

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

A

ARCHITECTURE OF SYSTEMIC KNOWLEDGE FORMALIZATION

BY

ZHANYANG ZHANG

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1995

UMI Number: 9530939

UMI Microform 9530939
Copyright 1995, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI

**300 North Zeeb Road
Ann Arbor, MI 48103**

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

April 19, 1995
Date

Miriam R. Tausner
Chair of Examining Committee

April 19, 1995
Date

Prof. Stanley Habib
Executive Officer

George Klir
Robert A. Orchard
Divyendu Sinha

Supervisory Committee:

Dr. Miriam R. Tausner (Chairperson)
Dr. Stanley Habib (Executive Officer)
Dr. Robert A. Orchard
Dr. George J. Klir
Dr. Divyendu Sinha

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT**ARCHITECTURE OF SYSTEMIC KNOWLEDGE FORMALIZATION****BY****ZHANYANG ZHANG**

Adviser: Professor Miriam R. Tausner and Professor Robert A. Orchard

Knowledge engineering is the process of building knowledge based systems. Knowledge elicitation is usually considered to be the major bottleneck in this process. In this dissertation we developed a domain-independent framework for modeling domain-specific knowledge that actively assists in the elicitation of knowledge.

Recent advances in the general systems theory and systemic knowledge engineering have led to the belief in the existence of knowledge structures that can intrinsically capture the expertise of a domain expert. Herein we define and formalize what these knowledge structures are and investigate their usage in modeling the problem solving process of a domain expert. This approach integrates an epistemological systems hierarchy, especially a meta-system definition into systemic knowledge engineering and leads to the formalization of systemic knowledge hierarchy.

In our formalism, knowledge consists basic knowledge and meta-knowledge. Basic knowledge is structured information. Basic knowledge has been well defined in the

literature. Meta-knowledge imposes further structure on basic knowledge as well as directly structures information to increase the knowledge content of the knowledge hierarchy. It is the first time in this dissertation that meta-knowledge is formalized using systemic knowledge structures. This formalism is powerful enough to formalize the actual knowledge of domain experts in numerous applications. It also helps organize the elicited knowledge and aids in the subsequent knowledge elicitation.

key words: general systems, systemic knowledge engineering, problem solving roadmap, basic knowledge, meta-knowledge, knowledge structures, canonical form, knowledge elicitation, knowledge formalization and knowledge representation.

ACKNOWLEDGMENT

I am grateful to the many people who have helped me during my writing of this dissertation. My special thanks go to Robert A. Orchard who has been my academic advisor for many years. He introduced me to the world of systems science and systemic knowledge engineering. His intellectual spirit sparked many of the ideas which formed an important part of this dissertation. This dissertation was greatly influenced by George J. Klir's work on general systems theory, particularly his masterpiece, "*Architecture of Systems Problem Solving.*" I would like to express my deepest appreciation to Miriam R. Tausner who became my dissertation advisor at the most difficult and crucial time in my academic life. I could not have finished my dissertation without her supervision and encouragement. It was her invaluable guidance, rigorous mathematical thinking and consistency that helped me transform the dissertation research from a collection of ideas into a mathematical formalism. She sacrificed many weekends and evenings to work with me on the dissertation, otherwise it would have been impossible for me to finish the dissertation while working full time. Special thanks are due to Divyendu Sinha for his constructive suggestions and criticisms. The development of the dissertation benefited from the discussions with Norman J. Bashias. He taught me all the Greek symbols used in the dissertation. I should express my lifelong gratitude to my parents who passed on to me their value and attitude toward life. They might not understand my dissertation, but they believe that whatever their son is doing is important to them. Finally I must thank my wife, ShanShan, my children, Xiaolin (Lynn) and Xiaochen (Richard) for their support and understanding. They fill my life with joy and hope.

Table Of Contents

Introduction	1
1 Introduction to Systemic Knowledge Engineering	4
1.1 Information and Knowledge.....	4
1.2 Domain Knowledge and Systemic Knowledge.....	5
1.3 Systemic Knowledge Engineering	10
2 Systemic Knowledge Hierarchy	13
2.1 Introduction to Epistemological General Systems Hierarchy.....	13
2.2 Basic Knowledge and Complex Knowledge	15
3 Systemic Knowledge Formalism.....	20
3.1 The Epistemological Systemic Knowledge Hierarchy	20
3.2 The Problem Solving Roadmap and Basic Knowledge Structures	25
3.3 Canonical Form for a Meta-Knowledge Structures	29
3.4 Problem Solving Control Constructs	34
3.5 Communication Constructs	36
4 Case Study - Cooperative Problem Solving	38
4.1 Cooperative Problem Solving	39
4.2 Modeling the Expertise of Each Expert	40
4.3 Modeling the Cooperation Between the Experts	42
4.4 Developing the Model.....	43
4.5 Using the Model to Simulate the Problem Solving Strategies.....	59
4.6 Summary	60
5 Open Areas for Future Research	61
Bibliography	64

List of Illustrations

Fig. 1.1 Basis for Systemic Knowledge Engineering	11
Fig. 2.1 Overview of Klir's Epistemological Systems Hierarchy	14
Fig. 2.2 Problem Solving Roadmap	16
Fig. 2.3 Overview of Systemic Knowledge Hierarchy	19
Fig. 3.1 Problem Solving Roadmap and Basic Knowledge Structures	25
Fig. 4.1 Cooperative Problem Solving - a Car Design Problem	40
Fig. 4.2 Roadmaps for Group Experts - a Complex Plan	41
Fig. 4.3 The Source Systems of the Model	46
Fig. 4.4 The Data Systems of the Model	48
Fig. 4.5 The Behavior Systems of the Model	51
Fig. 4.6 Iteration between Data and Behavior System	52
Fig. 4.7 Iteration between Data and Structure System	54
Fig. 4.8 The Structure Systems of the Model	57

INTRODUCTION

The area of knowledge-based systems is one of the most active research areas in AI. It is increasingly gaining ground in a wide range of applications, from integrated computer aided design and manufacturing systems to all kinds of decision support systems in the financial world. Currently knowledge engineering is loosely defined as the process of building a knowledge based system . In the literature it is often emphasized that knowledge elicitation is the major bottleneck in the whole process.

In this dissertation, knowledge engineering is viewed as having at least two branches, domain knowledge engineering and systemic knowledge engineering. At the present stage, research in the field of knowledge engineering is heavily concentrated on domain knowledge engineering, with emphasis on domain specific information and its representation, typically in expert systems. It is well known that expert systems derive much of their power from the actual knowledge of an expert. However, the power which can be derived from domain independent structures has been underdeveloped by knowledge engineering researchers. Based on recent research in systems theory, it seems there exist some kinds of knowledge structures which can capture (structure) the expertise of a domain expert in a very natural way. There is also strong evidence that domain experts use these structures intuitively (outside of their awareness). The motivation for this research is to define and formalize knowledge structures at the meta-level and to investigate the use of these structures to model the problem solving process. This formalism can also help knowledge engineers to organize the knowledge which has been elicited and to better direct their knowledge elicitation efforts. Therefore, the formalism can help knowledge engineers to guide the knowledge elicitation process and to make it more efficient.

This work takes a unique approach, a systemic approach, to knowledge engineering problems. We call this approach systemic knowledge engineering. It integrates an epistemological hierarchy of systems, and especially a meta-system definition, into knowledge engineering to form an epistemological knowledge hierarchy. It builds upon the recent research on general systems [Klir, 1985] and systemic knowledge engineering [Orchard *et al*, 1987, 1988, 1989, 1990] to formalize basic knowledge and meta-knowledge with respect to a problem solving process.

The core of this research is to define a formalism which, in many cases, appears to be "sufficient" to formalize the actual knowledge of an expert. Work by Orchard and Tausner [1987a and 1987b] has given precise definitions of basic knowledge structures and the concept of a problem solving roadmap to model experts' problem solving strategies. In this dissertation, a new formalism is developed to extend the basic knowledge structures to meta-knowledge structures. This is the first time systemic meta-knowledge has been formalized. A canonical form is then defined to represent the formalized systemic knowledge and to precisely express problem solving trajectories on the problem solving roadmap. Based on this new formalism, formalized knowledge can be expressed in pseudo code which is close to a computer representation.

A case study is presented in the dissertation to show how to model and formalize the expertise of a group of domain experts with respect to solving a car design problem. The need for the research conducted in this dissertation was documented in some early papers on knowledge engineering [Orchard *et al*, 1987a, 1987b, 1988, 1989, 1990].

This dissertation is organized into five chapters. In Chapter 1, *Introduction To Systemic Knowledge Engineering*, basic concepts and definitions about information and knowledge are introduced for the purpose of defining systemic knowledge engineering. Specifically,

the concepts of domain knowledge and systemic knowledge are discussed, their differences and the relationship between them. Systemic knowledge engineering is defined in a way to emphasize the systemhood aspect of the knowledge-based system building process.

In Chapter 2, *System Knowledge Hierarchy*, an epistemological general systems hierarchy [Klir, 1985] is introduced as the foundation of systemic knowledge formalization.

Following that, an epistemological knowledge hierarchy is outlined which is derived from the general systems hierarchy. We also discuss the definitions of basic-knowledge, meta-knowledge, and complex knowledge.

In Chapter 3, *Systemic Knowledge Formalism*, a new framework for systemic knowledge formalization is defined which encompasses the definitions of an epistemological systemic knowledge hierarchy, four basic knowledge structures, a canonical form for meta-knowledge structures, problem solving control constructs, and problem solving communication constructs.

In Chapter 4, *Case Study: Cooperative Problem Solving*, a case study of cooperative problem solving, a car design problem, is presented to show how the formalism can be used to formalize the expertise of a group of experts and model their problem solving process.

In Chapter 5, *Open Areas for Future Research*, the dissertation is concluded with a brief summary and the discussion of some open areas for future research, such as, formalizing knowledge of human experts in the perspective of problem solving with uncertainty.

CHAPTER 1 INTRODUCTION TO SYSTEMIC KNOWLEDGE ENGINEERING.

1.1 Information and Knowledge

Knowledge and information are probably two of the most frequently used words in knowledge engineering without a universally agreed upon definition. The two words are related but different. Looking at a dictionary [*Webster's Ninth New Collegiate Dictionary*] we find definitions like:

"Information is the communication or reception of knowledge or intelligence."

and

"Knowledge is the fact or condition of knowing something with familiarity gained through experience or association."

and

"Fact is a piece of information presented as having objective reality."

One can continue to make distinctions like "What is objective reality?". You can go on and on, and finally find yourself in a discussion of philosophical issues which may never end.

Instead of arguing philosophically, I attempt to take a practical approach and consider information as a message perceived through our five senses (instruments can be considered as an extension of human senses). Information is considered to be given and is primitive. It is not defined in this dissertation.

Knowledge is related to information, but it is not simply the accumulation of information. Rather, it is more related to the structure of information. Parsay and Chignell stated : "Knowledge is information about the world which allows an expert to make decisions" [Parsey and Chignell, 1988, p.119]. In the information age, the world is filled with all kinds of information. It is crucial for an expert to have the right information, at the right time, at the right place (situation) to make a right decision. I believe that the right information refers to information structured in such a way that it is applicable to the problem to be solved or decision to be made.

Information can contribute to one's knowledge about some aspect of a universe of discourse if the information is structured according to some criteria. It is in this context that Orchard stated: "basic knowledge is structured information" [Orchard *et al*, 89]. This definition was adopted and complied with here because of the key word "structure" which best conveys the broad meaning of selection, labeling, classification, storage and analysis. Based on this definition a systemic knowledge formalism is developed in Chapter 3.

1.2 Domain Knowledge and Systemic Knowledge

As mentioned in the introduction, the central theme of this dissertation is systemic knowledge formalization . The term systemic knowledge is used here in contrast to the term domain knowledge. *Systemic Knowledge* is information structured through a set of systemic knowledge structures. *Domain Knowledge* is information structured through a set of domain specific knowledge structure other then systemic knowledge structures.

The relation between domain knowledge and systemic knowledge is viewed in the same sense as Klir's view of the relation between traditional sciences (such as physics, chemistry, medicine etc...) and systems science.

Klir made a comment following a common definition of system:

"... the term 'system' stands, in general, for a set of some things and a relation among the things.

