ƦY

TO: ALL

Please sign *e used this thesis in* *accordanc* *author of this* *thesis.*

Thank you.

SUE McKI
University

Name	Date
John	12/8/97
KEVIN	15/10/01
DAVID J	14/10/02
Rodzi Jidin	15/12/03

HYPOTHESIS GENERATION FOR MANAGEMENT INTELLIGENCE

by

Fan CHEN (B.Sc. M.Sc.)

This thesis is submitted in total fulfillment for the degree of

Doctor of Philosophy

in the Faculty of Science and Technology,

Deakin University, Geelong, Victoria, Australia.

May, 1996

DEAKIN UNIVERSITY

Deakin University

Geelong, Victoria, Australia

DEAKIN UNIVERSITY

CANDIDATE CERTIFICATE

I certify that the thesis entitle HYPOTHESIS GENERATION FOR MANAGEMENT INTELLIGENCE and submitted for the degree of DOCTOR OF PHILOSOPHY is the result of my own research, except where otherwise acknowledged, and that this thesis or in part has not been submitted for an award including a higher degree to any other university or institution.

Name: FAN CHEN

Signed: Signature Redacted by Library

Date: 15/05/96.

DEAKIN UNIVERSITY

DISPOSITION OF THESIS

I am the author of the thesis entitled Hypothesis Generation For Management Intelligence submitted for the degree of Doctor of Philosophy and agree to the thesis being made available for such consultation, loan or photocopying as may be approved by the University Librarian, provided that no part of the thesis shall be reproduced without the prior approval of the University Librarian and with the appropriate acknowledgment of the source.

Name: FAN CHEN

Signature: Signature Redacted by Library

Date: 20/02/97

Abstract

Investigation of the role of hypothesis formation in complex (business) problem solving has resulted in a new approach to hypothesis generation. A prototypical hypothesis generation paradigm for management intelligence has been developed, reflecting a widespread need to support management in such areas as fraud detection and intelligent decision analysis.

This dissertation presents this new paradigm and its application to goal directed problem solving methodologies, including case based reasoning. The hypothesis generation model, which is supported by a dynamic hypothesis space, consists of three components, namely, *Anomaly Detection*, *Abductive Reasoning*, and *Conflict Resolution* models.

Anomaly detection activates the hypothesis generation model by scanning anomalous data and relations in its working environment. The respective heuristics are activated by initial indications of anomalous behaviour based on evidence from historical patterns, linkages with other cases, inconsistencies, etc.

Abductive reasoning, as implemented in this paradigm, is based on joining conceptual graphs, and provides an inference process that can incorporate a new observation into a world model by determining what assumptions should be added to the world, so that it can explain new observations. Abductive inference is a weak mechanism for generating explanation and hypothesis. Although a practical conclusion cannot be guaranteed, the cues provided by the inference are very beneficial.

Conflict resolution is crucial for the evaluation of explanations, especially those generated by a weak (abduction) mechanism.The measurements developed in this research for expla-

nation and hypothesis provide an indirect way of estimating the "quality' of an explanation for given evidence. Such methods are realistic for complex domains such as fraud detection, where the prevailing hypothesis may not always be relevant to the new evidence.

In order to survive in rapidly changing environments, it is necessary to bridge the gap that exists between the system's view of the world and reality.Our research has demonstrated the value of *Case-Based Interaction*, which utilises an hypothesis structure for the representation of relevant planning and strategic knowledge. Under the guidance of case based interaction, users are active agents empowered by system knowledge, and the system acquires its auxiliary information/knowledge from this external source.

Case studies using the new paradigm and drawn from the insurance industry have attracted wide interest. A prototypical system of fraud detection for motor vehicle insurance based on an hypothesis guided problem solving mechanism is now under commercial development. The initial feedback from claims managers is promising.

Acknowledgments

I am deeply grateful for the help and encouragement of many people throughout the long period of this research. First and foremost, I would like to thank my supervisor, Professor Brian J. Garner for his never ending patience and guidance during the period of research and preparation of this thesis, his support, encouragement and advice have all been invaluable and will never be forgotten.

Several people have provided tremendous support and encouragement during this research. I would like to take this opportunity to thank Dr. Dickson Lukose, Dr. C. Forrester, Dr. Swamy Kutti and Mr. John Liu for many hours of discussion on various aspects of knowledge engineering. I would also like to thank Professor John Sowa, State University of New York and Professor E. Edmond, University of Technology, Loughborough, for their discussion and encouragement.

To the people at GIO Australia, especial Mr. Paul Harvey, thanks you for the most cooperative assistance while I am carrying on the case study. Their helps make it possible for this research to be recognised the potential commercial value. Sincere thanks go to Robyn McConnell and Paul Beinat of AI Laboratory, Continuum Australia for their recognising the potential value of my research, and providing the unique chance for further development.

I am also appreciated the Department of Employment, Education and Training, Commonwealth of Australia, and Deakin University for their generous providing scholarships. Without these financial supports it is impossible for me to finish this research.

Finally, for the spiritual support, understanding and encouragement of my family throughout the past years, I am heartful appreciated to them all.

Table of Contents

CHAPTER 2

CHAPTER 3

CHAPTER 8

CHAPTER 9

Table of Figures

CHAPTER 1

Overview of the Thesis

1.1 Introduction: From Holmes to Einstein

All problem solvers generate hypotheses [118]. In so far as hypotheses go beyond the experimental observations, all our empirical knowledge about the world may be said to consist of hypotheses. Hypotheses represent a form of plausible explanation derived from previous knowledge and new facts. In the processes of human problem solving, hypothesis serves as purposive guidance. There is strong evidence (loc cit) for the view that hypothesis generation is a basic requirement in complex problem solving.

The origins of hypothesis as a guidance mechanism for reasoning lies in antiquity. It is mostly recorded in detective novels and scientific legends. Hypothesis based problem solving as a detective methodology is exemplified by Sherlock Holmes, who was a genius at generating hypotheses, and described the process as reasoning "backwards"[39]. On their first meeting, Holmes hypothesises that Dr. Watson had just recently came back from Afghanistan. Dr. Watson naturally supposed that someone had told him.

> "You were told, no doubt."

