tools should include facilities for storing and retrieving different versions of

models or prototypes from the dictionary.

Diagram Editor

AWB tools should make it possible to construct, save, and modify diagrams in an interactive and user-friendly way. Basic operations such as adding, removing and repositioning objects must be available. Various features like "zooming in" and "decomposition" are necessary. The developer should be able to move from one level of detail to another easily. Also, diagrams should reflect what is currently in the data dictionary, so diagram storage with automatic update is essential. Actually, diagram regeneration from the data dictionary is how this is accomplished.

Intelligent Text Editor

Eventually, the situation being described or modeled, becomes too intricate for simple graphical representations, and text or code is necessary. A 'syntax directed' or intelligent editor that knows the language is required. The developer will only be allowed to use 'correct' language constructions, and should be warned when grammar has been contravened. Again, updates to data elements must be reflected in the dictionary. This is accomplished as with diagrams, by regenerating text from the data dictionary.

Prototyping Facility

An AWB should facilitate different approaches to 'requirements definition', including prototyping. To be able to produce a functional prototype, the AWB should have a

screen painter, a report writer, a relational DBMS and a fourth generation language, or be coupled to a fourth-generation environment containing such facilities.

Analysis Facility

Often, the AWB must assist the developer when the developer is still working in an exploratory manner. Therefore, with Entity Relationship Diagrams (ERD), and Data Flow Diagrams (DFD), logical rules affecting objects must be obeyed. The more automatic assistance of this nature, the better the tool. "Although the graphics capabilities of a workbench are generally impressive, and are often the main motive for acquisition, its real strength is to be found in its analysis facilities" (Vonk, 1990, p. 117).

Report Generation

During the development process, various reports on the contents of the data dictionary will be required. The AWB must provide a query language and a facility for modifying the lay-out of generated reports. Often, the AWB will need to offer desk-top publishing facilities depending on the type of report required.

Facility for the shared use of Meta-Data

Ideally, since most software development is done in teams, a multiuser AWB is required where more than one developer can share the Data Dictionary. The AWB has to provide for integrity of the shared Data Dictionary, and the security issues can be quite complex.

User Interface

It is very helpful if all workbench components interface with the developer in a uniform way, therefore, through a single interface. Today's medium of choice is the 'mouse', with the keyboard only used for entering text. Menus vary, but pop-up menus with highlighted choices are currently popular.

Windows and windowing is quickly becoming a standard, and every software developed today tries to emulate or support windows. True multitasking is very useful, as a time consuming task may be going in the background while editing occurs in the foreground. Therefore, AWB tools that will be used to develop windows applications should also be able to take advantage of 'windowing'.

System Generators

These tools support the improvement of productivity in the conversion of business data and processing definitions automatically into computer activities. This produces a set of well documented and efficient programs and screens. These support an automatically generated database structure that is truly reflective of the business data relationships and contents of importance to users (Holloway 1990).

Generally back end tools, these tools support the construction and maintenance phases of the SDLC. These tools include code generators and re-engineering tools.

Code Generators

Given system requirements in terms of data relationships and data flow, these tools will generate necessary program code.

Re-Engineering Tools

These include two type of tools:

- 1) Tools which take existing code and "structure" it.
- 2) Tools which help reverse engineering of a system. That is, structure charts, data dictionaries etc., are created for an existing system.

Life Cycle Products

These tools are a hybrid of the previous two types of CASE tools mentioned.

IPSE Tools

These facilities offer an Integrated Project Support Environment which can enable a system to be developed in a unified computer environment. There are two basic categories of IPSE tools: Open and Closed.

Open IPSE provide a standard platform or framework so that other third party tools not provided by the IPSE supplier may be used. The market is moving towards these tools, as obviously utility and functionality are greatly increased.

Closed IPSE's offer a package of design, project management, data handling, methodologies and CASE tools that must be applied by all projects using it.

Strategic Planning Tools

Such CASE products are targeted at the end user community, who need to obtain and manipulate data for planning purposes. They contain (what if) facilities which allow data to be selected and represented graphically and then passed to other packages for more detailed statistical analysis (Williams, 1989).

CASE Environment

At present, considerable effort is being made to standardise the CASE environment. The incentive is enormous, for as CASE tools begin to work on various operating systems and across various hardware platforms from mainframe to microcomputer their use will increase tremendously.

IBM with its application development (AD\CYCLE) concept is attempting to generate an industry standard platform. Building on Systems Application Architecture or SAA, IBM intends to insert a series of "software interfaces, conventions and protocols between the various operating systems and the application development software, so that systems would perform in a standard manner once they are run on an SAA participating machine." (Richards, 1990) In other words, the user would be indifferent to the machine used. This portability would immediately attract third party vendors which would greatly improve the quality and variety, of CASE tools, and also reduce the cost. The combination of these factors will greatly increase use of CASE tools in the future.

New Software Development Life Cycle [SDLC]

It is important to understand that the 'way' systems are being developed is also changing. Users are now determining their system requirements. Another change is that data modelling is now much nearer the front end. "Traditionally, systems developers did not begin to think about file design until the logical system design had been completed. But in those days, systems tended to own their own data. Nowadays, data is acknowledged as being a corporate resource to be shared across systems, so the data requirements for a system are addressed in great detail during the functional definition stage." (Richards, 1990)

Various methodologies have been developed recently, like structured systems analysis and design method (SSADM), logical systems development methodology (LSDM), and Information Engineering by Yourdan and Jackson. This has important implications for CASE, as CASE tools automate the tasks that are performed under the method.