... a system, say system S, is thus an ordered pair $S=(A,R)$, where A denotes a set of relevant things and R denotes a relation among the things in set A. Such a conception of a system is too general and, consequently, of little pragmatic value. To make it pragmatically useful, it has to be refined in the sense that specific classes of ordered pairs (A,R), relevant to recognized problems, must be introduced, such classes can basically be introduced by one of two fundamentally different criteria:

- a. by a restriction to systems which are based on certain kinds of things.
- b. by a restriction to systems which are based on certain kinds of relations...." [Klir, 1985, p.4]

In his view, later on, criteria (a) leads to the classification of traditional sciences into disciplines and specialized domains. Each of them focuses on the study of certain kinds of things without committing to any particular kind of relations. Criteria (b) leads to classifying systems science as a science which focuses on the study of specific kinds of relations with no commitment to any particular kind of things on which the relations are

defined. Klir also points out the difference between traditional science and systems science as well as the relationship between the two. He argues in the same book:

"From the standpoint of the disciplinary classification of traditional science, systems science is clearly cross-disciplinary. There are at least two implications of this fact. First, systems science knowledge and methodology are directly applicable, at least in principle, in virtually all disciplines of traditional science. Secondly, systems science has the flexibility to study relational properties of such systems and the associated problems which include aspects derived from any number of different disciplines and specialization of traditional science. Such cross disciplinary systems and problems can thus be studied as wholes rather than collections of the disciplinary subsystems and sub-problems."
[Klir, 1985, p.7]

Based on the above argument, Klir considers systems science as a new dimension in science rather than a new science comparable with the other sciences. Speaking of a new dimension in science, he divides the history of science into three periods:

- i. prescientific period -- The ancient scientist/philosopher such as Aristotle, who was able comprehend almost all knowledge available in his time. There is no disciplinary specialization.
- ii. one-dimensional science -- characterized by the evolution of a highly complex hierarchy of disciplinary specialization which leads to the birth of each traditional science with ever increasing depth of knowledge and narrowness in interest and competence.

- iii. two-dimensional science -- characterized by the emergence of system science.

If knowledge is viewed as consisting of information and structures, knowledge can be defined as an ordered pair $K=(I,S)$. I denotes a set of information and S denotes a set of knowledge structures defined on I , which is used to structure the information in set I .

Accordingly, two different restriction criteria could be introduced:

- a. by a restriction to knowledge which is based on certain kinds of information.
- b. by a restriction to knowledge which is based on certain kinds of knowledge structures.

Criteria (a) leads to classifying knowledge as different domain knowledge which focuses on the study of information and phenomenon in a particular domain without commitment to any kind of knowledge structures. Criteria (b) leads to classifying knowledge as systemic knowledge which focuses on the study of knowledge structures without commitment to any specific kind of domain information which the knowledge structures are applied to.

It is obvious that systemic knowledge has its origin in systems theory. Rosen [1986] proposed the concepts "systemhood" and "thinghood", and defined "systemhood" as an independent aspect of things, independent of their "thinghood". He developed this idea along the same lines followed earlier in mathematics, in the extraction of the concept of a set from experience with particular aggregates. Systemic knowledge is the knowledge about "systemhood" of the objects and domain knowledge is the knowledge about "thinghood" of the objects which are under investigation.

The set of systemic knowledge structures are applicable virtually to all domain information. Each domain perhaps has its own knowledge structures to structure domain specific information, but those domain specific knowledge structures can be converted to systemic knowledge structures. There may be some domain knowledge structures which cannot be converted, but it is difficult to find an example of even one.

Systemic knowledge is derived from domain knowledge through an inductive modeling process. But it is higher level knowledge compared to domain knowledge in the sense that it is domain independent and applicable virtually to all domains. For example, the same problem solving strategy may be used to solve a system design problem in different domains.

If systemic knowledge is considered as a new dimension of knowledge rather than a new kind of knowledge comparable with other domain knowledge, then knowledge engineering development can be divided into three periods:

- i. preknowledge period -- represented by an exhaustive search through the problem space. Emphasis on general searching and problem solving methods.
- ii. one-dimensional knowledge period -- represented by expert systems, focusing on the importance of domain specific knowledge and heuristic search.
- iii. two-dimensional knowledge period -- emergence of systemic knowledge engineering, focusing on knowledge structures and methodologies which are useful in all disciplines or domains.

1.3 Systemic Knowledge Engineering

Currently knowledge engineering is loosely defined as the process of building an expert system, and the knowledge engineer as the expert system builder [Orchard and Tausner, 1987a]. At the present stage, this process is still considered an "art" because of the lack of scientific specification, which is not unusual in any fast developing fields. That is the reason why those fields are often identified as being at "the state of the art". Recent research has focused on using systemic knowledge to transform human expert knowledge into a computer based representation [Orchard and Tausner, 1987a, 1987b; Orchard, Reese and Tausner 1989; Orchard, 1990]. A definition of systemic knowledge engineering is now given: *systemic knowledge engineering is the process of building a knowledge based system using systemic knowledge structures, namely, systemic knowledge structure driven knowledge elicitation, knowledge formalization and knowledge representation.*

Systemic knowledge engineering is a merging of cognitive science, behavioral science, systems science, mathematics, and computer science. The ultimate goal is to reduce the "art" process to a procedural, algorithmic and scientifically based process.

Further research [Orchard, Reese and Tausner, 1989] has lead to the identification of three areas: *knowledge elicitation, knowledge formalization, and knowledge representation*, as fundamental to the continued development of systemic knowledge engineering (Fig. 1.1).

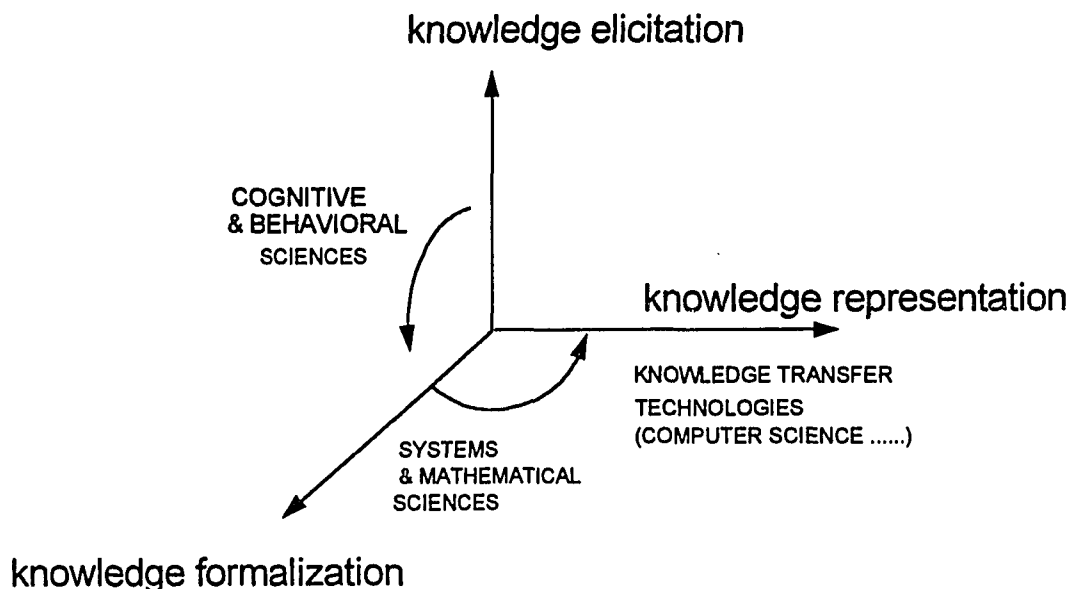


Fig. 1.1 Basis for Systemic Knowledge Engineering.

Such decomposition is quite natural. Because the process of building an expert system requires that knowledge be, first, elicited from the expert; second, put into an appropriate form which captures and organizes the knowledge in a formal way; and third, translated from its formal representation into a computer-based representation for automating the expert system. An extensive argument can be found in the papers cited earlier in the section for making such a decomposition. It is important to point out that frequently people do not distinguish between knowledge formalization and knowledge representation. Note that knowledge may be elicited and formalized but may not be representable by current computer technologies due to the limited power of a Turing machine (currently all computers are equivalent to Turing machines in computability measure). Take as a simple example, the knowledge of keeping balance, when one learns

how to ride a bicycle. A biological-computer may be needed to represent such kinetical knowledge. This fact could be strong evidence to support the separation of knowledge formalization from its representation.

Several established sciences and technologies are applied to form the framework for the three bases of systemic knowledge engineering [Orchard, Reese and Tausner 1989]. Among them, neurolinguistic programming, a field born out of the convergence of psychology and linguistics, helps form the initial content of systemic knowledge elicitation. General systems theory developed by system theorist Klir [1969, 1985] forms the initial content of systemic knowledge formalization. Many kinds of the well-known representation schemes from computer science can be used for systemic knowledge representation: rule-based systems, frame-based systems, semantic nets, and finite state automata [Tausner, 1986].

CHAPTER 2 SYSTEMIC KNOWLEDGE HIERARCHY

2.1 Introduction to Epistemological General Systems Hierarchy

Through an inductive modeling process, George J. Klir [1985] introduces an epistemological hierarchy of general systems which we refer to as Klir's hierarchy in this dissertation. In his view, inductive modeling is the interaction between investigator and investigated object in a certain environment. What we study is a model of the real world object, not the object itself. The model of the object is what we refer to as a "system".

Klir makes the comment:

"... the term system is thus always viewed as an abstraction - or an image - of some aspects of the object and not as a real thing."

It is actually impossible to study a real object without abstraction and ignorance¹. The significance of the distinction between the notions of object and its model as a system is well expressed by Ashby [1956]:

"At this point we must be clear about how a 'system' is to be defined. Our first impulse is to point at the pendulum and to say 'the system is that thing there'. This method, however, has a fundamental disadvantage: every material object contains no less than an infinity of variables and therefore of possible systems. The real pendulum, for instance, has not only length and position, it has also mass, temperature, electric conductivity, crystalline structure, chemical impurities, some radioactivity, velocity, reflecting power, tensile strength, a

¹ Here, ignorance means that some of the information about the real object is omitted.

surface film of moisture, bacterial contamination, an optical absorption, elasticity, shape, specific gravity, and actually the attempt is never made. What is necessary is that we should pick out and study the facts that are relevant to some main interest that is already given. ... The system now means, not a thing, but a list of variables."

It is in this context that Klir defined the lowest level of his hierarchy as *source systems*, systems defined by a set of variables and a set of potential values which describe a set of selected attributes of the real object. Based on source systems, *data systems*, *generative systems*, *structure systems* and *meta system* are defined in increasing order of knowledge content. For further details, see [Klir, 1985].

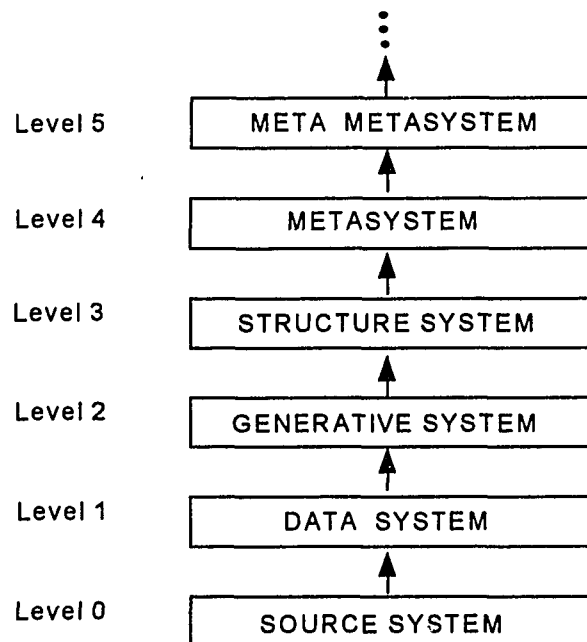


Fig. 2.1 Overview of Klir's Epistemological Systems Hierarchy.

2.2 Basic Knowledge and Complex Knowledge

Klir's works on General Systems Theory [1969, 1972, 1985] are chosen as the theoretic foundation of systemic knowledge formalization for two reasons. The first is that the inductive modeling aspects of Klir's hierarchy naturally model the ways experts view an object of interest. Secondly, Klir's hierarchy provides us with a very rich source of potential knowledge structures. To explain these reasons in another way, consider the following quote from Orchard and Tausner's [1988] paper:

"A fundamental problem of the knowledge engineering is the elicitation and formalization of knowledge from a domain expert. Most successful knowledge engineers appear to have an extremely broad range of interests and a natural curiosity which allow them to pick up 'buzz words,' nonverbal cues, and other neurolinguistic information, to structure the information of the domain expert. It appears that currently this is accomplished at an intuitive level outside the awareness of the knowledge engineer and accomplished by metaphorical analogy or some aspect of inductive modeling. The ways in which an expert views a system of interest are naturally captured by the various general systems epistemological levels as defined by Klir. "

For operational reasons, in practice, a morphological grid called the *problem solving roadmap* (fig. 2.2) is used to model an expert's expertise. This roadmap was introduced by Robert A. Orchard in the 1970's at Bell Laboratories.