> "Nothing of the sort. I knew you came from Afghanistan. From long habit the train of thoughts ran so swiftly through my mind that I arrived at the conclusion without being conscious of intermediate steps. There were such steps, however. The train of reasoning ran, "Here is a gentleman of a medical type, but with the air of a military man. Clearly an army doctor, then, he has just come from the tropics, for his face is dark, and that is not the natural tint of his skin, for his wrists are fair. He has undergone hardship and sickness, as his haggard face says clearly. His left arm has been injured. He holds it in a stiff and unnatural manner. Where in the tropics could an English army doctor have seen much hardship and got his arm wounded? Clearly in Afghanistan.' The whole train of thought did not occupy a second. I then remarked that you came from Afghanistan,..."[31]

In the detection process, Holmes demonstrated a masterful ability to link observation and deduction in hypothesis generation. In the story of "The Lauriston Garden Mystery[31]" Holmes generated an hypothesis based on first-hand evidence such as the ruts made by a cab, the marks of horse's hoofs, the cigar ash, et. al. The hypothesis which astonished Dr. Watson was:

> "There has been murder done, and the murderer was a man. He was more than six feet in height, wore coarse, square-toed boots and smoked a Trichinopoly cigar. He came here with his victim in a four wheeled cab, which was drawn by a horse with three old shoes and one new one on his off fore-leg. In all probability the murderer had a florid face, and the fingernails on his right hand were remarkably long."

Great scientists appear to emulate the genre of Sherlock Holmes, for they are all apparently masters in abductively generating hypotheses. In the history of science, Newton generated the hypothesis of ***Absolute Space and Time*** from the evidence of an apple falling to the ground. Einstein developed his hypothesis of ***Theory of Relativity*** by imagining what would happen and what would the person see in travelling at the speed of light.

These examples show that the methods of experimental science have much in common with the methods of detectives. A fundamental issue, which we address in the thesis, besides the psychological aspects, is the nature of the logic involved. When Einstein and Holmes solved problems, they felt (afterwards) compelled to give reasons (explanations) for correct hypotheses. Similarly, whenever a scientist proposed a hypothesis to account for some facts, peer review ensured that the derived explanations were scientifically sound. The selection of a candidate hypothesis is presumed to be a logical matter, decidable on a rational basis. A natural conclusion is that between scientist and detective there are common elements for experimental investigation and a common process for hypothesis generation.

In developing a computational paradigm for hypothesis generation, the notion of hypothesis as a knowledge (conceptual) structure has been developed, and a reasoning process investigated for hypothesis selection. The philosopher Charles Sanders Peirce names three main considerations that should guide our choice of a hypothesis[118]:

1. A hypothesis must explain the facts at hand,
2. It must be capable of experimental confirmation, and
3. It must be guided by economic considerations.

It follows that the presumption of hypothesis as derived knowledge can be restated as an axiom suitable for experimental verification.

AXIOM 1: Hypothesis verification is amenable to experimental study of its supporting propositions (facts) and validity of derived explanations.

The underlying structure of an hypothesis thus requires consideration of a set of non-conflicting explanations to a collection of evidence, together with implied/embedded strategies for self-verification. Any hypothesis has a truth value defined by searching supporting evidence. We should admit that on the basis of the supporting evidence actually available, some hypotheses will be quite credible, some will be quite unlikely, but others will have intermediate degrees of credibility. Empirical knowledge provides a method for correctly assigning various hypotheses a degree of credibility appropriate to each on the basis of the actual evidence.

In study of novices and experts in complex problem solving, experts were found to generate hypotheses more effectively and efficiently. However, once the hypothesis is generated the process of testing can proceed, and is done by deducing from the hypothesis, consequences which can be tested experimentally or by observation. This is what is known as the hypothesis-deduction method. We also expect differences in the breadth and depth of search due to the level of expertise.

The modern concept of a hypothesis in the field of artificial intelligence was exemplified by Charles Morgan in his writing on Hypothesis Generation by Machine[104]. Recently researchers in artificial intelligence have begun to notice the importance of hypothesis in problem solving[21][29][50][77][129]. Most of this research, however, has been restricted to the construction of hypothesis instead of hypothesis generation [76]. One of the more significant differences between hypothesis assembly and hypothesis generation is the capability to discover some new significance, the ability to exercise creativity. This research demonstrates that hypothesis generation is a basic requirement of reasoning processes and will provide a new paradigm for complex problem solving in the real world.

1.2 Hypothesis from the View of Philosophy

The Baconian doctrine of the origin of scientific hypotheses as a kind of distillation from innumerable factual observations, altogether devoid of theoretical bias, is seen by both Kneale [85]and Popper [123] to be futile in practice and untrue of science.

> "The advance of science is not due to the fact that more and more perceptual experiences accumulate in course of time.... Out of uninterpreted sense-experience science cannot be distilled, no matter how industrious we gather and sort them."[123]

Hypothesis generation was sometimes thought of as an almost mystical process, where a creative act produces something out of nothing. Under the influence of this mystery, it was believed that experts reach hypothesis by guess, and implying there is no method of discovery. The production of a scientific hypothesis remains in a mystery. It seemed only psychologists could reveal the intuitive process of inspired guesswork. However, there appears to be a fundamental flaw in any attempt to consider hypothesis generation as something that transcends current knowledge. No Human discovery occurs in a vacuum: If we really accepted that creative acts must build something out of nothing, we would be hard-pressed to demonstrate that human creativity exists[145].

It seems obvious that to generate a hypothesis, we must start with existing knowledge and past experience. New hypothesis comes from retrieving knowledge that is not routinely applied to a situation, or to give a different explanation to an old situation.

Hypothesis is described by Peirce as an argument "which proceeds upon the assumption that a character which is known necessarily to involve a certain number of others, may be probably predicated of any object which has all the characters which this character is known to involve" [118]. Explanatory hypotheses may be of widely different kinds and Peirce points out at least three[39]:

1. The kind which refers to facts unobserved when hypotheses are made, but which are capable of being observed.
2. There are hypotheses which are incapable of being observed This is the case of historical facts.
3. Finally, hypotheses may refer to entities which in the present state of knowledge are both factually and theoretically unobservable.