It is necessary to emphasise that it is the method or the structured approach to the development that gives the benefits in terms of quality and productivity. It is the CASE tool that allows those benefits to be delivered

New Types of Systems

Systems are now workstation based, so there is a real need for distributed databases and distributed processing. Also, a strong need for security, and reliability, exists as

information is more and more provided on a need to know basis. Moreover, today's systems are breaking down old barriers between data processing and text processing, and tomorrow's systems will be knowledge based. Consequently, as the complexity of application development increases, more tasks will have to be automated, and the need for CASE tools will increase.

In summary then, there is a wide variety of CASE tools, and no one tool will be sufficient to provide a complete development environment. Also, CASE tools are still in their infancy, and as tools improve, and their connectivity and portability increases, then a truly automated software development environment will become a reality. A system that can match tool and vendor, as well as detail the criteria for adoption, will definitely assist the adoption decision.

CURRENT CASE REQUIREMENTS

ABLE TO PRODUCE

- Entity Relationship Diagrams (ERD).
- Data Flow Diagrams (DFD).
- Process Decomposition.

MULTIPLE USER ACCESS

- Installation on a LAN
- Simultaneous access for multiple users>>implies item locking.

SCHEMA GENERATION

- Import existing Data Dictionary language.
- Export Data Dictionary language from design.

REPORTS

- Structure Charts.
- Decomposition Diagrams.
- Entity Relationship Diagrams.
- Data Flow Diagram.
- Standard Reports (Desktop Publishing).

FEATURES

- Open Data Dictionary Format.
- Store Design Data in Non-Proprietary Database.
- Ability to Query DATA.
- Create Non-Standard Reports.

APPENDIX B:

Expert Systems

Artificial Intelligence, or AI, is an interdisciplinary sub-field of computer science concerned with both symbolic processing and human problem solving (Harmon and King, 1985). The three main areas of AI are Natural Language Processing, Robotics, and Expert Systems. The relevant area to this research is Expert Systems.

AI focuses on defining the problem in terms of states and operators, and then generating a search space and examining it. If the problem is amenable to an algorithmic solution, that is, the search space is well defined and not too large, then the solution should be developed by conventional means (Basic, Pascal, C). However, if the search space is large, then knowledge engineering techniques can be applied to reduce the search space, and to develop a knowledge framework as a mix of inference and representation. These heuristic programs are called expert systems.

Components of an Expert System

An expert system is composed of a knowledge base, an inference engine, working memory, a user interface, and often a knowledge acquisition facility which allows a domain expert to impart his expertise directly.

A knowledge base holds the knowledge about a particular topic in the form of Facts (e.g. John is a man) and Relationships (e.g. John is the father of Bob).

An inference engine uses the knowledge in the knowledge base to infer new knowledge by questioning the user and interpreting the appropriate rules of relationship.

The working memory is a data area for storing intermediate results generated by problem solving.

The user interface encompasses all the various methods of communication between the user and the computer system.

Knowledge Acquisition

Knowledge acquisition, or KA, is the collection and analysis of information from one or more domain experts and any other sources leading to the production of a number of documents which form the basis of a functioning knowledge base. A substantial subset of knowledge acquisition is knowledge elicitation. Knowledge elicitation, or KE, is that area of knowledge acquisition which deals with acquiring information directly from domain experts. A diagram showing the various participants in the ES arena is presented in Figure 10.

Knowledge Representation

The three most common ways to represent knowledge in an ES are: rules, semantic nets, and frames.

Rules are expressed as IF-THEN statements. When the IF portion of a rule is satisfied by the facts [data elements], then the action specified by the THEN portion is performed, and the rule is said to fire or execute. There are two different inferencing methods for using rules: forward

chaining and backward chaining.

Forward chaining occurs when you start with facts and premises, and work "forward" to a goal. In backward chaining, you start with a goal and work backwards to determine what facts and premises must be true for this goal to occur.

A semantic net is a knowledge representation method based on a network structure. A semantic net consists of points called nodes connected by links called arcs describing the relationships between the nodes. The nodes in a semantic net stand for objects, concepts or events. Common arcs include isa and has-part. Items lower in the net inherit properties from items higher up in the net.

Frames are very similar to semantic nets. A frame is a network of nodes and relations organized in a hierarchy, where the topmost nodes represent general concepts and the lower nodes more specific instances of those concepts. The concept at each node is defined by a collection of attributes called slots which will contain values. Each slot can have procedures attached to it. For example, three common types of procedures are:-

- 1) If added - executes when new information is placed in a slot.
- 2) If removed- executes when information is deleted from a slot.
- 3) If needed executes when information is needed from the slot, but the slot is empty.

These attached procedures can monitor the assignment of information to the nodes, making sure that appropriate action is taken when values change.

Types of Expert Systems

The five major types of expert systems are knowledge based information systems, decision support systems, consultation systems, problem solving systems, and coaching systems (Greenwell, 1988).

"Decision support expert systems take the advisory role of the expert. The system assists the user in the decision making processes by asking questions and interpreting the answers, and has a powerful role as an organisational aid to decision making". (Greenwell, 1988).

Knowledge-based information systems tend to be intelligent front-ends for accessing large databases. The system assists with enquiry but does not solve problems or perform any recognised tasks. Generally these are not very complex systems but probably beneficial and cost-effective.

Consultation systems perform the reasoning in order to produce a diagnosis or report. A major feature is the ability of the user to interrogate the system's reasoning.

Problem solving systems reason and offer a diagnosis with optional advice. Some problem solvers are not knowledge-based but fall within the category of artificial intelligence solutions. They are usually very complex systems.

Coaching systems are used to coach a student through the

problem solving process. A feature of this type of system is the ability to adapt to individual users (Greenwell, 1988).