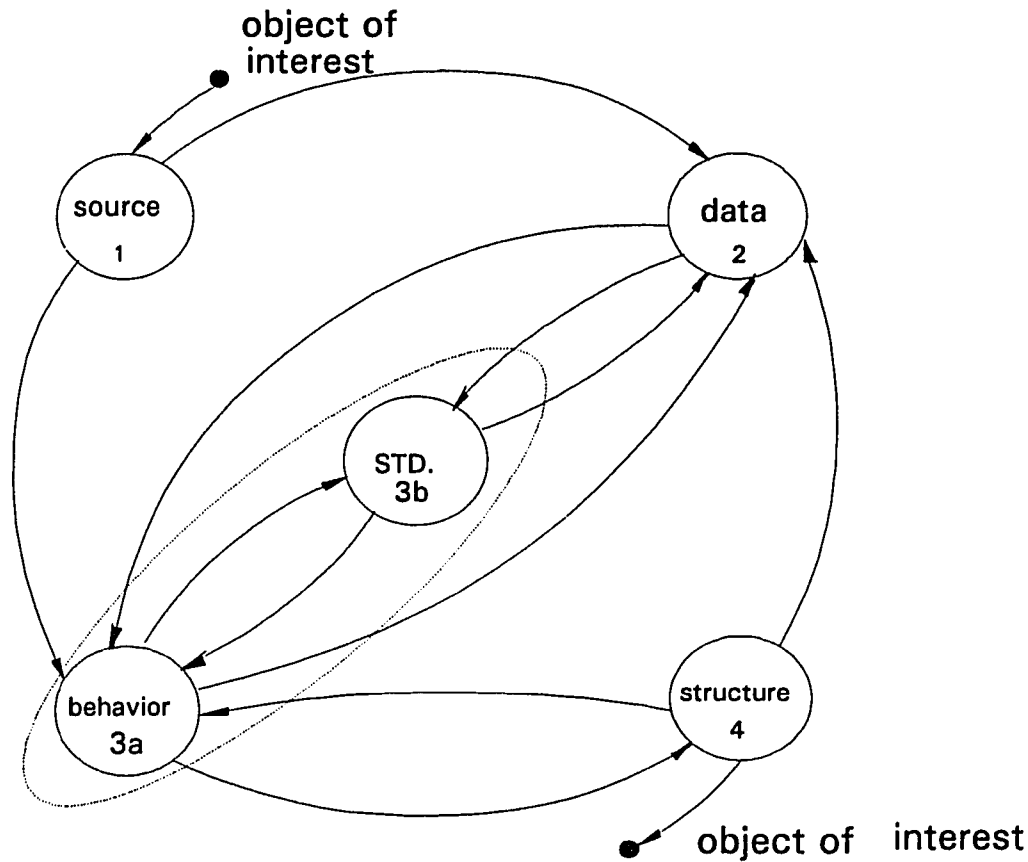


Fig. 2.2 Problem Solving Roadmap.

Orchard utilizes four basic general systems definitions (represented by the nodes or node clusters on the roadmap) as basic knowledge structures: *source system*, *data system*, *behavior system* and *structure system*. These knowledge structures represent the basic ways information can be structured. Consequently, he defines *basic knowledge* as structured information. Each of those basic knowledge structures is considered to be at a different epistemological level and is ordered by increasing knowledge content. By tracing the movements from one node to the next (a methodology used to make such a movement is represented by an arc on the roadmap), the roadmap provides ways of understanding the reasoning strategies and methodologies invoked by an expert to process

basic knowledge when solving a complex problem. Finally that will lead us to further structure the basic knowledge into meta-knowledge (the knowledge of how to sequence and process basic knowledge).

In the current literature, there is a growing awareness of the existence of multi-layers of knowledge, notably, Newell's physical symbol system [Newell, 1982], Dreyfus and Dreyfus's five stages of human knowledge learning process [Dreyfus and Dreyfus, 1986] and Breuker's four-layer model of human expertise [Breuker, 1987]. The words we hear frequently are the words such as, deep knowledge [Davis,1982], meta rules, meta reasoning, and meta planning [Hayes-Roth *et. al.*,1983]. Meta-knowledge is the name for all in general. In most cases the word meta-knowledge is used in an intuitive sense, such as knowledge about knowledge, without precise definition.

The term "meta-knowledge" is defined in this dissertation along the same lines as Klir defines "metasystem". In Klir's hierarchy, meta-system is well defined both terminologically and mathematically [Klir,1985]. He first examines the three major meanings of the prefix "meta" in Greek:

- " i. 'meta X' is a name of something that occurs after X. i.e., X is a prerequisite for meta X;
- ii. 'meta X' indicates that X changes and is a general name of that change;
- iii. 'meta X' is used as a name for something that is above X in the sense that it is more highly organized, of a higher logical type, or viewed from a larger perspective (transcending)."

Klir defines the term "meta-system" as following:

"We can see that the term 'metasystem' when used for systems in which several systems are integrated through appropriate replacement procedures, incorporates all three of these meanings. Clearly (i) a metasystem can be defined only after some other types of system are defined; (ii) it is a system that describes a change - a replacement of one system by another; and (iii) it is above the individual systems - its replacement procedure makes it more than just a collection of the individual systems. The name 'metasystem' is thus terminologically sound."

Our definition of meta-knowledge includes all three meanings of meta.

Although much research has been done to formalize systemic knowledge on the basic knowledge level, this dissertation represents the first time that formalization of systemic knowledge on the meta-knowledge level has been undertaken. Based on the above discussion, I now define *knowledge* to include basic knowledge and meta-knowledge. *Basic knowledge* is structured information. *Meta-knowledge* is the knowledge of how to sequence and process basic knowledge. In this sense, meta-knowledge imposes additional structures on top of basic knowledge as well as directly structuring information from the outside world to bring in additional knowledge beyond basic knowledge. *Complex knowledge* refers to all meta-knowledge beyond basic knowledge. These definitions lead to a formalization of an epistemological knowledge hierarchy (Fig. 2.3).

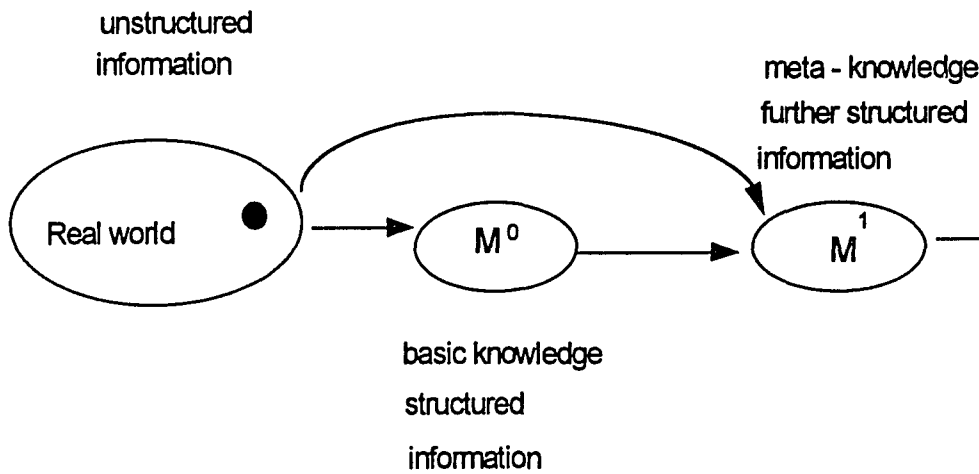


Fig. 2.3 Overview of Systemic Knowledge Hierarchy.

The major result of the research in this dissertation is to define a formalism which is general and flexible enough to capture both basic and meta-knowledge. In Chapter 3, a formal definition of the formalism is presented.

CHAPTER 3 SYSTEMIC KNOWLEDGE FORMALISM

In this chapter, a new knowledge formalism is defined. This formalism builds upon Orchard and Tausner's basic knowledge formalization [1987a, 1987b] to encompass meta-knowledge formalization. This formalism is general and flexible enough to formalize both basic knowledge and meta-knowledge. Meta-knowledge provides a way to formalize the problem solving process of a domain expert and to precisely express problem solving trajectories on the problem solving roadmap. Based on the formalism, a model and its pseudo coded representation can be developed to model the problem solving expertise of a domain expert and to simulate his/her particular problem solving process.

In the formalism, we define an epistemological systemic knowledge hierarchy, referred to as the systemic knowledge hierarchy throughout the dissertation. A brief summary of basic knowledge structure definitions and the problem solving roadmap developed by Orchard and Tausner [1987a, 1987b, 1988] is given. A canonical form is introduced which extends Orchard and Tausner's work to formalize meta-knowledge and to model the problem solving process. This is the first time systemic meta-knowledge has been formalized.

3.1 The Epistemological Systemic Knowledge Hierarchy

As indicated in Chapter 2, people have realized that multiple levels of knowledge exist. These levels are referred to meta-knowledge in general without further classification or

rigorous definition. We will introduce a systemic knowledge hierarchy which is used to classify and define the knowledge with respect to the knowledge levels.

An epistemological systemic knowledge hierarchy K is defined as a tuple (I, MS, MK, b)

where:

- I represents a non-empty set of unstructured information about the object of interest. $I^0 \subset I^1 \subset I^2 \subset \dots \subset I$, where I^i denotes the set of unstructured information needed to instantiate MS^i (described below).
- MS represents a non-empty collection of sets of meta-knowledge structures. The meta-knowledge structures will be enumerated in the order of appearance in the knowledge hierarchy.

$$MS = \{ MS^0, MS^1, MS^2, \dots \}$$

with MS^0 representing the set of basic knowledge structure, and MS^i , $i \geq 1$, representing the set of metaⁱ-knowledge structure. We assume that MS^i includes MS^{i-1} for all i , it is denoted as $MS^{i-1} \prec MS^i$, (\prec is the inclusion operator; $A \prec B$ means A is included in B).

- MK represents a non-empty collection of sets of instantiated meta-knowledge structures.

$$MK = \{ MK^0, MK^1, MK^2, \dots \}$$

with MK^0 representing the set of basic knowledge, and MK^i , $i \geq 1$, representing the set of metaⁱ-knowledge. Due to inducement, $MK^{i-1} \prec MK^i$ for all i .

- **b** represents a set of information binding functions

$$\mathbf{b} = \{ \mathbf{b}^0, \mathbf{b}^1, \mathbf{b}^2, \dots \}$$

where

$$\mathbf{b}^0 (\mathbf{I}^0, \mathbf{MS}^0) = \mathbf{MK}^0$$

$$\mathbf{b}^1 (\mathbf{I}^1, \mathbf{MS}^1) = \mathbf{MK}^1$$

$$\mathbf{b}^i (\mathbf{I}^i, \mathbf{MS}^i) = \mathbf{MK}^i$$

for all $i, i \geq 1$. \mathbf{I}^i represents all the information needed to bind $\mathbf{MS}^i, \mathbf{MS}^{i-1}, \dots$, and \mathbf{MS}^0 .

There are three properties defined based on the knowledge hierarchy:

Property **P1** (monotonicity):

$$\text{For all } i, \mathbf{MS}^{i-1} \prec \mathbf{MS}^i.$$

and due to inducement

$$\text{For all } i, \mathbf{MK}^{i-1} \prec \mathbf{MK}^i.$$

Property **P2** (transitivity):

$$\text{for all } i, j, k, \text{ and } i < j < k,$$

$$\text{if } \mathbf{MS}^i \prec \mathbf{MS}^j ; \mathbf{MS}^j \prec \mathbf{MS}^k \text{ then } \mathbf{MS}^i \prec \mathbf{MS}^k .$$

and due to inducement

$$\text{for all } i, j, k, \text{ and } i < j < k,$$

$$\text{if } \mathbf{MK}^i \prec \mathbf{MK}^j ; \mathbf{MK}^j \prec \mathbf{MK}^k \text{ then } \mathbf{MK}^i \prec \mathbf{MK}^k .$$

Property **P3** (inheritance/encapsulation):

$$\text{For all } i, \mathbf{MS}^{i-1} \in \mathbf{MS}^i \text{ and, due to inducement, } \mathbf{MK}^{i-1} \in \mathbf{MK}^i.$$

$$\text{Where } \mathbf{MS}^{i-1}, \mathbf{MS}^i \in \mathbf{MS} \text{ and } \mathbf{MK}^{i-1}, \mathbf{MK}^i \in \mathbf{MK} \text{ (} \in \text{ is defined as}$$

inheritance/encapsulation operator, $A \in B$ means that A is inherited/encapsulated by B).

Property **P1** and **P2** are always true for all the knowledge hierarchy. Property **P3** is true if the inclusion relationship is an inheritance/encapsulation relationship, a special case of inclusion.

If **P3** is true then the set of information binding functions $\mathbf{b} = \{ \mathbf{b}^0, \mathbf{b}^1, \mathbf{b}^2, \dots \}$ can be refined as:

$$\mathbf{MK}^0 = \mathbf{b}^0 (\mathbf{I}^0, \mathbf{MS}^0)$$

$$\mathbf{MK}^1 = \mathbf{b}^1 (\mathbf{I}^1, \mathbf{MS}^1 \setminus \mathbf{MS}^0) \uplus \mathbf{MK}^0$$

$$\mathbf{MK}^i = \mathbf{b}^i (\mathbf{I}^i, \mathbf{MS}^i \setminus \mathbf{MS}^{i-1}) \uplus \mathbf{MK}^{i-1}$$

where \mathbf{b}^i binds \mathbf{I}^i to $\mathbf{MS}^i \setminus \mathbf{MS}^{i-1}$ (the part of \mathbf{MS}^i that is not includes \mathbf{MS}^{i-1}). \uplus is defined as knowledge union operator. If \mathbf{MK}^i encapsulates \mathbf{MK}^{i-1} then \mathbf{MK}^{i-1} is invisible to meta^{i+1} - knowledge level or above, but \mathbf{MK}^{i-1} can be accessed or manipulated through \mathbf{MK}^i .