In order that the process of making an hypothesis should lead to a probable result, Peirce lists three rules which must be followed:

1. The hypothesis should be distinctly put as a question, before making the observations which are to test its truth. In other words, we must try to see what the result of predictions from the hypotheses will be.
2. The respect in regard to which the resemblances are noted must be

taken at random. We must not take a particular kind of prediction for which the hypothesis is known to be good.

3. The failure as well as the success of the predictions must be honestly noted. The whole proceeding must be fair and unbiased.

Although they were used interchangeably at an early stage, hypothesis and induction are separate forms of inference according to the present theory. "The essence of an induction is that it infers from one set of facts to another set of similar facts, whereas hypothesis infers from facts of one kind to facts of another[2]".

AXIOM 2: Hypothesis Generation based solely on induction constrains the hypothesis space.

1.3 Hypothesis and Its Role in Human Problem Solving

Questions about the origins of scientific reasoning have been posed by developmental psychologists many times in the past 60 years[32]. A number of broad questions about the nature of scientific reasoning require contextual knowledge. Within psychology, one approach to these questions has been to consider science as a form of problem solving. The science-as-problem-solving view is stated most explicitly in Herbert Simon's characterization of scientific discovery as a form of search and in his elucidation of many of the principles that guide this search[148]. For instance, he has used the notion of search in a problem space to analyze what science is, how scientists reason, and how scientists should reason.

From cognitive psychology, it is well known that one's ability to process information (perceiving, coding, storing, recognizing, retrieving etc.) is a function of past experience and can be improved through learning. The manner in which information is processed is a function of the structure and organization of the human mind, which is dynamic and adaptive in nature[8].

The relationship between hypothesis and discovery was pointed out by Herbert Simon when making comments on his own research program:

> The hypothesis that drives this research is that scientific discovery is a problem solving activity like other problem solving activities that human beings engage in, using the same basic information-processes. This hypothesis rests, in turn, on the belief that the scientist does not stand outside the lawful scheme of Nature; he is part of that scheme, and it is an important goal of scientific research to understand his mental processes, just as it is to understand the processes of a star, an atom, or a cell[148].

The research conducted by psychologists at Carnegie-Mellon University[32] identified three major differences between adults and children. First, children proposed hypotheses for that were different from adults. Second, the children did not abandon their current frame (hypothesis space) and search the hypothesis space to form a new frame, or use the results of experimental (space) search to induce a new frame. Third, the children did not attempt to check whether their hypotheses were consistent with prior data. From their results, we can conclude that the hypothesis generated reflects the experience and maturity of the individual.

According to the research conducted by psychologists in Germany[87], novice-expert comparisons in 'semantically rich domains' revealed that in diagnostic reasoning, experts show more flexibility in the interpretation of data and in the modification of diagnostic hypothesis. Flexibility in problem solving also encompasses the selection and application of search strategies. It was also identified that novices, after having selected a single diagnostic category, would retain the assumption for a longer time than experts would. We would also expected differences in the breadth and depth of search due to the level of expertise.

The same source also showed that experts verify hypotheses in a more limited search space. This points to a depth-first search strategy of experienced diagnosticians compared to the breadth first approach of novices. Confronted with very unspecific symptoms, the experts' set of hypotheses is comparatively small. It suffices by using efficiency criteria in finding the cause of a malfunction.

Hypothesis guided problem solving is the essence of how human reasoning appears to work. People generate hypotheses from experience! They use their own experiences if relevant, or they make use of the experience of others to the extent that they can obtain information about such experiences. An individual's knowledge may well be the collection of personal experiences or that he has access to. People's ability to generate hypotheses is grounded in the accumulation of experience.

In essence, problem solving is a two part process. There is a data-driven side, where making observations, filtering them, and generating hypotheses from them is important. There is also another side, that is essentially goal driven: evaluation of hypotheses against new data can, and does, lead to reformulation or even rejection of preliminary hypotheses[68].

The functions of hypothesis in complex problem solving are seen to be:

1. Hypothesis bridges the difference between human cognition and real world reality.
2. Hypothesis provides an inference engine to perform deductive reasoning, and to focus inference strategy on interesting aspects of problem solving.
3. Hypothesis serves as an heuristic device to assist the user test the strengths, weaknesses, ramifications of an analysis or argument by exploring and augmenting the space of known cases and indirectly, the attendant requirements for explanation and conflict resolution.
4. Problem solving ability is dependent upon the ability to constitute a suitable hypothesis from the facts available.
5. The difference in complicated problem solving (non-routine) lies in the different requirements for hypothesis generation.

In order to improve the machine assisted problem solving capability, it is necessary to investigate possible paradigms for hypothesis generation. Although there is still room for debate, the dimension for productive experimentation lies in our study and understanding of such paradigms.

1.4 The Importance of Hypothesis Generation

If hypotheses do not spring from the brain of Zeus, where then do they come from? A plausible answer is that they stem from some hypothesis generation processing[148]. However, a convincing experimental framework for elucidating the nature of this process is, as yet, undefined.

> **AXIOM 3**: The goal of an hypothesis generation paradigm is to model possible processes whereby hypotheses, whether fully formed or incomplete, may be generated.

Studies of the contemporary literature on scientific reasoning has identified two main approaches. The first is by searching memory for a hypothesis space that could be used to generate an hypothesis. The second is by conducting experiments and evaluating prototypical knowledge structures from the results of these experiments. Once a hypothesis has been generated, values must be assigned to the objects so that a specific hypothesis can be generated.

Based on Simon and Lea's Generalized Rule Inducer (GRI)[150], Dunbar and Klahr proposed a model of Scientific Discovery as Dual Search (SDDS)[32]. The fundamental assumption is that scientific reasoning requires search in two related problem spaces: an hypothesis space, consisting of the hypotheses generated during the discovery process, and an experimental space, consisting of all possible experiments that could be conducted. Search in hypothesis space is guided both by prior knowledge and by experimental results. Search in the experimental space may be guided by the current hypothesis, and it may be used to generate information to formulate new hypotheses.

These researchers have recognized that an hypothesis guided deductive process is a psychologically plausible cognitive model [132] of human expertise. The author has distinguished the following ways of applying hypothesis by human experts:

1. as a knowledge frame, in which problem solving is the process of knowledge instantiation using knowledge either from previous experience, or from new observations.