Stages of Development

Waterman (1986), defines five stages of evolution of an expert system. These are: Demonstration prototype, Research prototype, Field prototype, production model, and commercial system.

A demonstration prototype is a small demonstration program that solves a portion of the problem undertaken, suggesting that the approach is viable and system development is achievable. A typical rule-based demonstration prototype will contain 50-100 rules and take one to three months to develop.

A research prototype is a medium sized system that displays credible performance on the entire problem but may be fragile due to incomplete testing and revision. A typical research prototype will contain 200-500 rules and take one to two years to develop.

A field prototype is a medium to large system that displays good performance with adequate reliability and has been revised based on extensive testing in the user environment. A typical rule-based field prototype will contain 500-1000 rules and take 2-3 years to develop.

A production model is a large system, exhibiting high quality, reliability, speed, and efficient performance. A typical production system will contain 500-1500 rules and take

2-4 years to develop.

And finally, a commercial system is a production model that is being used on a regular commercial basis. Few exist, and typically such a system takes over 5 years to develop.

Advantages and Disadvantages

Floyd and Wakefield (1992) identify three main advantages of ES. These are:-

- 1) They can handle less structured management problems.
- 2) They can handle missing information, and
- 3) They are much more adaptive and flexible than classical systems.

Some disadvantages are high cost and lengthy development time. These problems are largely due to the difficulties associated with knowledge elicitation. Another disadvantage is that it is sometimes difficult to validate the knowledge base, which affects the confidence which can be placed in the system.

Players in the Expert System Game

Figure 10

Source: A Guide to Expert Systems, Waterman, 1986, p. 8

Appendix C:

Dependency Diagrams

"Dependency diagrams graphically model a knowledge based system. They show the knowledge segments and their interrelationships. They also indicate some of the basic reasoning processes by which the knowledge is manipulated ... They give a complete and accurate picture of the entire system or system component they represent" (Mockler, 1989).

The following components are present in a complete system dependency diagram:

Triangles - The triangles have numbers that refer to the rules in the knowledge base that manipulate the knowledge segments adjacent to the triangles. There will be a decision chart for every triangle.

Boxes and Arrows - Boxes and arrows adjacent to a triangle show the name of the knowledge segments to which they are related and that they manipulate.

Question Marks - Question marks indicate questions asked by the system in order to get information input from users into the system.

The subject of the questions is given in the parentheses.

Acceptable values for each phase in the process are given under the input arrows and boxes. The recommendations to be made by the system during a consultation are named under the final box in the diagram, which represents the decision to be made.

The system dependency diagram serves as a complete reference outline for understanding, controlling, and coordinating a knowledge based system and its development effort. Such diagrams are mandatory for any well-planned and executed systems development effort (Mockler 1989).

Appendix D:

MultiCriteria Decision Making (MCDM) and The Analytic Hierarchy Process

"Multiple Criteria Decision Making occurs when decisions are made in the presence of usually conflicting objectives." (Zionts, 1987) Or similarly, "The most common problem in business and in other situations, including personal life, is selecting one or more alternatives from among a number of competing alternatives. Such decision is considered as multiple criteria analysis. (Hobbs, 1984)

Stanley Zionts divided the field of MCDM into four broad categories: 1) Multiple Criteria Mathematical Programming, 2) Multiple Criteria Discrete Alternatives, 3) Multiattribute Utility Theory, and 4) Negotiation Theory. In this research, MCDM is synonymous with multicriteria discrete alternatives, and the alternatives are different CASE products. Also, MCDM is mentioned only in relation to the AHP. If a more rigorous treatment of MCDM is desired then Zionts, 1987, or Zahedi 1984 is recommended.

The Analytic Hierarchy Process (AHP)

The AHP is merely one of several MCDM methods, but because of the energy and eminence of its creator, mathematician, Thomas Saaty, it is probably the best known. Moreover, Expert Choice which is a computer program embodying the AHP, is an inexpensive, easy to use product.

According to Saaty, "the AHP is a theory of measurement. When applied in decision making it assists one to describe the general decision operation by decomposing a complex problem into a multilevel hierarchic structure of objectives, criteria, subcriteria and alternatives" Saaty 1990

The following is a brief outline of the steps followed by the AHP. (Saaty, 1989, p 27)

1. Define the problem and specify a set of possible outcomes.
2. Structure the hierarchy from the overall viewpoint (the top levels in the hierarchy) through relevant intermediate levels to the final level.
3. Construct a pairwise comparison matrix of the relevant contribution or impact of each element on each governing objective or criterion in the adjacent upper level. In such a matrix of the elements by the elements, they are compared in a pairwise manner with respect to a criterion in the superior level. In comparing i, j elements, give a judgement that indicates the dominance as an integer. Thus, if the dominance does not occur in the (i, j) position while comparing the i th element with the j th element, then it is given in the (j, i) position as a_{ji} . The reciprocal would then be automatically assigned to a_{ji} .
4. Obtain the $n(n-1)/2$ judgements required for each matrix in step 3.