Here we offer a high level definition. It expresses three basic concepts mathematically. First, there exist multi-levels of knowledge. Second, knowledge consists of information content and knowledge structures, knowledge is instanciated knowledge structures. Third, the knowledge hierarchy is ordered by amplifying the knowledge already obtained through binding additional information to knowledge structures at a higher level.

Based on the definition of knowledge hierarchy, different knowledge hierarchies can be developed due to the variations of inclusion relationship between two meta-levels on the

knowledge hierarchy. In this dissertation, we developed the formalism based on a inheritance/encapsulation relationship, a special case of inclusion.

Knowledge includes basic knowledge and complex knowledge. Complex knowledge refers to all meta-knowledge beyond basic knowledge. Basic knowledge, MK^0 , is structured information using basic knowledge structures (MS^0). Meta-knowledge, MK^i is structured information using meta-knowledge structures, MS^i . In addition meta-knowledge, MK^i , imposes further structure on top of the meta-knowledge at the lower level, MK^{i-1} . When information is structured using a particular knowledge structure, we also say that the knowledge structure is instantiated. So it is equivalent to define knowledge as instantiated knowledge structures or knowledge as structured information.

In the systemic knowledge hierarchy definition, b is an information binding function or instantiating function. b is defined through the systemic knowledge elicitation processes, An elicitation process which is particularly well suited to filling in these knowledge structures. The systemic knowledge elicitation process is presented in a variety of papers [Orchard, Reese and Tausner, 1987a, 1987b, 1988, 1989]. During the knowledge elicitation processes, the information binding function will fill the knowledge structures with the particular domain information content. b^0 binds information to the basic knowledge structures to form basic knowledge. b^i binds extra information to the metaⁱ-knowledge structures to form metaⁱ-knowledge.

In the following sections we will define the other components of the knowledge hierarchy.

3.2. The Problem Solving Roadmap and Basic-Knowledge Structures

In the 1970's, Robert A. Orchard, while at AT&T Bell laboratories, introduced a morphological grid for modeling complex problem solving strategies. The grid, in the form of a directed graph, was called the problem solving roadmap (see Fig. 3.1). Based on the problem solving roadmap, Orchard and Tausner developed the definition for basic knowledge structures [Orchard, Tausner, 1987a, 1987b]. They referred to the definitions for the nodes on the roadmap as basic knowledge structures (MS^0), namely, *source system*, *data system*, *behavior system* and *structure system*, and defined "basic knowledge

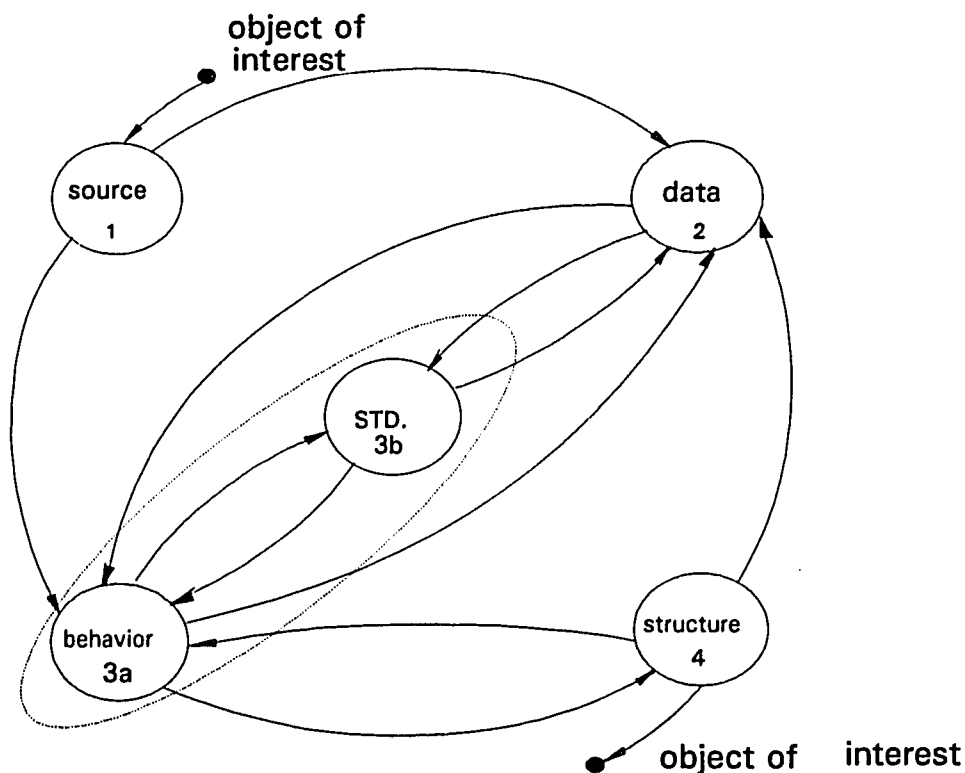


Fig. 3.1 Problem Solving Roadmap and Basic Knowledge Structures.

as structured information" [Orchard, 1990]. The arc between two nodes on the problem solving roadmap represent the methodologies which are used to process the knowledge

structures (represented by the first node) and output other knowledge structures (represented by second node).

A precise definition of basic knowledge structure is presented by Orchard and Tausner [Orchard and Tausner, 1988] as below:

"Definition 1: Source System

A source system is a collection of defined variables and resolution levels. A resolution level is associated with a variable and represents the set of values a variable can take on. The variables may further be distinguished as dependent or independent (support). That is,

$$X_1 = \text{"definition of } X_1\text{"} \quad L_1 = \{v_{11}, \dots, v_{1j}, \dots, v_{1n_1}\}$$

$$X_k = \text{"definition of } X_k\text{"} \quad L_k = \{v_{k1}, \dots, v_{ki}, \dots, v_{kn_k}\}$$

$$X_N = \text{"definition of } X_N\text{"} \quad L_N = \{v_{N1}, \dots, v_{Nj}, \dots, v_{Nn_N}\}$$

where X_k is the k th variable being defined and v_{ki} is the i -th value among the n_k values of the variable in the resolution-level set, L_k . The values of the variables occurring in the resolution-level set may be logical, arithmetic, characters, pictures, etc. The resolution-level set may also contain intervals.

Definition 2: Data System

The definition of a system by its activity through the manifestation of data consists of

- (1) a definition of the source system and
- (2) a collection of instances of dependent variable values versus a time variable and/or other independent (support) variables.

A data system is a system at a higher epistemological level than a source system. A definition of a data system must have as one of its parts a definition of the source system. If one is simply given data, then by default, the resolution levels of the source system are determined by the data. However, it is preferred that the source system definition be given independent of actual data, since one possibly then has additional information, namely, values of the variable which could occur even if they do not occur in the particular data system under investigation.

Definition 3: Behavior System

The definition of a system by its behavior consists of

- (1) a definition of the source system and
- (2) a relationship R among the variables X_k , $k = 1, \dots, N$, of the source system

$$R(X_1, \dots, X_N)$$

The relationship may be described in many ways using mathematical notation, linguistic statements, or even the presentation of experiential examples of the relationship. These issues have to do with the presentation of the relationship. The same relationship may be presented in many ways.

There is an important special case of behavior which appears cross disciplinary at a frequency which warrants a special definition.

Definition 3a: State Transition Description

The definition of a system by its state transition description consists of

- (1) a definition of a source system and
- (2) the state space, S , defined as

$$S = \{s | s \text{ is a state of the system}\}$$

where the state of a system is a vector $s = \langle v_1, \dots, v_i, \dots, v_N \rangle$, where v_i is contained in the resolution-level set of the source system variable X_i ; and

- (3) a relationship R , defined on $S \times S$, i.e., $R(S \times S)$. In addition, a measure may be placed on the relationship R associating each element of $S \times S$ with a value.

Definition 4: Structure System

The definition of a system by its structure consists of the following.

- (1) a definition of the source system for the environment,
- (2) a collection of P of M elements,

$$P = \{p_1, \dots, p_M\}$$

called the primitive elements of the system. For each element p_i there is a

definition of the element by definition 1 (source), definition 2 (data), definition 3 (behavior), or, recursively, definition 4 (structure).

(3) a collection C of K (greater than or equal to M) elements,

$$C = \{c_0, c_1, \dots, c_K\}$$

called the components of the system. A component, c_i , $i > 0$, is an instance of a primitive element. c_0 names the environment.

(4) a collection C' , of couplings or connections $C_{i,j}$, $C' = \{C_{i,j} | C_{i,j} \text{ is the set of variables that component } i \text{ and component } j \text{ share in common}\}$. If the notions of input to and output from are to be distinguished, then the notation $C_{i,j}$ indicates the set of variables which are output from component i and input to component j . In this case, $C_{i,j}$, is not necessarily the same as $C_{j,i}$. The indices i and j are contained in $\{0, 1, \dots, K\}$ and C_{0j} and C_{i0} represent the couplings of component i with the environment. Each $C_{i,j}$ is defined by a source system."

3.3. Canonical Form for Meta-Knowledge Structures

The problem solving roadmap is a rich conceptual model which can be used to model a domain expert in terms of problem solving. The problem solving roadmap includes more than basic knowledge structures. It also contains trajectories to track the problem solving strategies by capturing the sequences of movements between the nodes.

For a knowledge engineer, to understand the problem solving strategies of a domain expert, he/she can trace a path on the problem solving roadmap to see what knowledge structures the expert is currently investigating and what knowledge structures the expert is going to investigate next. In addition, the knowledge engineer has to understand under what conditions (when and why) the expert makes such movements on a roadmap and what methodologies the expert will use to make such movements. The methodologies which the expert selects will process the current knowledge structures as inputs and form the target knowledge structures as outputs. We refer to these strategies as **meta-knowledge**, which are beyond basic knowledge. Meta-knowledge controls how basic knowledge is processed. It is the meta-knowledge which forms the problem solving strategies. Therefore, it is essential to define meta-knowledge structures so that meta-knowledge can be formalized.

A **canonical form** is introduced here for meta-knowledge structures. The canonical form is general and flexible enough that, when supplemented by communication and control constructs, it can be used to formalize meta-knowledge at all levels. We refer to a knowledge structure on level i as a meta^i -knowledge structure. Basic knowledge structures can be viewed as meta^0 -knowledge structures. Meta^i -knowledge structures control how the lower level meta^{i-1} -knowledge structures are processed.

A canonical form Φ for a meta-knowledge structure is defined as follows:

$$\Phi = (\Sigma, \Lambda, \Delta, \Pi, \Gamma, H).$$

The components in the canonical form can be divided into two groups. The components, Σ , Λ and Δ , define and deal with knowledge structures. The components, Π , Γ and H define and deal with methodologies.

$\Sigma = (\Sigma_0, \Sigma_1, \Sigma_2, \dots, \Sigma_m)$, is a collection of sets of knowledge structures, where $m \geq 1$, the set of nonnegative integers.

$\Sigma_i = (\sigma_1, \sigma_2, \dots, \sigma_u)$ where $u \geq 1$. Σ_i ($i=0, 1, \dots, m$) consists of knowledge structures of the same type, where the possible knowledge structure types are defined as: source systems; data systems ; behavior systems; structure systems.

In particular, we let

$\Sigma_0 = \alpha$, is a set of initial knowledge structures (e.g. Source Systems).
 $\Sigma_m = \omega$, is a set of final knowledge structures (e.g. Structure Systems),
 These depend on the goal of problem solving.

$\Lambda = (\Lambda_0, \Lambda_1, \dots, \Lambda_m)$, is a collection of sets of satisfaction criteria variables.

$\Lambda_i = (\lambda_1, \lambda_2, \dots, \lambda_r)$ is a set of satisfaction criteria variables defined for Σ_i . Each satisfaction criteria variable has its own resolution level - $l\lambda_j$, where $j = 1, 2, \dots, r$. A r -Cartesian product of their resolution levels defines a r dimensional satisfactory state space - T_i , a set of r -tuples.

$$T_i = l\lambda_1 \times l\lambda_2 \times \dots \times l\lambda_r, \text{ where } r \in \mathbb{N}.$$

$\Delta = (\delta_0, \delta_1, \dots, \delta_m)$, is a set of satisfaction measurement functions.

$\delta_i(\Sigma_i, T_i) = \tau_i$, δ_i measures Σ_i against T_i and produces a value τ_i .

Where $\tau_i \in \{0, 1\}$, a Boolean set or $\tau_i \in [0, 1]$, an interval of real numbers from 0 to 1. This value τ_i can be used to control the problem solving process together with control constructs such as **If ... Then** and **While ... End** which will be defined later.