2. as an inference strategy which will invoke the desired inference processes using appropriate inference engines with condition branches.
3. as a problem solving benchmark using high level primitives and related actor models [95].
4. as a discovery tool (e.g. logic programming) for probing the real world state.

It should also be noted that modern approaches to management intelligence requirements, including data mining, appear to use all four ways of applying hypothesis directed search.

1.5 Management Intelligence Systems and Hypothesis Generation

Advancements in computational paradigms have had a significant impact on how humans and their organizations are able to improve performance. In recent years, the business community has become aware of the need for Management Intelligence Systems (MINTS)[7][81][48][161]. This is in large part the result of increasing global competition, which makes the availability of machine (assisted) intelligence more important in managing the complexity of contemporary business and society in a timely and efficient manner. The advent of global telecommunications networks for electronic trading, for example, has eliminated manual options for management control and audit. The price, power and usability of computers now makes them indispensable tools for business and governments.

A management intelligence system is intended by the organization it serves to scan the organization's environment so that management can better assess its position with a view to enhancing the value of the organization and its services.

When discussing the role of intelligence systems M. Kochen commented:

> Contemporary intelligence systems rely far more on overt, public sources than on covert espionage missions. They must screen, evaluate, correlate, interpret, analyze and synthesize vastly more

> information. These activities require judgment, hypothesis-formation, reasoning, and a great deal of knowledge, understanding, and intelligence. They are performed by persons with the help of computers[80].

But it takes them a long time. It often demands more talented people than can be mobilized and paid to produce high-quality intelligence. The timing and complexity of what an organization must accomplish to prosper in a highly competitive environment are becoming critical. If the capabilities of human intelligence analysts and managers in an organization can be amplified to produce higher quality intelligence much more quickly, and in the face of vast quantities of data, then new opportunities for market leadership and profitability may be discovered, while recognising that ultimately, a superior management intelligence capability is necessary to beat competitors.

A concluding statement from M. Kochen is worth considering in development of strategic information systems:

> Therefore, the management intelligence systems required by competing organizations in government and business will be the best they can obtain. The best likely to make good use of advanced technology, such as artificial intelligence. The purpose of such man-machine MINTSs is to support professional strategists, planners, and researchers as would a good semi-automated research assistant, enabling the strategists to produce better plans, more quickly and at lower cost[80].

1.6 Themes of the Research

Development of a management intelligence system employs all the procedures used in developing any computerized information system, such as requirements analysis, rapid prototyping, feasibility studies and implementation [80]. It also requires some unique functions such as environmental scanning and hypothesis generation.

Throughout this research, four fundamental research themes involved in management intelligence systems have been identified. They are an hypothesis generation model, abductive inference engine, anomaly detection, and human-computer interaction each making a unique contribution to the success of the ensuing management intelligence system.

1.6.1 Hypothesis generation model

In management intelligence systems, the most important function is carried out by an *hypothesis generation model* (HGM). The kernel of this model is an h*ypothesis space* (HS), which is a dynamic knowledge structure, including a lattice consisting of nodes and relations.

The nodes in my hypothesis space represent evidences, facts, and conclusions. The nodes are connected by various relations. Two new forms of relations have been identified in this research, they are: explanation relations, and executable relations. This unique relation classification provides the hypothesis space with an ability to employ executable knowledge, and more importantly, the flexibility to perform abductive reasoning, generating explanations and hypotheses.

Existing hypotheses are concealed in hypothesis space in the idle state. All hypotheses are linked and overlap each other in the hypothesis space. A summary of the key features is given below:

1. *hypothesis* is a set of nodes in the hypothesis space connected by relations, and is a part of the total hypothesis space.
2. when there is new evidence which cannot be matched by hypothesis space, the system has the ability to create a new relation to connect the (apparently related) evidence into the existing hypothesis space.
3. an hypothesis will be activated from hypothesis space by an evidence indexing mechanism.
4. when all the index evidence can be matched to the hypothesis space, the system defaults to a case based reasoning system.

Case-based Explanation[143][145][79], which is a parallel research topic, replaces explanation by chaining with a memory-based approach: It builds a new hypothesis by retrieving stored explanations for prior cases, evaluating their applicability and usefulness for the new situation, and adapting them as needed. By comparison, the main contribution of the hypothesis generation model lies in its attempt to use the knowledge from a generalised hypothesis space instead of selection and repair of packaged explanations.

In this model of hypothesis generation, there are three components supporting each other to empower the hypothesis generation ability. They are as shown in FIGURE 1.1.

1. Abduction,
2. Inference strategies, and
3. Abstraction (conceptual hierarchical structures).

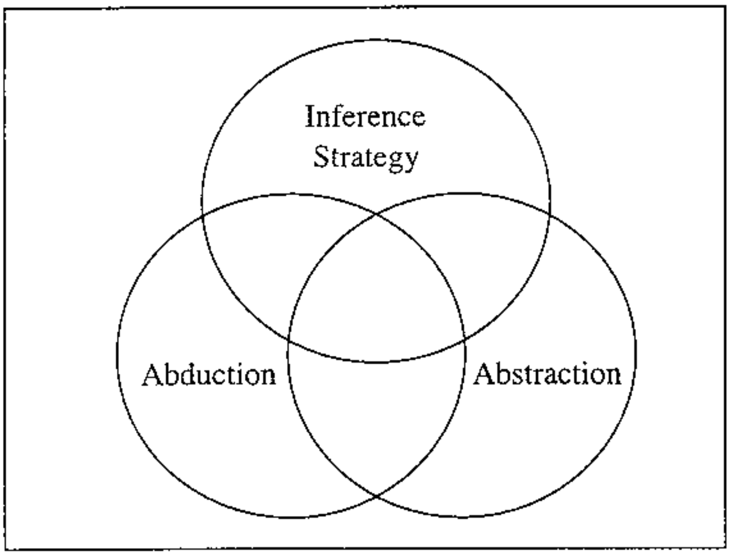


FIGURE 1.1 Hypothesis Generation Model

An hypothesis is generated from a combination of all relevant evidences and domain knowledge in long-term memory. Hypothesis is regarded as new knowledge which is presumed to enhance previous knowledge (consistent with previous evidence), and represents a kind of possible knowledge (subject to further evaluation), or may be adopted as new knowledge on validation with new findings.