5. Having collected the pairwise comparison data and entered the reciprocals together with n unit entries down the main diagonal, the eigenvalue problem $Aw = \lambda_{\max} w$ is solved and consistency is tested: $(\lambda_{\max} - n)/(n-1)$ is divided by the corresponding average value for a matrix with random entries. This yields the consistency ratio C.R.
6. Steps 3, 4, and 5 are performed for the matrices on all levels and clusters in the hierarchy.
7. Hierarchical composition is now used to weight the eigenvectors by the weights of the criteria, and the sum is taken over all weighted eigenvector entries corresponding to those in the next lower level, and so on, resulting in a composite priority vector for the lowest level of the hierarchy.
8. Consistency is evaluated for the entire hierarchy by multiplying each consistency index by the priority of the corresponding criterion and adding over all such products. The result is divided by a similar expression using the random consistency index corresponding to the dimensions of each matrix weighted by the priorities as before. The ratio (C.H.) should be about 10% or less for acceptable overall consistency. Otherwise, the quality of the judgemental data should be improved, perhaps by revising the manner in which questions are posed to make the pairwise comparisons. If this fails to improve consistency, it is likely that the problem has not been

well structured; that is, similar elements are not grouped under a meaningful criterion. A return to step 2 is then required, although only the problematic parts of the hierarchy may need revision.

Appendix E:

The prototype consists of three modules; a knowledge based module, a database module, and a choice module. The interface is accomplished by using *Direct Access* as a control program. The main menu which welcomes the user on entering the system, as well as the main entry screen for the knowledge base are depicted on the following pages.

ADTEK MAIN MENU

- A) CASE ADOPTION MODULE
- B) PRODUCT ADOPTION MODULE
- C) FINAL SELECTION MODULE

ADTEK MAIN SCREEN

Welcome to ADTEK the CASE Adoption Advisor Expert System
You are going to be asked a series of questions about
your firm and the system will assist you in deciding
whether you should adopt CASE and which specific CASE
Tool you should adopt.

press any key to continue.

<u>RULES</u>	<u>FACTS</u>
TRAINING_COST > 1 AND RESTRUCT_COST > 1 AND PROCURE_COST > 1 THEN TOTAL_COST = TRAINING_COST + RESTRUCT_COST + PROCURE_ COST	TRAINING_COST = 6000 CNF 100 RESTRUCT_COST = 30000 CNF 100 PROCURE_COST = 50000 CNF 100 TOTAL_COST = 86000 CNF 100

1Help 2Go 3Whatif 4Variable 5Rule 6Set 7Edit 8Quit

Knowledge Base

The knowledge base consists of three blocks: Actions block, Rules block, and Statements block. The knowledge base is presented below.

Actions Block

Actions

Display

" Welcome to the CASE Adoption Advisor Expert System.
You are going to be asked a series of questions about your
firm and the system will assist you in deciding whether you
should adopt CASE and which specific CASE tool you should
adopt.

press any key to continue. ~"

CLS

FIND TOTAL_COST

DISPLAY "Total Costs are {TOTAL_COST}."

FIND PROJECTED_BENEFIT

DISPLAY "Projected Benefits are {PROJECTED_BENEFIT}."

FIND surplus

DISPLAY "Surplus is {surplus}."

FIND just

DISPLAY "Cost Benefit Analysis is {just}."

FIND recommendation

DISPLAY " My advice is that you {recommendation}.";

Rules Block

RULE 1a

IF Software_Backlog = yes

AND Maintain_Exist = yes

Then Production_Quality = low;

RULE 2a

IF Software_Backlog = no

OR Maintain_Exist = no

THEN Production_Quality = high;

RULE 3a

IF Productivity_Inc = yes

AND State_Of_Art = desired

THEN Empl_Need = high;

RULE 4a
IF Productivity_Inc = no
OR State_Of_Art = undesired
THEN Empl_Need = low;

RULE 5a
IF Production_Quality = low
AND Control = yes
AND Empl_Need = high
THEN Int_Factor = Strong;

RULE 6a
IF Production_Quality = high
OR Control = no
OR Empl_Need = low
THEN Int_Factor = weak;

RULE 7a
IF Comp_Adv = yes
AND Standards = yes
AND Documentation = yes
THEN Ext_Factor = strong;

RULE 8a
IF Comp_Adv = no
OR Standards = no
OR Documentation = no
THEN Ext_Factor = medium;

RULE 9a
IF Comp_Adv = no
AND Standards = no
AND Documentation = no
THEN Ext_Factor = weak;

Rule 10a
IF Int_Factor = Strong
AND Ext_Factor = Strong
THEN Need_Case = high;

RULE 10b
IF Int_Factor = strong
AND Ext_Factor = medium
THEN Need_Case = high;

RULE 11a
IF Int_Factor = Weak
OR Ext_Factor = Weak
THEN Need_Case = Low;

RULE 13a
IF Structured_Methods = yes
AND Programmer_Flexible = yes
AND Soft_Engineering = yes
THEN Staff_Expertise = adequate;

RULE 14a
IF Structured_Methods = no
OR Programmer_Flexible = no
OR Soft_Engineering = no
THEN Staff_Expertise = inadequate;

RULE 14b
IF Staff_Expertise = adequate
AND Resources = adequate
THEN capacity = good;

RULE 14c
IF Staff_Expertise = inadequate
OR Resources = inadequate
THEN capacity = poor;

RULE 14d
IF Capacity = poor
OR Support = poor
OR Sp_Level = level1
THEN Org_Ready = no;

RULE 15a
IF Project_Champion = yes
AND Steering_Committee = yes
AND Top_Man_Part = high
THEN Support = good;

RULE 15b
IF Project_Champion = no
OR Steering_Committee = no
OR Top_Man_Part = low
THEN support = poor;

RULE 15c
IF Capacity = good
AND Support = good
AND Sp_Level = level2
THEN Org_Ready = yes;

RULE 15d
IF Capacity = good
AND Support = good
AND Sp_Level = Level3
THEN Org_Ready = yes;

RULE 18a
IF Num_Suppliers = many
AND Search_Time = long
THEN Passive_Search = yes;

RULE 19a
IF Num_Suppliers = few
OR Search_Time = short
THEN Passive_Search = no;