$\Pi = (\Pi_0, \Pi_1, \dots, \Pi_m)$, is a collection of collections of methodology sets.

$\Pi_i = (M_{ij} \mid j = 0, 1, 2, \dots, m)$, is a collection of sets of methodologies which are applicable to the set of knowledge structures Σ_i .

$M_{ij} = (\mu_{ij,1}, \mu_{ij,2}, \dots, \mu_{ij,v})$, is one set of methodologies within the collection Π_i . Each M_{ij} consists of methodologies of the same type which process Σ_i and outputs Σ_j , where $v \in \mathbb{N}$.

In general, there are three types of methodologies that can be applied to a set of knowledge structures Σ_i . They are:

Transformation, Refinement and Selection. When $i \neq j$, M_{ij} is a set of **Transformation** Methodologies. When $i = j$, M_{ij} is either a set of **Refinement** methodologies or a set of **Selection** methodologies. Each $\mu_{i,j,k}$ ($k=1, 2, \dots, v$) is a particular methodology within $M_{i,j}$, all members of $M_{i,j}$ being of the same type, where the possible types are defined as follows:

Transformation: $\mu_{i,j,k} : \Sigma_i \rightarrow \Sigma_j$.

Refinement: $\mu_{i,i,k} : \Sigma_i \rightarrow \Sigma_i$.

Selection: $\mu_{i,i,k} : \Sigma_i \rightarrow P(\Sigma_i)$, where $P(\Sigma_i)$ is a power set of Σ_i .

Transformation is a type of methodology which transforms a set of knowledge structures of one type to another set of knowledge structures of another type. Either all or none of the knowledge structures in the set are being transformed. Otherwise a selection methodology should be applied before the transformation.

Refinement is a type of methodology which refines a set of knowledge structures of the same knowledge type to produce a set of knowledge structures of this type.

At least one knowledge structure in the set is refined. The total number of knowledge structures within the set does not change.

Selection is a type of methodology which selects a subset of knowledge structures from a set of knowledge structures. Selection is useful for reducing the number of knowledge structures in a set, therefore, reducing the complexity of the model.

$\Gamma = (\Gamma_0, \Gamma_1, \dots, \Gamma_m)$, is a collection of sets of methodology selection variables.

$\Gamma_i = (\gamma_1, \gamma_2, \dots, \gamma_w)$, is a set of methodology selection variables which will be used to select one particular methodology, $\mu_{i,j,k}$, out of a set of methodologies, $M_{i,j}$, where $w \in \mathbb{N}$, i and $j \in \{0, 1, \dots, m\}$, $k \in \{1, \dots, v\}$.

$H = (\eta_0, \eta_1, \dots, \eta_m)$, is a set of methodology selection functions.

$\eta_i(M_{i,j}, \Gamma_i) = \mu_{i,j,k} \in M_{i,j}$, η_i selects a particular methodology, $\mu_{i,j,k}$, from the methodology set, $M_{i,j}$, based on the set of methodology selection variables, Γ_i , where i and $j \in \{0, 1, \dots, m\}$, $k \in \{1, \dots, v\}$.

3.4 Problem Solving Control Constructs

On the problem solving roadmap, there are trajectories to track the problem solving strategies by capturing the sequences of movements between the nodes. There is a decision making process at each node of the trajectory to determine which methodology to use for the next problem solving step. To model the problem solving strategy, the following control constructs are defined as part of the formalism.

a. **IF** < condition expression > **THEN**

< problem solving process >

END

b. **WHILE** < condition expression >

< problem solving process >

END

with

< condition expression > :=

$\delta_i(\Sigma_i, T_i) = \tau_i$ |

$\delta_i(\Sigma_i, T_i) \neq \tau_i$ |

$\delta_i(\Sigma_i, T_i) < \tau_i$ |

$\delta_i(\Sigma_i, T_i) > \tau_i$ |

$\delta_i(\Sigma_i, T_i) \leq \tau_i$ |

$\delta_i(\Sigma_i, T_i) \geq \tau_i$ |

< NOT condition expression > |

< condition expression AND condition expression > |

< condition expression OR condition expression >

< problem solving process > :=
 < problem solving process > < problem solving process > |
 ε (a null problem solving process with no action) |
 $\mu_{i,j,k} : \Sigma_i \rightarrow \Sigma_j$ (transformation) |
 $\mu_{i,i,k} : \Sigma_i \rightarrow \Sigma_i$ (refinement) |
 $\mu_{i,i,k} : \Sigma_i \rightarrow P(\Sigma_i)$ (selection)

where i and $j \in \{ 0, 1, \dots, m \}$, m is the number of knowledge structure sets,
 $k \in \{ 1, 2, \dots, v \}$, where v is the number of methodologies in the set $M_{i,j}$.

These control constructs can be used together with a satisfaction measurement functions to model a problem solving strategy and to control a problem solving process. For example:

IF $\delta_i(\Sigma_i, T_i) = \tau_i$ **THEN**
 < problem solving process >
END

or

WHILE $\delta_i(\Sigma_i, T_i) = \tau_i$
 < problem solving process >
END

If the value τ_i indicates that Σ_i is a set of satisfactory knowledge structures with respect to δ_i , and Λ_i then the model can proceed to the next step. If the value τ_i indicates that Σ_i is a set of unsatisfactory knowledge structures, then the model may need to refine Σ_i

until Σ_i is a set of satisfactory knowledge structures. In the case of T_i is a fuzzy set, fuzzy logic may then be applied to control the problem solving process.

3.5 Communication Constructs

The canonical form provides a way to model an individual expert and formalize his/her expertise. It is often the case that a team of experts needs to work together to solve a complex problem, each one acting as a specialized intelligent problem solving agent. A problem solving roadmap is used for each problem solving agent to model its problem solving process. These roadmaps form a complex plan and communicate each other through a set of communication and control constructs.

As part of the formalism, the following operations are defined for communication purposes.

Send(SenderId, ReceiverId, KnowledgeType, Knowledge)

Receive(SenderId, ReceiverId, KnowledgeType, Knowledge)

Send and **Receive** is used by a roadmap for data communication with other roadmaps. The **SenderId** and **ReceiverId** indicate the sender and the receiver respectively. The **KnowledgeType** indicates what type of knowledge structure is involved in the communication. The knowledge, a instantiated knowledge structure, is the contents of the communication. A roadmap uses the **Receive** operation to receive data.

Ask(SenderId, ReceiverId, KnowledgeType, Knowledge)

Ask allows one roadmap to send a request to another roadmap (indicated by RoadmapId) asking for certain data. The last parameter, Knowledge, is partially instantiated knowledge structures passed to the receiver for filling in the requested data.

Activate(RoadmapId)

Suspend(RoadmapId, KnowledgeType)

Activate and **Suspend** are used by a roadmap, with certain privileges, to send messages to different roadmaps and to control their problem solving processes. **Activate** will put a roadmap in **Active** mode in which the roadmap will resume the normal problem solving process. **Suspend** will put a roadmap in **Suspend** mode in which the roadmap will stop the normal problem solving process upon reaching the knowledge structures specified by KnowledgeType and wait for further instruction. Each RoadmapId is a unique identification for a roadmap.

The complete formalism we introduced in this chapter includes the definitions of systemic knowledge hierarchy, the definitions of basic knowledge structures, the definitions of meta-knowledge structures (canonical form), the definitions of control constructs and the definitions of communication and constructs .

A case study of cooperative problem solving is presented in the next chapter. In the case study, the formalism is used to model basic knowledge and meta-knowledge.

CHAPTER 4 CASE STUDY - COOPERATIVE PROBLEM SOLVING¹

In this chapter, we present an example of cooperative problem solving in which a team of automotive engineers work together to solve a car design problem. The case study shows how our systemic knowledge formalism can be used to model this real world problem. The case study will demonstrate how to formalize the expertise of a group of domain experts into basic knowledge and meta-knowledge within the framework of our formalism.

Modeling a cooperative problem solving process demonstrates one way to apply the systemic knowledge formalism. But it represents a wide range of problems in the real world. As science and technology keep advancing through this and the next century, the domain expertise of an expert will become deeper and more specialized within a particular domain. On the other hand, many of the most urgent problems we try to solve lie on the boundaries of different domains. Solving such problems requires a cooperative effort of different domain experts. Meta-knowledge is a key factor in building knowledge-based systems for cooperative problem solving.

In his discussion of the role of meta-knowledge in different levels, Robert A. Orchard [1990] states:

"The expertise no longer lies within the domain of a single expert but rather in a group of 'disjoint' experts ..., a first level supervisor of experts must have such a

¹ The model developed in this dissertation has been successfully utilized and validated in several industrial projects. Because these applications are proprietary and complex, a simplified case study is presented instead.

collection of methodologies or sequencing strategies for managing the individual expert behavior in order to manifest the group expert behavior."

In the case study, we will first present the problem followed by problem analysis, then use the systemic knowledge formalism to develop a model of the car design problem. Finally, we will show how to use this model for different problem solving strategies with respect to the car design problem.

4.1 Cooperative Problem Solving - A Car Design Problem

We take as our case study a team of four experts working together to design a passenger car. The case study is simplified so it can be used as an example in the following discussion but the characteristics of cooperative problem solving are retained. A car is complex system. As in designing any complex system, the task is decomposed into several subtasks: design of the power system (engine system, transmission system, exhaust system), design of the control system (steering system, brake system), and design of the support system (suspension system, passenger and engine compartments). Each subtask is manageable and can be best performed by an individual specialist.

One expert acts as a team manager. The team manager is responsible for the system integration, team problem solving strategy and coordination. Each of the other team members is responsible for one particular subsystem design. A successful design depends not only on the extensive in-depth expertise that each individual expert possesses but also on their expertise's for problem solving strategy and team management. In our formalism these extensive in-depth expertise of each individual expert are identified as basic knowledge. The problem solving strategy of each expert and team management expertise of the manager are identified as meta-knowledge.

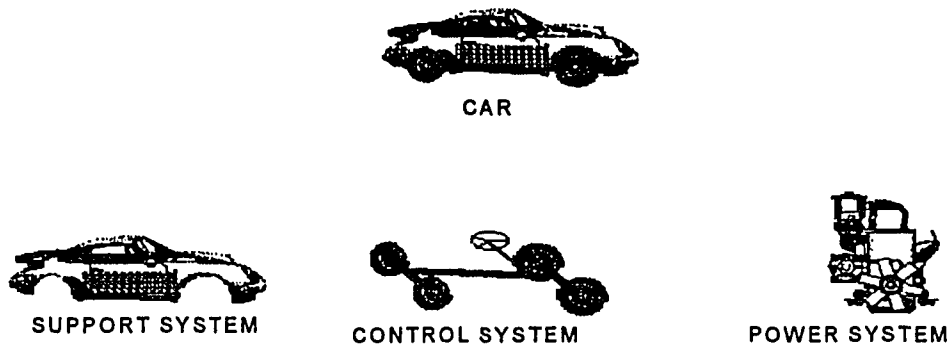


Fig 4.1 Cooperative Problem Solving - A Car Design Problem

4.2 Modeling the Expertise of Each Expert

Each expert acts as an intelligent problem solving agent and works independently until there is a need for communication with other team members. The experts study the subsystems which they are to design and apply their expertise in the design process to produce a design which best fits the specifications. They then report their results to the manager. Should there be any changes in the design specification, the manager will pass those changes to the corresponding expert, so that they can modify their design. Finally the manager integrates all the subsystem designs together to complete the car design task.

The problem solving activity of each expert, including the manager, is modeled using a separate roadmap. These roadmaps form a complex plan (Fig. 4.2). To keep the figure simple, Fig. 4.2 only shows the nodes (source system, data system, behavior system and structure system) without the arcs between the nodes.