Once the hypothesis is generated, the intelligence analysis process can proceed, by deducing from the hypothesis, predictions which can be tested experimentally or by observation (hypothesis-deduction method) of known situations.

1.6.2 Anomaly detection

As described earlier, the main function of management intelligence systems is to scan database or on-line information sources, and then provide an intelligence report for high level decision support. It is obvious that anomaly detection is a vital step for hypothesis generation. In our *Anomaly Detection Model* (ADM), the detection process is divided into two stages: anomalous data detection and anomalous relations detection.

Theoretically, anomaly (data) detection may be simplified in terms of statistical deviations from other data in the samples. There are various methods to deal with such detection requirements[3][97][84]. This function can be partially automated with the help of an expert system if criteria for data evaluation or judgement can be expressed in the form of rules.

Although scanning anomalous data in a structured environment is easy to perform, the search for anomalous relationships among the data is complicated. One idea is to detect anomalous patterns that fail to correspond to any known pattern or exhibits unusual variations from known patterns[145].

In our research, the main interest was focussed on detecting anomalous relations. The methods can be seen as hypothesis guided detection, which attempts to establish new relations in the data set. The respective heuristics are activated by initial indications of anomalous behavior, based on evidence from:

- historical pattern of suspicious events (e.g. fraud patterns known to insurers);
- linkage with other (suspicious) circumstances;
- facts determined in the course of verifying explanations;

- inconsistencies, as determined by reference to other knowledge sources; and
- model-directed discovery of new methods for verifying the information provided.

The ability to combine knowledge from multiple sources in hypothesis generation is an essential requirement for the success of the process.

1.6.3 Case based interaction

In this thesis, cooperative problem solving, which refers to the cooperation between a user and a computer system, facilitates the process of solving complex problems by utilizing knowledge from both sides of the interface, namely user and system. In such cooperative systems, users are active agents empowered by the systems' knowledge, and auxiliary (expanded) knowledge sources are identified by the user. Cooperative problem solving enables the strengths of both partners to be exploited to the full through interaction between the user and the computer.

The communication module plays a key role in information or knowledge interchange through critiquing of user and computer responses in facilitating HCI metaphors[34]. Computers provide external memory for the user, insure consistency, hide irrelevant information and summarize and visualize information [44]. And humans work like an extra knowledge base from the point of view of systems. The communication module brings the power of both sides together.

A satisfactory environment for creative decision support should ideally satisfy the following:

- An adaptive interface to the user capable of accepting input in free form (natural language) and delivering information output that is comprehensible and acceptable to the decision maker.
- Provision of the necessary information required by the user without any constraints on user behavior or the decision process.

- The support must be "intelligent" in directing the decision maker to appropriate decision models[46].

1.7 Overview of the Thesis

The thematic purpose of *hypothesis* and its historical development have been reviewed from the viewpoint of psychology and artificial intelligence. The aims of my research were then outlined with the requisite axioms for experimental study.

An overview of contemporary information systems and knowledge engineering techniques relevant to hypothesis generation is provided in Chapter 2. This review spans the main areas of research interest from artificial intelligence to information systems, such as *conceptual graphs for knowledge representation, case based reasoning, knowledge acquisition, abductive reasoning, executable knowledge structures, on-line information retrieval, strategic information systems*, and *human computer interaction.*

Chapter 3 addresses the architecture and design of management intelligence systems focussing on the analysed requirement for a novel conceptual framework accommodating expanded anomaly detection and hypothesis generation capabilities.

The details of my hypothesis generation model are discussed in chapter 4, which outlines the design, including data structures, for the representation of hypothesis and the hypothesis space. The design of declarative and procedural knowledge for use within the hypothesis structure is then demonstrated.

In Chapter 5, the computational issues in implementing the new constructs are discussed. In this implementation, an hypothesis can be considered as an autonomous, intelligent agent, which will be called upon to perform a varied range of tasks under a wide range of circumstances. From the view of object-oriented design, hypothesis can be seen as a Superobject, which contains both data and related procedures (actor model).

Chapter 6 describes an abductive reasoning model, and presents an improved algorithm for a conceptual graph based abductive reasoning engine, combining the advantage of a generalized set covering model and maximal-join operation, with the aim of generating a suitable hypothesis.

Chapter 7 concentrated on the topic of hypothesis evaluation and conflict resolution. The acceptability of explanation and hypothesis is proposed for technical evaluation to determine the quality of the hypothesis generation process. A set of strategies for conflict resolution is then proposed.

In chapter 8, the principle of case based interaction [35][49] and the basic requirements for management intelligence systems are linked to enhance cooperative problem solving through utilization of knowledge from both sides of the interface: user and systems.

In chapter 9, an anomaly detection model has been developed for incorporation in management intelligence systems to monitor user environments. This model is novel and mutually compatible with the hypothesis generation model described in chapter 4. Finally, a case study on fraud detection is presented.

Chapter 10 provides a summary of conclusions, and expands on the basis for evaluation of hypothesis generation model. The ideas for extension and direction of future research are also included.

1.8 Significance of the Research

There is ample evidence supporting the view that hypothesis plays an important role in scientific discovery and in complex problem solving [39][118]. This view is of increasing interest to psychologists[32][148]. The research results presented here are seen to offer a novel and original model for studying cognitive processes of human experts in complex problem solving.