RULE 20a
IF History_Exist = yes
AND Firm_supplier = good
THEN Org_Search = yes;

RULE 21a
IF History_Exist = no
OR Firm_supplier = suspect
THEN Org_Search = no;

RULE 22a
IF Num_Consultant = many
OR Num_Conferences = many
OR Num_Magazines = many
THEN Pro_search = yes;

RULE 23a
IF Num_Consultants = few
AND Num_Conferences = few
AND Num_Magazines = few
THEN Pro_Search = no;

RULE 24a
IF Site_Visits = many
AND Search_Time = long
THEN Active_Search = yes;

RULE 25a
IF Site_Visit = few
OR Search_Time = short
THEN Active_Search = no;

RULE 26a
IF Passive_Search = yes
OR Org_Search = yes
OR Pro_Search = yes
OR Active_Search = yes
THEN Solution_Search = adequate;

RULE 27a
IF Passive_Search = no
AND Org_Search = no
AND Pro_Search = no
AND Active_Search = no
THEN Solution_Search = inadequate;

RULE 36
IF Num_Alt = Several
OR Org_Ready = no
OR Company_New = yes
OR Solution_Search = inadequate
AND Cost_Tool = no
THEN recommendation = Wait;

RULE 37
IF Num_Alt = None
AND Solution_Search = Adequate
AND Need_Case = High
AND Org_Ready = Yes
AND Company_New = no
AND Cost_Tool = no
AND Just = favourable
THEN recommendation = Adopt;

RULE 38
IF Need_Case = low
OR Cost_Tool = yes
THEN recommendation = reject;

RULE 41
IF TRAINING_COST > 1
 AND RESTRUCT_COST > 1
 AND PROCURE_COST > 1
THEN TOTAL_COST = ((TRAINING_COST + RESTRUCT_COST + PROCURE_COST));

```

RULE 42
IF PRODUCT_SAVINGS > 1
  AND QUALITY_SAVINGS > 1
THEN PROJECTED_BENEFIT = ((PRODUCT_SAVINGS +
QUALITY_SAVINGS));

```

```

RULE 43
IF PROJECTED_BENEFIT > 1
  AND TOTAL_COST > 1
THEN SURPLUS = ((PROJECTED_BENEFIT) - (TOTAL_COST));

```

```

RULE 44
IF SURPLUS < 5000
THEN JUST = UNFAVOURABLE;

```

```

RULE 45
IF SURPLUS > 5000
THEN JUST = FAVOURABLE;

```

```

RULE 46
IF JUST = UNFAVOURABLE
THEN Recommendation = wait;

```

Statements Block

```

ASK TRAINING_COST: "What are projected training costs of CASE
Adoption?";
ASK RESTRUCT_COST: "What are expected restructuring costs due
to CASE?";
ASK PROCURE_COST: " What are procurement costs of CASE
adoption?";
ASK QUALITY_SAVINGS: "What are quality savings due to CASE
adoption?";
ASK PRODUCT_SAVINGS: " What are increased productivity savings
due to CASE?";
ASK Company_New: " Is your Company New?";
ASK Cost_Tool: " Is cost greater than 10% of Sales?";
ASK Software_Backlog: " Does a software backlog exist?";
ASK Maintain_Exist: " Is too much effort spent maintaining
existing Software?";
ASK Productivity_Inc: " Is there a need to increase
productivity?";
ASK State_Of_Art: " Do employees want to use latest
technology?";
ASK Control: " Is more control of the development process
needed?";
ASK Comp_Adv: " Is a competitive advantage needed?";
ASK Standards: " Are quality standards required?";
ASK Documentation: " Is detailed documentation necessary?";
ASK Resources: " Are resources adequate for a 6 month training
period?";
ASK Project_Champion: " Does a project champion exist?";

```

ASK Steering_Committee: " Does a steering committee exist?";
 ASK Top_Man_Part: " What is the level of management participation?";
 ASK Num_Suppliers: " How many suppliers were contacted?";
 ASK Search_Time: " Was enough time allocated to search?";
 ASK History_Exist: " Does the Firm have similar adoption experience?";
 ASK Num_Consultants: " How many consultants were contacted?";
 ASK Num_Conferences: " How many conferences were attended?";
 ASK Num_Magazines: " How many magazines were referenced?";
 ASK Site_Visits: " How many site visits were made?";
 ASK Num_Alt: " Can you identify alternatives to CASE?";
 ASK Structured_Methods: " Is staff aware of structured methods?";
 ASK Programmer_Flexible: " Are programmers flexible?";
 ASK Soft_Engineering: " Are staff aware of new software engineering techniques?";
 ASK Sp_Level: " What is the level of the software development process?";

CHOICES Num_Alt: several, none;
 CHOICES History_Exist, Software_Backlog, Productivity_Inc: yes, no;
 CHOICES Control, Comp_Adv, Standards: yes, no;
 CHOICES State_Of_Art: desired, undesired;
 CHOICES Top_Man_Part: high, low;
 CHOICES Documentation, Project_Champion, Steering_Committee: yes, no;
 CHOICES Num_Suppliers, Num_Consultants, Num_Conferences: many, few;
 CHOICES Num_Magazines, Site_Visits: many, few;
 CHOICES Search_Time: Yes, no;
 CHOICES Resources: adequate, inadequate;
 CHOICES Company_New, Cost_Tool, Maintain_Exist: yes, no;
 CHOICES Structured_Methods, Programmer_Flexible, Soft_Engineering: yes, no;
 CHOICES Sp_Level: Level1, level2, Level3;

Database Module

The database module allows the user to access and query a database of approximately 50 product records. A listing of the CASE tools contained in this database is presented below.