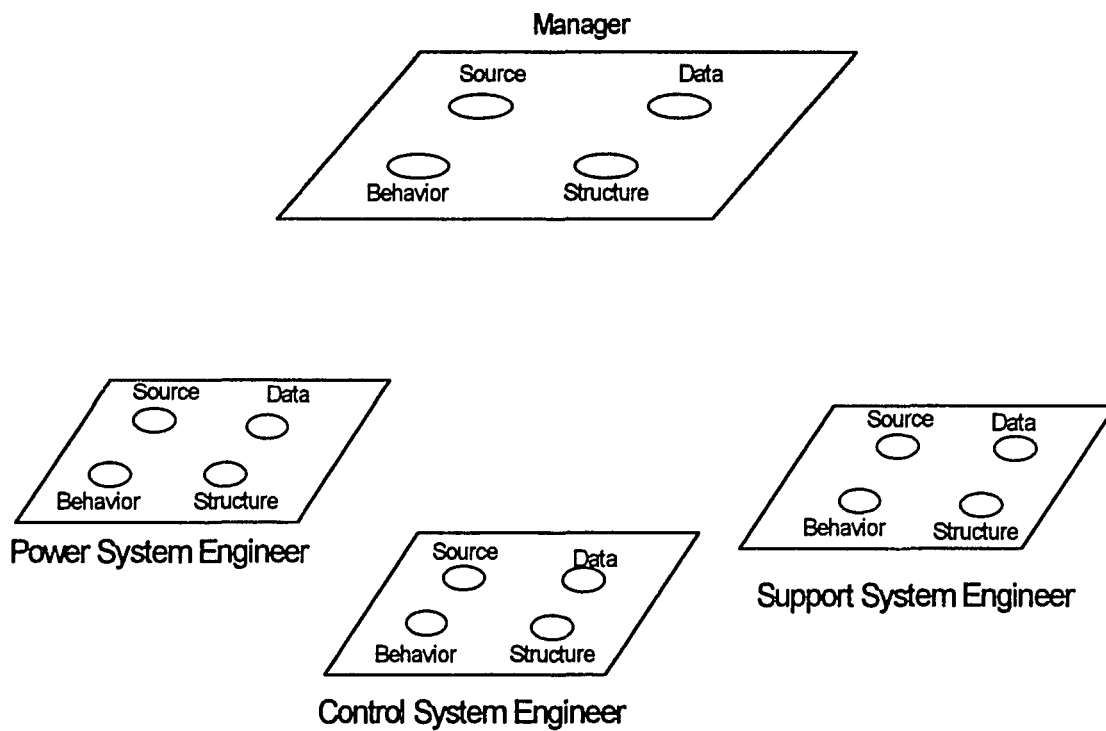


Fig 4.2 Roadmap for Group Experts - A Complex Plan.

There is a difference between a model of the expert and the real expert. In the rest of this dissertation, when we refer to an expert we mean the model of the expert.

4.3 Modeling the Cooperation Between the Experts

In our model of cooperative problem solving, beside the roadmaps there are links between nodes of the different roadmaps for communication. While experts work independently on their own problem solving, the manager can control and coordinate the whole process with different overall problem solving strategies.

A manager has different ways of controlling the problem solving process. For example, tight control may be defined as the case in which each expert can only send or receive data and messages to/from the manager. Individual experts cannot communicate with each other directly. Loose control may be defined as the case in which each expert can send and receive data and messages directly to/from each other. Between these two extreme cases, the model can provide a different range of control to model the manager's problem solving strategy by making certain roadmap identifications available only to particular experts.

As part of our model, the following operations or procedures are used for communication propose.

Send and Receive are used by all experts including the manager for communicating data with other experts. The **SenderId** and **ReceiverId** indicate the sender and the receiver respectively. The **KnowledgeType** indicates what type of knowledge structure is involved in the communication. The **Knowledge**, a instanciated knowledge structure, is the contents of the communication. After receiving data from another expert, an expert uses the **Receive** procedure to acknowledge the sender the data has been received.

Ask allows the manager and experts to send a request to an expert (indicated by ReceiverId) asking for certain data.

Activate and **Suspend** are used by manager to send messages to different experts and to control their problem solving process. **Activate** will put an expert in **Active** mode in which the expert will resume the normal problem solving process. **Suspend** will put an expert in **Suspend** mode in which the expert will stop the normal problem solving process upon reaching the knowledge structures specified by KnowledgeType and wait for further instruction. Each RoadmapId is a unique identification for a roadmap.

4.4 Developing The Model

For cooperative problem solving, the formalism can model each individual expert's problem solving expertise as well as the manager's expertise for problem solving strategy and coordination throughout the whole process. The dynamics of the problem solving process and the interactions between the team members are represented in our model by snapshots of where the experts are on their individual roadmaps. The knowledge (basic and meta-knowledge) involved in the problem solving process is rigorously formalized in our formalism.

The model includes an instance of a roadmap and a canonical form for the meta-knowledge structures for each expert in the team. The model is built in steps, starting with source systems, then data systems, behavior systems and structure systems.

In the model, we assume that person A is the team manager, person B is the power system engineer, person C is the control system engineer and person D is the support system engineer.

The four instances of canonical form for each engineer are denoted as:

$$\Phi = (\Sigma, \Lambda, \Delta, \Pi, \Gamma, H)$$

$$\Phi_A = (\Sigma_A, \Lambda_A, \Delta_A, \Pi_A, \Gamma_A, H_A)$$

$$\Phi_B = (\Sigma_B, \Lambda_B, \Delta_B, \Pi_B, \Gamma_B, H_B)$$

$$\Phi_C = (\Sigma_C, \Lambda_C, \Delta_C, \Pi_C, \Gamma_C, H_C)$$

$$\Phi_D = (\Sigma_D, \Lambda_D, \Delta_D, \Pi_D, \Gamma_D, H_D)$$

In the model, the initial knowledge structures in all the canonical forms are source systems. The instances of these source systems are follows:

α_A - Source System for Person A (manager):

X_{A1} : Total Cost of the car;

X_{A2} : Maximum Speed of the car;

X_{A3} : Air Pollution Standard;

X_{A4} : Number of passengers;

X_{A5} : Car size.

α B - Source System for Person B (power system engineer):

XB1 : Unit Cost;

XB2 : Fuel Efficiency;

XB3 : Air Pollution;

XB4 : Space Requirement (engine compartment);

XB5 : Max. Speed of the engine (RPM);

XB6 : Horse Power of the engine;

α C - Source System for Person C (control system engineer):

XC1 : Unit Cost;

XC2 : Braking Distance;

XC3 : Number of gear positions;

XC4 : Steering(radius of 90 degree turn);

α D - Source System for Person D(support system engineer):

XD1 : Unit Cost;

XD2 : Aerodynamic factor;

XD3 : Space(passenger compartment, engine compartment, trunk);

XD4 : Safety factor;

After forming the source systems, certain measurement is taken to measure the newly formed knowledge structure against certain satisfaction criteria with respect to the problem. A set of satisfaction criteria variables and a satisfaction measurement function, which are applicable to the source systems, are elicited from domain experts and formalized as Λ_i and Δ_i ($i=0$, for source systems) accordingly. Based on the measurement, one of the three possible types of methodologies is applied to the source system. It can be refinement, $M_{i,i}$, or selection, $M_{i,j}$, or transformations, $M_{i,j}$ ($M_{i,j} \in \Pi_i$, $i = 0$ and $j = 0, 1, 2, \dots, m$), which represent the arcs coming out from the source system node on a road map. A methodology selection function is used to pick up a particular methodology from the set to process the source system according to certain methodology selection criteria. These methodologies, methodology selection variables and methodology selection functions are elicited from domain experts and formalized as Π_i , Γ_i , and H_i ($i=0$, for source systems) in the canonical form accordingly.

In general, the above measurement and selection processes take place after forming each knowledge structure. All the variables and functions needed to perform these processes are elicited from the domain experts and formalized in our canonical form.

In the example, the data systems are formed and refined in steps. First partial data is collected for certain variables of the source system on each roadmap. This initial data comes from the design specification and is used to set the initial conditions for the car design problem. The rest of data is generated by solving the relationships specified by behavior systems on each roadmap. For example, some of the initial data in the manager's data system might be the total cost of the car and the maximum number of passengers (such as, total cost must be below \$6000 and it is a 5 passenger car).

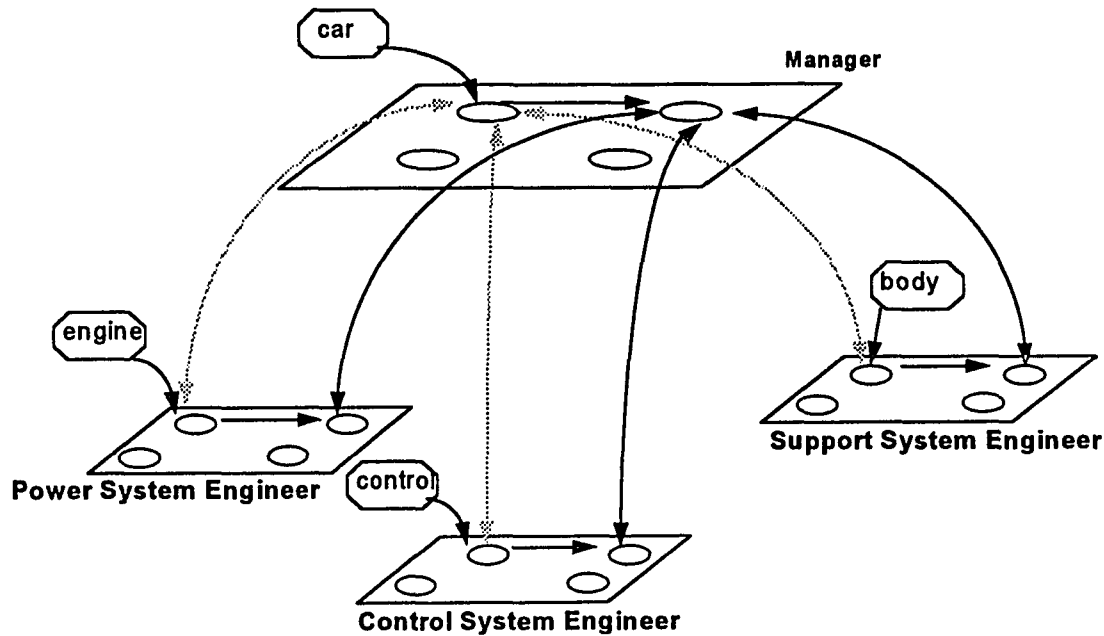


Fig. 4.4 The Data Systems of the Model.

A feasible solution for the car design problem must satisfy these initial conditions. Some of the initial data in manager's data system will be projected down and become the initial data in the individual expert's data system through the communication links between their data systems. For example, the total cost of the car \$6000 may initially be divided equally into the unit cost of \$2000 for each subsystem and then the unit cost may be projected down to each individual expert's data system. On the other hand, each individual expert may need to verify his/her design with the manager by projecting up some of the data in his/her data system to the manager's data system (see Fig. 4.4).

The behavior system is formed on each roadmap by describing the relationship among the variables defined in the source system. These relationships are elicited from the individual experts and are formalized in a behavior system for each expert. Such relationships can be

represented in many ways, and in our example some are represented as mathematical relationships.

The following appear in the behavior system of the manager:

$X_{A1} \geq X_{B1} + X_{C1} + X_{D1}$ (total cost and unit cost relationship)

$X_{A2} \leq f(X_{B5}, X_{B6})$ (car speed, engine speed, and engine horse power relationship. Function f is defined on engine speed and engine horse power.)

$X_{A3} \geq g(X_{B2}, X_{B3})$ (air pollution standard, fuel efficiency and engine emission relationship. Function g is defined on fuel efficiency and engine emission.)

$X_{A5} \geq h(X_{A4}, X_{D3})$ (car size, number of passenger and space relationships, Function h is defined on the number of passengers and the space required for the separate compartments.)

The following appear in the behavior system of the power system engineer:

$X_{B1} \leq c$ (unit cost for power system. c is a constant.)

$X_{B6} \leq o(X_{B5}, X_{B2})$ (horse power, engine speed and fuel efficiency relationship. Function o is defined on engine speed and fuel efficiency.)

$X_{B4} < b X_{A5}$ (engine space and car size relationship. b is a constant.)

The following appear in the behavior system of the control system engineer:

$X_{C1} \leq d$ (unit cost of control system. d is a constant)

$X_{C2} \leq p(X_{A2})$ (braking distance and maximum car speed relationship.
Function p is defined on maximum car speed.)

$X_{C4} \leq q(X_{A5}, X_{A2})$ (radius of 90 degree turn, car size and maximum car speed
relationship. Function q is defined on car size and maximum
car speed.)

The following appear in the behavior system of the support system engineer:

$X_{D1} \leq e$ (unit cost of support system. e is a constant.)

$X_{D2} \geq t(X_{A2}, X_{B2})$ (aerodynamic factor, maximum car speed and fuel efficiency
relationship. Function t is defined on maximum car speed and
fuel efficiency.)