The generalised hypothesis generation paradigm provides complementary reasoning capabilities to other problem solving paradigms and, in particular, extends the power of anomaly detection processes through hypothesis guided search. The hypothesis guided process elaborated here is believed to provide a psychologically plausible cognitive model for human reasoning (deductive) process

More generally, hypothesis generation can be used to make better use of existing knowledge, to make novel (intelligent) use of new facts/relations and to improve the nature of intelligent interaction in HCI[52], Anomaly detection[50], Diagnostic analysis of problems, Case based explanation, and Natural language understanding etc. (FIGURE 1.2)

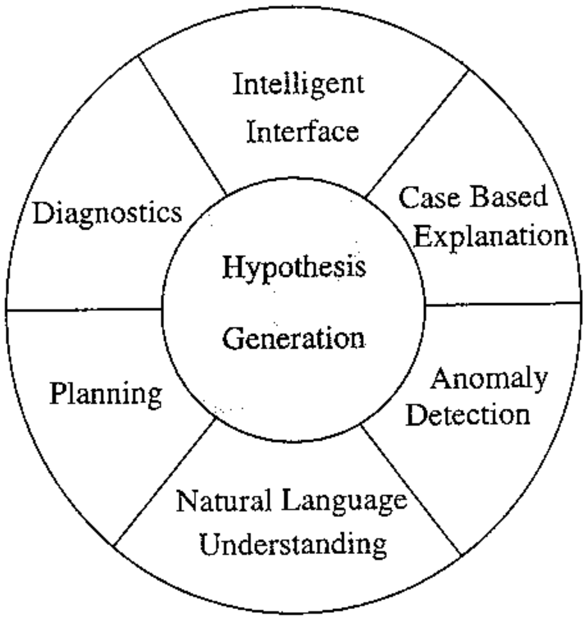


FIGURE 1.2 Potential Use of hypothesis generation

The achievement in developing and applying this hypothesis generation model has implications for four of the fundamental research areas in artificial intelligence.

Case Based Explainer theoretically solves the problem of building new explanations from old ones and relies on having explanations available in memory[145]. One of the more serious problems remaining is the capability to deal with unfamiliar situations. The capability to generate an hypothesis is most appreciated when, in problem solving, there is no existing solution, which can be adapted to explain the evidences. These kinds of problems will arise when knowledge gaps between computer systems and the real world exist or dynamic changes are occurring in complexity. My framework is partially based on the Case Based Interaction principle[35][49] to perform the new hypothesis generation, and, thereby, overcome the limitations encountered in Explanation Generation mechanisms.

The key features of the research that distinguishes it from previous contributions may be summarized as:

- An unifying hypothesis space representation can be developed to store and process both declarative and procedural knowledge. Two types of relations have been identified in the hypothesis space. They are explanatory relations and executable relations.
- An abductive inference engine, based on the theory of conceptual graphs[151], is original. The generalized set covering model[132] is used to reduce the combinatorial explosion of abductive inference, which is a central problem in abductive reasoning.
- An anomaly detection model, which is mutually complementary to and integrated with the hypothesis generation model, can be used to discover not only anomalous data, but also anomalous relations between normal data. This is a great advantage over contemporary anomaly detection techniques.
- The principle of case based interaction provides a sophisticated interface for cooperative problem solving, which refers to synergistic utilization of knowledge from users and computer systems. Under the guidance of this interaction mode, the users are active agents empowered by the system knowledge, and the systems get their auxiliary(extended) information/knowledge sources from the user.

Finally, it should be recognized that the experimental framework has been validated in complex commercial situations, specifically fraud detection in the insurance field. The author has considerable confidence in the theoretical and practical utility of the new constructs.

CHAPTER 2

Background to the Research

2.1 Current Problems: The Need for MINTS

Firms today are experiencing rapidly changing markets, massive technological change and an increasingly global and competitive climate with problems in access to skilled labour, capital, land and knowledge. The timing of business decisions, sourcing of expertise requirements [70] and prevention/detection of fraudulent activity are now critical for business prosperity.

Despite the fact that companies are engaged in a dynamic, and continuous struggle with their competitors for market share, customers, profits, and capital, few companies study their competitors as closely as they study internal manufacturing variances or new product proposals. The current information systems of modern corporations are primarily designed to monitor internal operations, to facilitate management control and to manage assets.

More recently, however, business strategist have recognised the value of scanning their working environment, tracking the activities of their competitors, and seeking competitive intelligence.The rational for such interest in strategic intelligence, lies in;

1. Contemporary intelligence systems rely far more on overt, public sources than on covert espionage missions;
2. The public availability of useful information about companies[158];
3. Growing appreciation of the role of information in business planning.

A common response to the increased need for business intelligence has been to create formal intelligence units for monitoring and interpreting information from different sectors of the environment. In fact, there are very few, if any, large corporations that do not have a unit devoted full time to the tasks of acquiring, interpreting, and circulating information internally on their external environments. Therefore, the management intelligence systems required by competing organizations in government and business will need continual review and upgrading.

Despite their presence, such units have often proved to be singularly ineffective. There are a number of problems that impede the effectiveness of formal intelligence units in a large corporation. Simply creating a formal intelligence unit is rarely an effective means to meet the diverse needs of the total organisation. What is required is that an organization, be culturally sensitive to its environment and to the social impact of its activities, and thus commit itself to establishing an holistic approach to the development of corporate intelligence. Clearly, intelligence is the primary resource that will help management improve product and service, and will lead to improved productivity in competitive environments.

More and more people have accepted the requirement for business intelligence. As evidence of this, many of the Fortune 500 companies have already established an intelligence department and begun to build databases dedicated to competitor (market) intelligence [61].

2.2 The Definition of MINTS

Management Intelligence Systems (MINTSs) are computer systems that were developed in response to corporate planning (innovation) requirements and are intended by the organization they serve to scan the organization's environment and monitor its activities, so that management can better assess its opportunities for competitive advantage, with a view to enhancing the position of the company vis-a-vis its competitors and to managing loyalty marketing initiatives. Better management control and loss prevention (e.g. fraud prevention) may also result.

Manfred Kochen who first proposed the concept of MINTS in 1989 wrote, when mentioning the general requirement for MINTS:

> to help the firm it serves to clarify its map or image of the firm's environment, to clarify the concept of "position' and interest in that environment, and to discriminate between positions and interests it values highly and ones it values negatively[80].

Significantly in his writing Kochen treated MINTS as intelligent decision support, noting:

> MINTS is an executive support system, i.e., a decision support system for high-level managers responsible for strategic decision-making and planning, supplying them with intelligence rather than information, and based on advanced information technologies, notably artificial intelligence.

When discussing the purpose of MINTS, Kochen commented:

> The purpose of such man-machine MINTS is to support professional strategists, planners, and researchers as would a good semi-automated research assistant, enabling the strategists to produce better plans, more quickly and at low cost.

One of the features of these applications is that they support managers in commercial or profit-seeking organizations. Therefore, the Management Intelligence Systems required by competing organizations in government and business should be "state of the art"!