PRICE	PRODUCT NAME	VENDOR NAME
60000	INSTALL/1	Andersen Consulting
25000	DESIGN/1	Andersen Consulting
80000	EVOLVE/1	Andersen Consulting
50000	METHOD/1	Andersen Consulting
25000	HP ENCAPSULATOR	Hewlett Packard
45000	HP SOFTBENCH	Hewlett Packard
37000	IE-CBT	James Martin Assoc.
37000	IE-EXPERT	James Martin Assoc.
62000	RAD-EXPERT	James Martin Assoc.
47000	IE-METRICS	James Martin Assoc.
9000	IEW/Analysis Workstation	KnowledgeWare Inc.
9000	IEW/Design Workstation	KnowledgeWare Inc.
115000	IEW/Mainframe KC & Encyclopaedia	KnowledgeWare Inc.
9000	IEW/Construction Workstation	KnowledgeWare Inc.
209300	IEW/GAMMA	KnowledgeWare Inc.
9000	IEW/Planning Workstation	KnowledgeWare Inc.
5000	MF COBOL/2 Workbench	Micro Focus Inc.
60000	TELON/TEAMWORK	Pansophic Systems
285000	TELON/TEAMWORK	Pansophic Systems
100000	AD/VANTAGE	XL/Datacomp Inc.

	Teamwork	Cadre Technologies
10125	ADW	KnowledgeWare Inc.
925	ANATOOL	Advanced Logical Software
50000	APS DEVELOPMENT CENTER	Intersolv Inc.
70000	AS/SET	Systems Software Associates
8625	AUTO-MATE PLUS	Learmonth & Burchett
60000	AUTOCODE	Integrated Systems Inc.
25000	BACHMAN/Analyst	Bachman Info. Systems Inc.
1500	CA-DB:Architect	Computer Associates Intl
	CA-TELON	Computer Associates
2500	CASE:PM	Caseworks Inc.
12000	CASEworks/RT	Multiprocessor Toolsmiths
263000	CorVision	Cortex Corp.
21000	Software Through Pictures	Interactive Dev. Environment
5000	SQLlink-Plus	ECS Associates Inc.
4000	System Developer II	CADWARE Inc
9000	SYSTEMS ENGINEER	Learmonth & Burchett
8000	Visible Analyst Workbench	Visible Systems Corp.
92000	PROKIT WORKBENCH	McDonnell Douglas
32000	SCAN/COBOL	CDSI
500000	SNAP	Software Architecture & Eng.
6700	OBJECT PLUS	Easyspec Inc.
300000	PACBASE	CGI Systems, Inc.

8000	PC PRISM	INTERSOLV
2150	POSE	Computer Systems Advisers
16500	PowerCASE	Cognos Inc.
7500	Metavision	Applied Axiomatics Inc.
120000	METHODMANAGER	Manager Software Products
10125	IEW	KnowledgeWare Inc.
275000	LPS	American Management System
25000	LPS micro	American Management System
1600	MacAnalyst	Excel Software

Choice Module

A sample hierarchical model produced by *Expert Choice* is presented on the following page.

Sample Hierarchical Model in Expert Choice

SELECT "BEST" CASE TOOL

COMPLEX --- Complexity of the Tool

COST --- COST of Tool

DESIGN1 --- Design Tool called DESIGN1

OBSERVE --- Observeability of Benefits

REL.ADV --- Relative Advantage

TEAMWORK --- Development Tool by CADRE

TRIAL --- Trialability

L --- Local Priority: Priority Relative to Parent

G --- Global Priority: Priority Relative to Goal

Appendix F:

PRELIMINARY QUESTIONNAIRE (PROBE)

- 1) Company Name _____
- 2) Your Name _____
- 3) Your title/position at this company _____

- 4a) In which industry does Your company operate? [Electronic
Appliances, Semiconductors e.g.] _____

- b) Which Sector? Private____, Public____, Manufacturing____
Retail____, Service____, Other (Please Describe)_____

- 5) When was your company established? (year) _____
- 6) Number of employees _____
- 7) Approximate annual sales of your company in 1993: _____

- (Needs Recognition)
- 8) When did your Company perceive a need for CASE for the first
time? (year) _____

9) Who brought CASE technology to the company's attention?

___ President/CEO

___ Vice President

___ Information Director

___ Software Engineer

___ Other (Please explain) _____

10) Can you list the steps in the CASE adoption process that were followed at your firm.

11) How did the following factors contribute to the perception of a need for CASE? [Please indicate their relative importance e.g. 20%, 30% ...]

[] Large Backlog of Software Systems waiting to be developed

[] Majority of Software personnel effort spent on maintaining existing software systems

[] Lack of control over the Software Development Process

[] Need to increase software productivity

[] Desire to use state-of-the-art software technologies

[] Achieve competitive advantage

[] Other (please mention) _____

100%

12a) In your opinion, which shortcomings in the current software development practices, pertaining to quality, can you identify at your firm? [Please indicate their relative importance e.g 20%, 30%..]