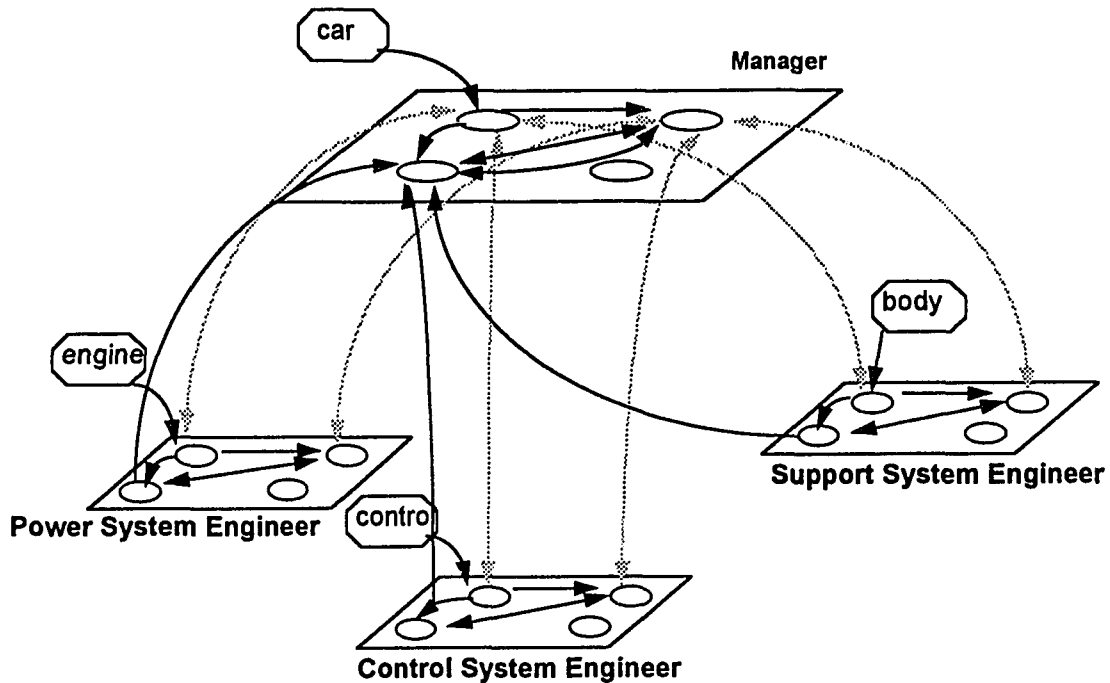


Fig. 4.5 The Behavior Systems of the Model.

Let us refer to one roadmap as a model of one expert. Each model is used to determine a set of values which satisfy the required relationships specified in the behavior system. Initial conditions are set using the initial data of the data system. When a solution is found, all the variables involved are instantiated by assigning values to them. These values complement the initial data and form the data system on each roadmap.

Data is communicated between the manager's roadmap and each expert's roadmap through the variables they share in their source system. When new data is generated on each expert's roadmap which is to be reported to the manager's model, the manager's model has the expertise to compare the reported data with other data in the manager's data system to check if the new data satisfies the overall design specification using a measurement function, $\delta_i(\Sigma_i, T_i)$. If the new data fails to meet design specifications, or a

feasible solution cannot be found within given initial conditions, the manager's model has the expertise to determine a new set of initial conditions and request that an expert continue problem solving using the new set of initial conditions.

On the other hand, if the design specification changes, the manager's model projects down a new set of data as initial conditions and asks the expert's model to resolve the relationships in the expert's behavior system using new data. This iteration can continue until the feasible solution is found. If there exists more than one feasible solution, the manager's model has the expertise to select the best one with respect to the overall design specification.

For each expert and the manager, such problem solving activities can be modeled by a roadmap (see Fig. 4.5). The iteration is indicated by a loop between data system and behavior system.

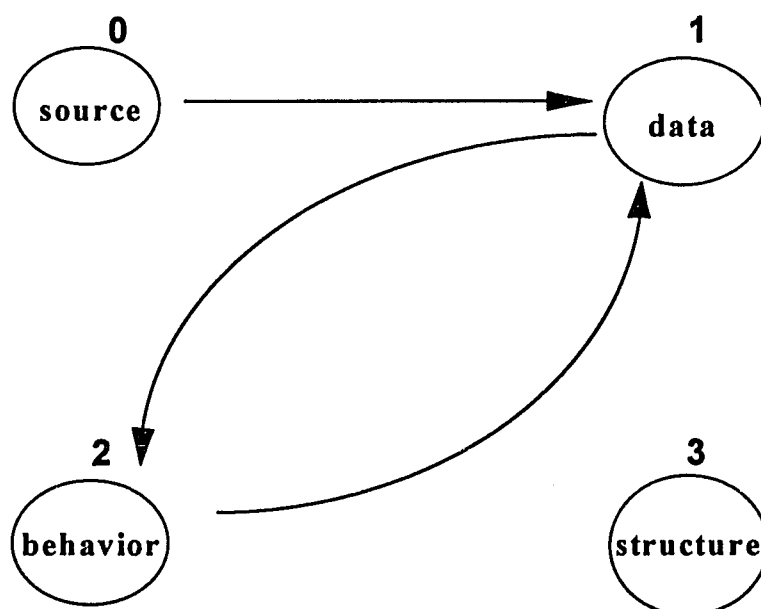


Fig. 4.6 Iteration between data and behavior system.

The problem solving process of each individual expert in the example can be expressed in the formalism as pseudo code:

```

 $\eta_0(M_{0,1}, \Gamma_0) = \mu_{0,1,k}$  ;select a transformation methodology.
 $\mu_{0,1,k}: \Sigma_0 \rightarrow \Sigma_1$  ;transformation from source to data.
WHILE  $\delta_1(\Sigma_1, T_i) = \text{FALSE}$  DO ;measure the data against the
;design specification.
 $\eta_1(M_{1,2}, \Gamma_1) = \mu_{1,2,k}$  ;select a transformation
;methodology.
 $\mu_{1,2,k}: \Sigma_1 \rightarrow \Sigma_2$  ;update the constrains on the
: ;relationships on behavior system.
 $\eta_2(M_{2,1}, \Gamma_2) = \mu_{2,1,k}$  ;select a transformation
;methodology.
 $\mu_{2,1,k}: \Sigma_2 \rightarrow \Sigma_1$  ;generate new data from the
;behavior system.

```

END

The structure systems for each expert are formed after solving the relationships in the behavior systems and the results have been reported to the manager and the results are approved by the manager. The structure system is a implementation of the behavior system. After the structure system is formed, the model performs a simulation based on the structure system. The model will compare the data generated by the simulation to the data generated earlier from the behavior system to evaluate whether or not the structure system is a faithful implementation of the behavior. If not, the model will modify (refine) the structure system based on the feedback of the comparison. This iteration is

continue until the structure system is a satisfactory implementation of the behavior system. The iteration is indicated on the roadmap below (Fig. 4.6).

Some of the constraints imposed on individual expert's behavior system, such as unit cost, will be used as guide lines to design or optimize the structure system. For example, if there are many alternative choices available on the market for certain sub-system component, the expert has to choose the one which fits the given budget. If the expert cannot find a machine part for certain sub-system component on the market then the expert has to report this incident to the manager. The manager might decide to increase the unit cost for that expert and find a way to balance the overall budget.

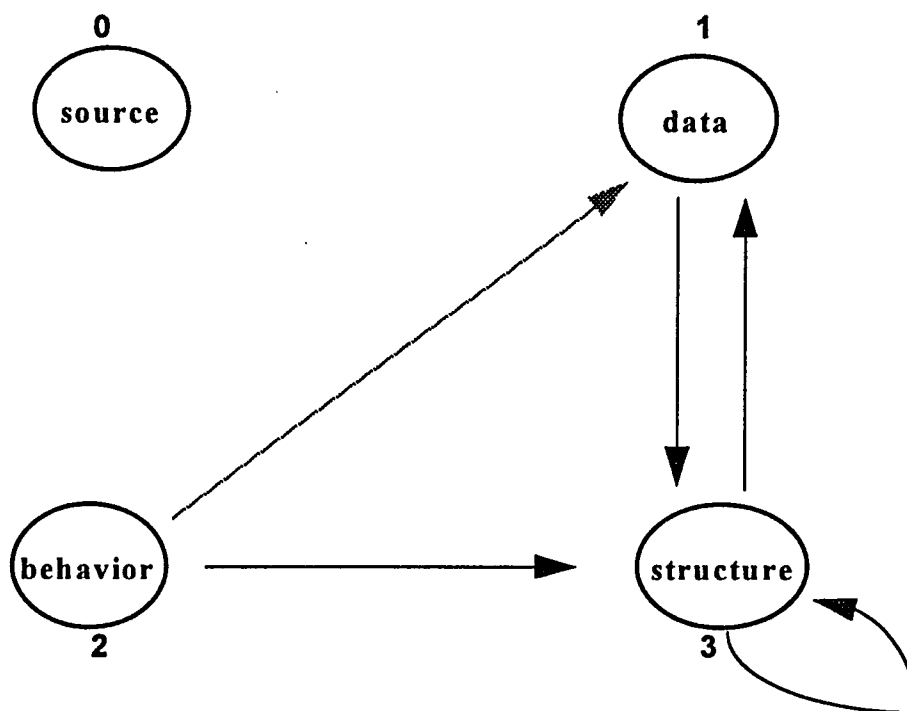


Fig. 4.7 Iteration between data and structure system.

The problem solving process of above example can be expressed in the formalism as following pseudo code:

$\eta_2(M_{2,3}, \Gamma_2) = \mu_{2,3,k}$;select a transformation methodology.

$\mu_{2,3,k} : \Sigma_2 \rightarrow \Sigma_3$;transform from behavior to structure
;systems.

$\eta_1(M_{1,3}, \Gamma_1) = \mu_{1,3}$;select a transformation methodology.

$\mu_{1,3} : \Sigma_1 \rightarrow \Sigma_3$;set up initial data for simulation.

$\eta_3(M_{3,1}, \Gamma_3) = \mu_{3,1,k}$;select a transformation methodology.

$\mu_{3,1,k} : \Sigma_3 \rightarrow \Sigma_1$;run simulation on structure system and
;generate data.

While $\delta_1(\Sigma_1, T_i) = \text{FALSE}$ **Do** ;measure the data generated by
;structure system against the design
;specification and feed back the
;differences.

$\eta_3(M_{3,3}, \Gamma_3) = \mu_{3,3,k}$;select a transformation
;methodology.

$\mu_{3,3,k} : \Sigma_3 \rightarrow \Sigma_3$;refine the structure system based
;on the feedback.

$\eta_1(M_{1,3}, \Gamma_1) = \mu_{1,3,k}$;select a transformation
;methodology.

$\mu_{1,3,k}: \Sigma_1 \rightarrow \Sigma_3$;update the initial condition for
 ;simulation.
 $\eta_3(M_{3,1}, \Gamma_3) = \mu_{3,1,k}$;select a transformation
 ;methodology.
 $\mu_{3,1,k}: \Sigma_3 \rightarrow \Sigma_1$;run simulation again and generate
 ;new data from the structure
 ;system.

End

After the structure systems are finalized in each expert's model, these structure systems are reported to the manager by each expert through the communication links using the communication mechanism defined in our formalism,

Send(SenderId, ReceiverId, KnowledgeType, Knowledge)

Then the manager will integrate all the designs together to form the structure system in the manager's model. There might be a iteration between the data system and the structure system on the manager's roadmap for integration test and modification. The manager is responsible for making sure the final structure system meets the overall design specification and for completing the car design task (see Fig 4.7).

After all the work is done, the final knowledge structure on each team member's canonical form, ω , is an instanciated structure system which is the final state of the canonical form for the problem solving process in the case study.

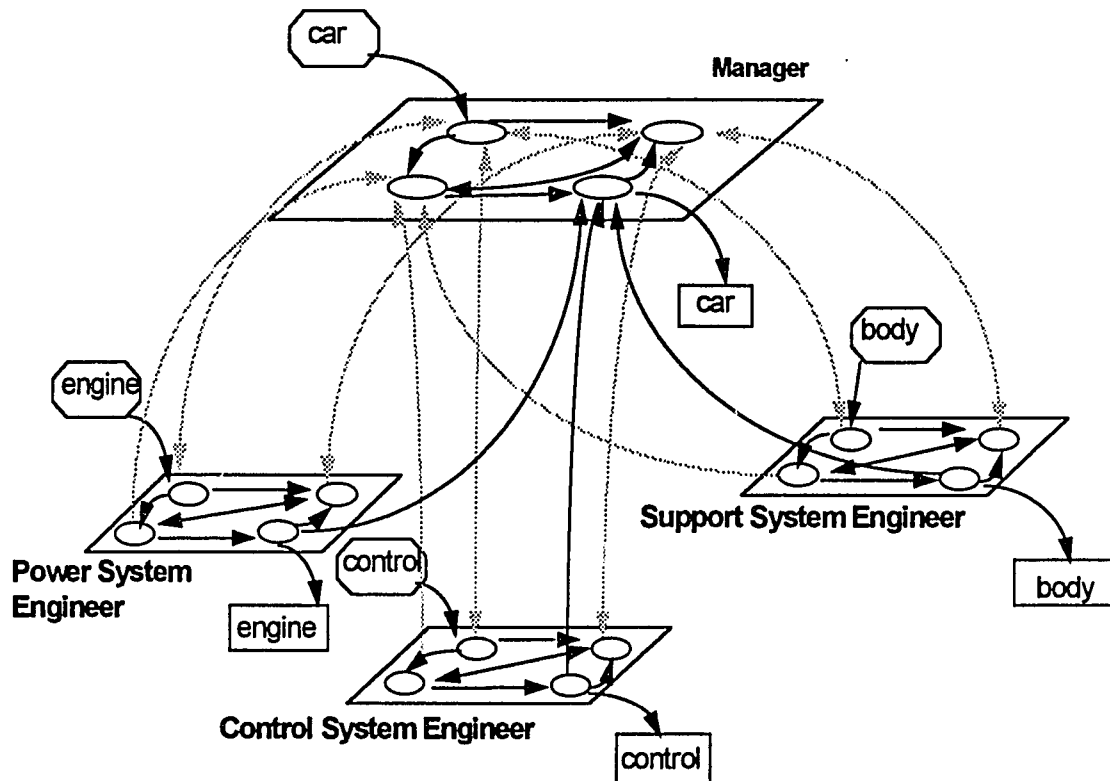


Fig. 4.8 The Structure Systems of the Model

For each expert the instance of the final knowledge structure in the canonical form will be:

- $\omega_A = \Sigma_{mA}$ structure system for the car.
- $\omega_B = \Sigma_{mB}$ structure system for the power system.
- $\omega_C = \Sigma_{mC}$ structure system for the control system.
- $\omega_D = \Sigma_{mD}$ structure system for the support system.