2.3 The Spectrum of Information Systems

Information system is defined here as a computer-based system capable of serving organizational purposes [161]. Current information systems can be divided into three groups according to their functions and knowledge level: Management Information Systems, Management Support Systems which consists of Decision Support Systems and Executive Information Systems, and Strategic Information Systems.

2.3.1 Management Information System (MIS)

Management Information System (MIS) lacks unanimous agreement on definition, but may be broadly defined as an integrated, computer-based, user-oriented system that provides information for support of operations and decision-making functions [1]. A contemporary requirement of MIS is a corporate database with common data shared by users and management according to needs and priorities. The principal objectives of MIS at the operational level include summarising key financial results, performance reports and support for management by exception. Commercially sensitive records, such as payroll, customer orders, purchase orders, production and marketing costs, provide the sources for this management information.

Existing management information systems (MIS) directly support lower levels of management, particularly in decisions on:

- reduction in operating level costs;
- planning and scheduling of manufacturing/inventory;
- customer service levels; and
- enhanced corporate communications.

MIS has been more effective in management control than in the business planning area. Control functions tend to be more structured, deterministic, logical, and, therefore, more amenable to automation. MIS has had less impact on middle management, other than in resource allocation and in medium term budgetary planning. Top management typically relies on reports from middle management for control and MIS has had little impact on strategic planning processes.

However, in the past decade, a greater understanding is apparent of Management Information Systems amongst business managers. Recognition that MIS is a key tool for future online access by management to all corporate data is consistent with current management philosophies that emphasise corporate communications. New executive roles (e.g. in change management) are also leading to expanded MIS functions using online decision analysis tools.

2.3.2 Management Support System (MSS)

The role of Management Support Systems is to address those inherently unstructured problem areas, that by their nature make it difficult for a classical transaction processing system to address effectively. A number of new terms have been introduced to identify specific user classes and to create desired differentiations. Some of the terms in the literature that are related, or overlap are:

- Executive Information Support Systems
- Executive Support Systems
- Professional Support Systems
- Management Control Systems
- Operational Support Systems
- End User Computing Systems

Decision Support System (DSS) is a computer-based information system developed specifically to assist the professional classes to optimize their decisions. DSS is a computer-based analytical system. It provides relevant information on demand, possibly based on incomplete (subjective) data, and may require access to widely dispersed information sources.

Executive Information System (EIS) can be defined in its broadest sense as one that deals with all the information that helps an executive make strategic and competitive decisions, keeps track of the overall business and its functional units, and minimises the time spent on routine tasks performed by an executive [139][155]. EIS is an information system developed to supply senior executives with appropriate information on demand, in response to unstructured inquiries. It usually consists of a summary database supported by high level access methods. EIS answers data inquiry and retrieval requests, with some report generation and graphical display capability.

Within the past decade, the use of information systems by managers and professionals has grown exponentially, This has been stimulated by the increasing supply of software tools for end-user computation, and by external database services. The benefits of an effectively implemented MSS are to be found in the improved performance of the organization.

MSS is useful to departmental heads and to middle-level management in their focus on tactical planning and policy implementation. For example, sales analysis, production scheduling, and budget allocation involve a time horizon measured in weeks or months rather than years. The tasks of summarising and distributing information can be automated to a significant extent by information technology.

The traditional role of middle management in the collation, sifting and distribution of information across organisation is increasingly being performed by MSS. In summary, MSS is perceived as support activities to better business integration. It mechanizes operations for better efficiency, control, and effectiveness, but it does not in itself increase corporate profitability. It is simply used to provide users with sufficient, reliable information to manage the total business.

2.3.3 Strategic Information System (SIS)

Strategic information system (SIS) is one of the most important issues facing management today. Although several attempts have been made at defining an SIS, a useful short definition of an SIS is an IT application which directly assists the firm in achieving its corporate strategy[133].

More generally, strategic Information System (SIS) is a computer system that reflects business strategies, and is developed in response to a corporate business initiative [13][15][63]. SIS devotes information services to strategic business opportunities. The desired payoff in terms of an improvement in the organisation's products and business operation can then be monitored using the MIS.

Strategic information system is becoming an integral and necessary part of any business organization and is increasingly directed towards securing competitive advantage through repositioning or market stratification [13]. Expected benefits of SIS include improved market penetration, future earnings growth and customer loyalty. SIS may identify the need for new products, market diversification and new ways of doing business (e. g. telemarketing). An important goal of SIS applications is high quality decision making, which often leads to some centralisation of top management functions due to the far reaching impact of changes in corporate strategy!

From a corporate point of view, successful SIS systems for achieving competitive advantage require that the business realign operational information systems to monitoring the effectiveness of future competitive strategies.

2.4 The Relationship between MINTS and Contemporary IS

A Management Intelligence System is different to contemporary information systems in three major respects; management levels, information sources, and operational functions.

2.4.1 Management Level

By the end of the 1960's, a theory of business computing and data processing had emerged, and was subsequently applied to management issues [161]. According to Anthony, the enterprise may be seen in terms of a trinity consisting of the three processes of strategic planning, management control, and operative control. FIGURE 2.1 indicates the classic management triangle which was described by Robert B. Anthony in 1965 [161].

Strategic information is future oriented, involving uncertainty and subjective data (e. g. estimates). It is useful in long-range policy planning, which is the task of senior management. Management information is useful to middle management and department heads, who focus on tactical planning and policy implementation. Operational information is the short-term, day-to-day information used in operating the business. It is usually highly structured and well-defined.

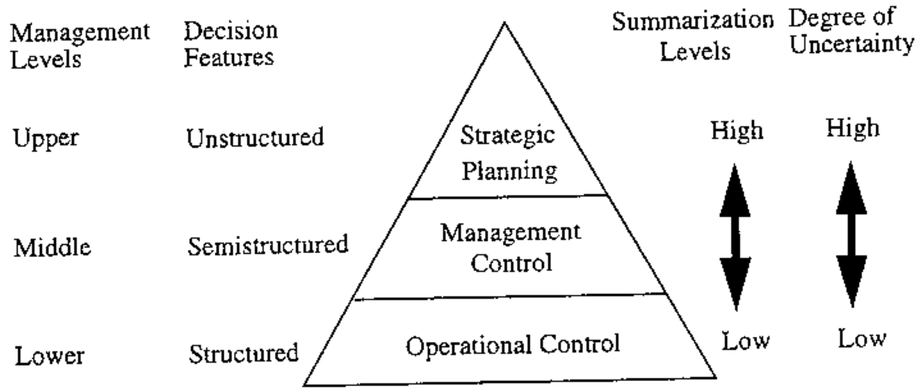


FIGURE 2.1 The Levels of Management

MIS, as defined, may be used in automation of the basic business processes of the organization. It offers great value to management for operational control and provides decision support in structured environments.