- [] Inconsistent, incoherent, incomplete specifications and designs
- [] Developed product hard to operate
- [] Poor performance of resulting product
- [] Histories are unavailable
- [] No metrics are available to test overall system quality
- [] No effective documentation aids
- [] Unreliable and unmaintainable products
- [] Other [Please Describe]

100%

b) Did/will CASE help solve any of these problems? [Please explain]

c) Did/will any of these shortcomings prevent you from fully utilizing CASE? [Please explain]

13a) In your opinion, which shortcomings in the current software development practices pertaining to cost/productivity can you identify at your firm? [Please indicate their relative significance e.g 20% or 30%]

- ☐ Designs fail to provide for change
- ☐ Unreusable software and software designs
- ☐ Responsibilities, duties and accountability are poorly defined and controlled
- ☐ No provision to perform impact analysis in relation to specific parameters
- ☐ Completion criteria are poorly defined
- ☐ Excessive development costs and schedule
- ☐ Other [Please describe]

100%

b) Did/will CASE help solve any of these shortcomings? [Please explain]

c) Did/will any of these shortcomings prevent you from fully utilizing CASE? [Please explain]

14a) In your opinion, which organizational shortcomings in the current software development practices can you identify at your firm? [Please indicate their relative significance e.g. 20%, or 30% ..]

- ☐ No effective means to manage complexity
- ☐ Ineffective communication/feedback between development and personnel
- ☐ "Team" efforts are not coordinated
- ☐ Performance assessment done after the fact
- ☐ No assistance available for the resolution of conflicts or issues
- ☐ low effect of resources
- ☐ Other [Please explain]

100%

b) Did/will CASE help solve any of these shortcomings? [Please explain]

c) Did/will any of these shortcomings prevent you from fully utilizing CASE? [Please explain]

[Organizational Readiness]

- 15) How was it decided that your organization was "ready" for CASE?
(e.g. Was the staff trained in basic Software Engineering concepts such as structured analysis, or structured design? Were resources adequate to absorb a 3-6 months training period?)

- 16) In terms of the software development process, at what level of maturity would you place your organization?

- 1) Little formalization - tools informally applied to process. Ad hoc procedures.
- 2) Achieved a stable process with a repeatable level of statistical control. i.e quantitative performance measures identified.
- 3) Achieved foundation for major and continuing progress.
- 4) Substantial quality improvements - comprehensive process measurement.
- 5) Major quality and quantity improvements.

Comment: _____

17) What type(s) of CASE tools did you adopt?

- ☐ Code Generators
- ☐ Analysis/Design Toolkits
- ☐ Re-engineering Tools
- ☐ Strategic Planning Tools
- ☐ IPSE open or closed
- ☐ Other [Please describe]

18a) What was the approximate cost (hardware and software) of the CASE tool adopted?

b) What was the approximate training and personnel cost?

19) What hardware platform are you running CASE on?

- ☐ PC
- ☐ LAN or other network
- ☐ Mainframe
- ☐ Other [Please describe]

20) When did your Company decide to adopt CASE? i.e. commit financial resources for CASE. (year) _____

- 21) Was the decision to adopt CASE made by one person or many?
If more than one, how was any conflict resolved?

- 22) Please indicate the extent of participation of the various players in the CASE adoption decision.

Top Management High___ Medium___ None___

Middle Management High___ Medium___ None___

Software Engineers High___ Medium___ None___

Other [Please describe]

[Solutions Search]

- 23) Did you consider any alternatives to CASE? If so, which? If not, skip the question.

[] Object Oriented

[] Rapid Applications Development

[] Prototyping

[] Other [Please describe]

24) To search the CASE market, which of the following search techniques were used? (Please tick)

- ☐ Waiting for unsolicited vendors to appear.
- ☐ Using past experience to identify a suitable vendor.
- ☐ Gathering of information by hiring consultants, attending conferences and trade shows, or reading professional magazines.
- ☐ Inviting potential vendors to submit options.
- ☐ Site visits to users and/or vendors of CASE.
- ☐ Other [Please describe]

[Solutions Evaluation]

25a) How did you make the final choice of vendor and product? (eg. relied on engineers totally, relied on consultants, used criteria that are applied for all acquisitions.)

b) Can you identify any rules of thumb that you may have used to make the choice?

26) Which of the following factors did you consider in the CASE adoption decision? [Please indicate the relative importance of each factor e.g. 20%, 30% ..]

- ☐ Cost/Benefits - tangible
 - ☐ Cost/Benefits - intangible
 - ☐ Promise of better Productivity
 - ☐ Promise of improved Software Quality
 - ☐ Effects on the Organization, bad and good
 - ☐ Organizational Readiness
 - ☐ Competitive Advantage
 - ☐ Promise of better control of Software Development Process
 - ☐ Reputation in the Industry
 - ☐ Employee satisfaction/dissatisfaction
 - ☐ Other [Please describe]
- 100%

- 27) If you have used specified criteria in selecting the CASE product, please identify them. If more than one were used please indicate their relative weights. [eg. 20%, 30% ..]

☐ Methodology
☐ Customization and Extensibility
☐ Desktop Publishing
☐ Product Documentation
☐ Validation [consistency checking and method for reporting discrepancies]
☐ Vendor Viability
☐ Common Repository Concept [common data representation]
☐ Work station/ Mainframe Integration
☐ User Interface
☐ Cooperative Work Environment
☐ Platform
☐ SDLC Integration
☐ Cost
☐ Ease of Learning
☐ Other [Please describe]

100%

- 28) Did you use a Capital Budgeting Technique?
If yes, which of the following did you use?

☐ Payback Period
☐ Accounting Rate of Return (ARR)
☐ Net Present Value Method (NPV)
☐ Internal Rate of Return (IRR)
☐ Profitability Index (PI)
☐ Other [Please describe]

- 29) Did you use any software to assist in your selection and adoption decision?

a) If yes, which?

- b) If no, in retrospect, is there any information and tool that could have been used in the decision making process?

- 30) Did you try to determine the degree to which the adoption decision was correct? If so, how?

- 31) In retrospect, if you were to go through the adoption decision again, what would you do differently? Which factors would you emphasize or de-emphasize? Which new factors would you include in the decision?