In general, the iteration of problem solving activities are not limited between data system and behavior system nor between data system and structure system on the roadmap. The starting and ending points can be any nodes on the roadmap. There can be different

numbers of nodes on a roadmap which can be involved in the iteration path. A generalized expression for such iteration can be expressed as:

$$\begin{aligned} \eta_i(M_{i,i+1}, \Gamma_i) &= \mu_{i,i+1} && \text{;select a transformation methodology.} \\ \mu_{i,i+1} : \Sigma_i &\rightarrow \Sigma_{i+1} && \text{;transform from } \Sigma_i \text{ to } \Sigma_{i+1} . \\ &: \\ \eta_{j-1}(M_{j-1,j}, \Gamma_j) &= \mu_{j-1,j} && \text{;select a transformation methodology.} \\ \mu_{j-1,j} : \Sigma_{j-1} &\rightarrow \Sigma_j && \text{;transform from } \Sigma_{j-1} \text{ to } \Sigma_j . \\ \text{WHILE } \delta_j(\Sigma_j, T_j) &= \text{FALSE DO} && \text{;measure } \Sigma_j \text{ against } T_j . \\ \eta_j(M_{j,j+1}, \Gamma_j) &= \mu_{j,j+1} && \text{;select a transformation methodology.} \\ \mu_{j,j+1} : \Sigma_j &\rightarrow \Sigma_{j+1} && \text{;transform from } \Sigma_j \text{ to } \Sigma_{j+1} . \\ &: \\ \eta_{k-1,k}(M_{k-1,k}, \Gamma_{k-1}) &= \mu_{k-1,k} && \text{;select a transformation methodology.} \\ \mu_{k-1,k} : \Sigma_{k-1} &\rightarrow \Sigma_k && \text{;transform from } \Sigma_{k-1} \text{ to } \Sigma_k . \\ \eta_k(M_{k,j}, \Gamma_k) &= \mu_{k,j} && \text{;select a transformation methodology.} \\ \mu_{k,j} : \Sigma_k &\rightarrow \Sigma_j && \text{;transform from } \Sigma_k \text{ to } \Sigma_j . \end{aligned}$$

End

Where $\mu_{i,j} \in M_{i,j}$ and $M_{i,j} \in \Pi_i$; $\delta_i \in \Delta_i$, $i = 0$, if it starts at source system.

4.5 Using The Model To Simulate The Problem Solving Strategies

After the model of the car design problem is developed, the model can be used to simulate different scenarios of problem solving strategies. For example, the manager can use the following scenario for cost control strategies.

Since the car engine is the most costly part of the car, the manager might decide to ask the power system engineer start his/her problem solving process first by sending down a set of initial data including unit cost to the power system engineer using **Ask**, **Activate** and **Receive** operations:

Ask (Manager, Power System Engineer, Data System, Data),

Activate (Power System Engineer),

Receive (Power System Engineer, Manager, Data System, Data).

Then the power system engineer starts his/her own problem solving process. The power system engineer solves the relationships defined in the behavior system of his/her roadmap and report the data to the manager using **Send** operation :

Send (Power System Engineer, Manager, Data System, Data).

If it is not feasible for the power system engineer to solve the relationships defined in the behavior system of his/her roadmap under the constraint of the given unit cost. He/She must report that back to the manager. The manager might increase the unit cost for the engine system and ask the power system engineer to re-start his/her problem solving process. This process might repeat a few times before the manager finally accepts the design of the power system under a new unit cost constraint for the power system. Then

the manager re-computes the unit costs for the other subsystems and ask one or both the control system engineer and support system engineer to start their problem solving process as the manager did before for the power system engineer.

There are different problem solving strategies that the manager can apply to achieve different design goals, such as to maximums fuel efficiency or passenger space utilization. All these problem solving strategies can be simulated using our model.

4. 6 Summary

The knowledge being formalized in the model for cooperative problem solving are basic knowledge, meta-knowledge (problem solving strategies of individual experts) and meta² - knowledge (problem solving strategies of the manager). It is possible to use the formalism to model higher level management strategies which may involve multiple levels of meta-knowledge that are higher than level two. The discussion of such models is beyond the scope of this dissertation.

Chapter 5 **Open Areas for Future Research**

Building upon Klir's general systems hierarchy and Orchard and Tausner's work on the problem solving roadmap and basic knowledge definitions, we have explored a new approach towards knowledge engineering . We call this approach systemic knowledge engineering. We have developed a well defined formalism which provides ways of structuring the problem solving expertise of a human expert, formalizing the problem solving strategies using meta-knowledge structures supplemented by control and communication constructs.

This formalism can be used to facilitate (or to drive) the knowledge elicitation process by providing a set of knowledge structures for knowledge engineers to organize the knowledge which has been elicited and help knowledge engineers to better direct their elicitation effort. Therefore, the knowledge engineer can guide the knowledge elicitation process and make it more efficient. This formalism also narrows the gap between knowledge elicitation and knowledge representation by providing a pseudo coded expression of the formalized knowledge which is close to a computer representation. The formalism is machine representation independent. After knowledge has been formalized, a knowledge engineer still has the freedom to selected the best machine representation to build a knowledge-based systems. In case we do not have a "sufficient" machine representation for the elicited knowledge possibly due to the limitations of current technology, the formalism at least provides a conceptual model for the problem solving expertise of a human expert.

In the real world, most complex problem solving require a human expert to deal with uncertainty, when the information available for problem solving is vague and/or ambiguous. We refer to this type of problem solving as problem solving with uncertainty.

The knowledge or expertise involved in such a problem solving process is called uncertain knowledge or uncertain expertise. The problem solving strategies used in such problem solving processes are called uncertain problem solving strategies. In this dissertation, the concepts and definitions in the formalism are developed without explicitly dealing with uncertain knowledge formalization. But the formalism provides a general framework which can be extended to encompass uncertain knowledge Formalization.

To formalize uncertain knowledge, further research is needed in several areas: the generalization of basic knowledge structures to formalize basic uncertain knowledge, the generalization of meta knowledge structures to formalize meta uncertain knowledge.

Klir [1988] has classified uncertainty into two categories, vagueness and ambiguity. Vagueness is associated with the difficulty of making sharp or precise distinctions. Ambiguity is associated with one-to-many relations where the choice between two or more alternatives is not certain.

A initial research is being conducted currently to extend the problem solving roadmap [Bashias and Tausner, 95]. This research is aimed at formalizing basic uncertain knowledge in terms of uncertain source systems, uncertain data systems, uncertain behavior systems and uncertain structure systems as well as methodologies associated with these basic uncertain knowledge structures.

Research is also needed to extend meta knowledge structures to formalize uncertain meta knowledge, such as uncertain problem solving strategies which include decision making and reasoning with uncertainty. Based on our model there are two kinds of decision making at each junction of problem solving trajectory, decision making based on the measurement of the degree of satisfaction associated with the knowledge structures and

decision making to select a particular methodology. Both types of uncertain, vagueness and ambiguity, exist in above decision making processes. Therefore uncertain meta knowledge is the key to form uncertain problem solving strategies. We are looking forward to integrating some of the recent research results in the areas of fuzzy sets, fuzzy logic, and fuzzy measure theory [Klir & Folger, 1988 and Wang & Klir, 1992] to extend our formalism.

BIBLIOGRAPHY

Addis, T. R., [1987], A Framework for Knowledge Elicitation. *Proceedings of the First European Conference on Knowledge Acquisition for Knowledge Based Systems* (Section A1).

Ashby, W. R. [1956], *An Introduction to Cybernetics*. Wiley, New York.

Bashias, N. J and **Tausner, R. M.**, [1995]. On the Systemic Formalization of Uncertain Knowledge to be submitted.

Bueno, A. D. [1995], *Inductive Reasoning and Reconstruction Analysis*. Ph.D dissertation, Polytechnic University of Catalunya

Breuker, J., Wielinga, B., Van Someren, M., De Hoog, R., Schreiber, G., Bredeweg, B., Wielemaker, J., Billault, J-P., Davoodi, M., and Hayward, S., [1987], Model-Driven Knowledge Acquisition: Interpretation Models. *Deliverable Task AI, Esprit Project 1098 Memo 87, 1987*.

Castells, A. N. [1994], *Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning*. Ph.D dissertation, Polytechnic University of Catalunya

Cellier, F. E., [1991], *Continuous System Modeling*. Springer, New York, 1991.

Davis, R., [1982], Expert Systems: Where Are We? and Where Do We Go From Here?, *AI Magazine*, Vol. 3, No. 2, 1982

Dreyfus, H. and Dreyfus, S., [1986], *Mind Over Machine: The Power of Human Intuition and Expertise in the Era of Computer*. New York, NY, The Free Press, 1986

Feigenbaum, E. A., and **McCorduck, P.**, [1983], *The Fifth Generation*. Reading, Mass. Addison Wesley.

Firebaugh, M. W. [1988], *ARTIFICIAL INTELLIGENCE: A Knowledge-Based Approach*. PWS-KENT, Boston.

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

Klir, G. J. [1969], *An Approach to General Systems Theory*. Van Nostrand Reinhold, New York.

Klir, G. J. [1972], (Ed.), *Trends in General System Theory*. Wiley-Interscience, New York.

Klir, G. J. [1985], *Architecture of Systems Problem Solving*. Plenum Press, New York.

Klir, G. J. and Folger T. A. [1988], *Fuzzy sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, New Jersey.

Klir, G. J., [1990], A Principle of Uncertainty and Information Invariance. *International Journal of General Systems*, Vol. 17, pp. 249-275.

Luger, G. F., and Stubblefield, W. A. [1989], *ARTIFICIAL INTELLIGENCE: and the Design of Expert Systems*. Benjamin and Cummings, California.

Minsky M. (Ed.) [1968], *Semantic Information Processing* The MIT Press, Massachusetts.

Orchard, R. A. and Tausner, M. R. [1987a], The Foundational Role of General Systems Formalizations in Knowledge Engineering. *31st Annual Meeting of the International Society For General Systems Research*, Vol. II, June 1987, Budapest, Hungary.

Orchard, R. A. and Tausner, M. R. [1987b], Transformation of Knowledge in General Systems Formalizations to Expert System Representations. *31st Annual Meeting of the International Society For General Systems Research*, Vol. II, June 1987, Budapest, Hungary.

Orchard, R. A. and Tausner, M. R. [1988], General Systems: a Basis for Knowledge Engineering. *Systems Practice*, Vol. 1, No. 2, 1988.

Orchard, R. A., Reese, E. J. and Tausner, M. R. [1989], On the Foundations of Knowledge Engineering. *Modeling and Simulation Methodology; Knowledge Systems Paradigms*, Elzas, M., Oren, T. and Zeigler, B. (eds.), 1989, North-Holland, Amsterdam.

Orchard, R. A. [1990], Knowledge Processing: A Semantics for the Klir Hierarchy of General Systems. *Proceedings of Computer Aided Systems Theory -EUROCAST'89*, F. Pichler and R. Moreno-Diaz (Eds). Springer-Verlag, pp. 33-40, 1990, Berlin Heidelberg.

Parsaye, K. and Chignell, M. [1988], *Expert Systems for Experts*. John Wiley & Sons, Inc., New York, pp. 119.

Rosen, R. [1986], Some Comments on Systems and System Theory. *International Journal of General System*. Vol. 13, pp. 1-3.

Shannon, C. E., A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 1948.

Slatter, P. E., [1990], Models of Expertise in Knowledge Engineering. *Knowledge Engineering*, Vol.1, Adeli. H. (Ed), pp.130-157, McGraw-Hill.

Tastle, W., (editor), [1995], A Special Issue on General Systems Problem Solving. *International Journal of General Systems*, Vol. 24, No. 1 and No 2, 1995.

Tausner, M. R., [1986]. *Formalization and Representation of Expert Systems*. University Microfilms, Ann Arbor, Michigan.

Tausner, M. R. and Orchard R. A., [1986]. A Formalism For Expert Systems With Front-End/Back-End Architecture. *Proceedings 1986 SGSR International Conference*. Philadelphia, Pennsylvania, pp. G_52 - G_61.

Wang, Z. and Klir, G. J. [1992], *Fuzzy Measure Theory*. Plenum Press, New York.