Appendix G:

Semi Structured Interview Schedule

Preamble: The purpose of this interview is to elicit the necessary information to create a knowledge based aid to assist in the CASE adoption decision. As such, any information or comments solicited or not, are welcome.

Section A -Company Profile

The questions in this section identify the respondents, and elicit information about the firm which may affect the CASE adoption decision process.

- 1) Company Name _____
- 2) Your Name _____
- 3) Your title/position at this company _____

- 4a) In which industry does Your company operate? [e.g. Electronic Appliances, Semiconductors etc.] _____

- b) Which Sector? Private____, Public____, Manufacturing____
Retail____, Service____, Other (Please Describe)_____

- 5) When was your company established? (year) _____
- 6) Number of employees _____
- 7) Approximate 1993 sales of your company: _____

Section B - CASE Need

The following questions seek to establish how a firm should decide if CASE is the right solution at this time.

8) Who brought CASE technology to the company's attention?

___ President/CEO

___ Vice President

___ Information Director

___ Software Engineer

___ Other (Please explain) _____

9) When did your Company perceive a need for CASE for the first time? (year) _____

And what were the main reasons?

10a) Can you list the steps in the CASE adoption process that were followed at your firm?

b) In retrospect, are there any steps that should have been taken?

11) Are the following reasons for CASE adoption valid.

Large Backlog of Software Systems waiting to be developed Agree ___ Disagree ___

Majority of Software personnel effort spent on maintaining existing software systems Agree ___ Disagree ___

Lack of control over the Software Development Process Agree ___ Disagree ___

Need to increase software productivity Agree ___ Disagree ___

Desire to use state-of-the-art software technologies Agree ___ Disagree ___

Achieve competitive advantage Agree ___ Disagree ___

Other (please mention) _____

12) Are the following alternatives to CASE valid?

[] Object Oriented

[] Rapid Applications Development

[] Prototyping

[] Other [Please describe]

Reasons: _____

Section C - Organizational Readiness

The questions in this section seek to ascertain how it may be determined if the Firm is ready for CASE.

13) Are The following conditions necessary for CASE Adoption?

Staff must be competent in Software Engineering techniques such as structured analysis and structured design. Agree ____ Disagree ____

A project champion must exist. Agree ____ Disagree ____

A steering committee must have been established. Agree ____ Disagree ____

Resources must be adequate to Absorb a 3-6 month training period. Agree ____ Disagree ____

Other.

Comments: _____

14) In terms of the software development process, and Humphreys 5 levels of development, attainment of level 3 is necessary for the successful adoption of CASE technology.

level 1) Little formalization - tools informally applied to process. Ad hoc procedures.

level 2) Achieved a stable process with a repeatable level of statistical control. i.e quantitative performance measures identified.

level 3) Achieved foundation for major and continuing progress.

level 4) Substantial quality improvements - comprehensive process measurement.

level 5) Major quality and quantity improvements.

Agree ____ Disagree ____

- 15) What is the cutoff point for # of programmers, beyond which CASE becomes cost effective and therefore necessary.

Section D - Product Profile

The questions in this section seek to determine the product considerations, including search and selection, that affect the CASE adoption decision process.

- 16) Which of the following criteria are important features of the product profile and should be considered when selecting a specific CASE product? [Please Rank if Possible?]

- ☐ Methodology
- ☐ Product Documentation
- ☐ Validation [consistency checking and method for reporting discrepancies]
- ☐ Vendor Viability
- ☐ Common Repository Concept
- ☐ User Interface
- ☐ Platform
- ☐ SDLC Integration
- ☐ Cost
- ☐ Ease of Learning
- ☐ Other [Please describe]

17) To search the CASE market adequately, which of the following search techniques are necessary? [Please Rank if Possible]

[] Waiting for unsolicited vendors to appear.

[] Using past experience to identify a suitable vendor.

[] Gathering of information by hiring consultants, attending conferences and trade shows, or reading professional magazines.

[] Inviting potential vendors to submit options.

[] Site visits to users and/or vendors of CASE.

[] Other [Please describe]

18a) How did you make the final choice of vendor and product? (e.g. relied on engineers totally, relied on consultants, used criteria that are applied for all acquisitions.)

b) Can you identify any rules of thumb that you may have used to make the choice?

19) Rogers identifies 5 attributes of innovations by which innovations can be described and compared. In relation to CASE products, can you weight these criteria in terms of importance? [e.g. 20%, 30%, 30%, 15%, 5%]

[] Relative Advantage

[] Compatibility

[] Complexity

[] Trialability

[] Observability

Section E - General

The questions in this section solicit miscellaneous information that may affect the adoption decision process

- 20) Top Management participation in the CASE adoption decision process must be medium or high for adoption to be successful.

Agree ____ Disagree ____

- 21) Should the decision to adopt CASE be made by one person or many?

If more than one, how should any conflict be resolved?

- 22) Did you use a Capital Budgeting Technique?
If yes, which of the following did you use?

[] Payback Period
[] Accounting Rate of Return (ARR)
[] Net Present Value Method (NPV)
[] Internal Rate of Return (IRR)
[] Profitability Index (PI)
[] Other [Please describe]

- 23) Did you use any software to assist in your selection and adoption decision?

a) If yes, which?

- b) In retrospect, is there any information and tool that could have been used in the decision making process, in addition to what you may have used?

- 24) Did you try to assess the degree to which the adoption decision was correct? If so, how?

- 25) In retrospect, if you were to go through the adoption decision process again, what would you do differently? Which factors would you emphasize or de-emphasize? Which new factors would you include in the decision?

END

0 1 - 0 6 - 9 5

FIN