The primary function of MSS, by definition, is to provide end users with information for managers and professionals, closely related to their functional responsibilities. MSS works at the middle-level of management, and provides decision support in a semi-structured environment with limited uncertainty.

MINTS and SIS both work at the highest management level, typically unstructured decision support environments with high uncertainty. Unstructured decision making implies that a routine (methodology) for the decision process does not exist, that multiple decision phases may exist and novel reasoning strategies may be necessary. Methodologies developed in the mid eighties for design of such systems placed so much emphasis on the structure of the DSS, and the modelling of input requirements, that it became almost impossible to use such systems for ad-hoc decisions [103].

2.4.2 Differentiation in Information Sources

From a strategic perspective, management usually needs intelligence for two general purposes: environmental scanning and competitive analysis [140]. This requirement makes the MINT use not only internal databases, the primary information sources for MIS and MSS, but also the databases available for public retrieval. The common goal of information systems is to make information access easy, quick, inexpensive, and flexible for the users at different levels.

New technologies, such as high speed global networks and inexpensive online storage have combined to produce a sharp increase in the availability of on-line information. Knowledge workers have became increasingly interested in the activity of market researchers and corporate planners; namely, monitoring their environment and tracking the activities of their competitors.

It has been noted that management intelligence systems devote special attention to the public online information sources [80], and take advantage of text information in many applications. Access to online information sources in electronic form (e.g. World Wide Web) is now recognized as crucial by most professionals. However the volume of information demands advanced techniques for searching, selecting and managing data/information.

2.4.3 Differentiation in Function

MINTS has many extraordinary functions such as scanning the environment and monitoring external activities of interest. In a broad sense, a firm's environment includes the market, technology, the financial world, and organizational aspects (e.g. strategic alliances). It even includes changes in the firm's internal environment. The activities monitored by MINTS embrace competitors' strategic movements, and customers' behavior patterns.

The result of MINTS is an intelligence report. Intelligence differs from conventional reports (information) through the use of diagnostic capabilities to identify previously unknown relationships between items from disparate sources, and by means of interpretive skills, in analysing the meaning and significance of new, but apparently unrelated facts. Such intelligence may be of strategic or of tactical importance, but more analysis is typically required for higher-level, long-range strategic planning.

In combination with the original work by F. Kochen, and the brief reviews provided on information system development, the following functions have been identified as the principal requirements for MINTS design and development in fraud detection:

- to provide searching, collating and analytical capability leading to the definitive matching of ownership records of assets and property, where there is some degree of suspicion, with data provided;
- to recognise that some of the information may be obscure, incomplete, or otherwise presented to conceal true ownership or identity;
- to enable the investigation of such information, together with other data and knowledge owned by system, by annotation, sorting, collation, indexing, cross referencing, linking extraction into separate files with other relevant data, financial analysis spreadsheeting and output in an appropriate manner;
- to facilitate the interpretation of anomalous evidence, to identify how participants and companies interrelate, scheduling documents according to witnesses, to chronicles of events showing major relevant items such as dates of appointments, major transactions, flow charts; and
- to coordinate the contradictory information according to the reliability of the information source, which has a default value, and can be updated after coordination of processing.

We conclude that the major functions of MINTS is to bring together the data from a variety of sources, which have a variety of formats, into a single facility, having a common set of front-end enquiry and analysis procedures.

2.5 Opportunities for MINTS Development

Many advances in technology (e.g. statistical methods) have assisted in general problem solving, but none so profound as information technology (IT). As we approach the next century, systems which are easier to use and more flexible will bring IT into many more applications, such as strategic planning and business management. New opportunities are evident for the role of IT in business development through research into MINTS.

The new paradigm for proactive management substantially changes the scope for business problem solving. Furthermore, access to global networks promises vastly increased opportunities for remote collaboration and resource sharing. A fundamental problem confronting business management is how to discover relevant information from existing sources, such as documents, network services, and expertise of other users.

The vision that some day manager will write down a problem or a goal on an electronic medium sustains the search for a management intelligence system that will intelligently respond with a reasonable solution. Such a goal is basic to this research, even if its "full" realisation has not been possible at this time.

2.5.1 Development of Networks

Today's telecommunication networks provide the links that move massive volumes of data across continents and oceans in seconds. With satellite communication, for example, it makes no difference whether two or more databases on earth are close together or far apart. Access to a system or database is independent of its location. The development of telecommunications makes it possible to compress time and space.

There are many networks around the world providing services to academic and research users on a national and international basis. There are commercial networks for which users pay to have access on an individual basis, and there are also corporate networks that link the branches of large companies. Many of these networks are compatible and are interconnected with each other, so that the full range of network services can be used, in principle, to access any resource on any connected network. The largest collection of such compatible networks is the *INTERNET* [37] and is characterised by the fact that computing systems connected to it use the *Internet Protocol* (IP) to communicate with each other.

Today the *INTERNET* reaches nearly 1 million hosts in more than one hundred countries, with an estimated five million users[73]. *BITNET* is the largest non-*IP* network, based on IBM mainframe technology, and has extended to many locations that *INTERNET* does not yet reach, including much of Eastern Europe and the Middle East. Although *BITNET* is incompatible with Internet, computing systems with both an Internet and BITNET connection can act as gateways, allowing information from both networks to be exchanged. Other networks[136] include *DECnet Internet, NSFNET, AARNET, SPAN, JUNET and USENET* et al, and Commercial networks such as *BIX, CompuServe, Dialog, The Source* and *Telebase*.

2.5.2 Development of Network Resources

Widely accessible information resources may be assumed to be available to the business community (e.g. ASC data), including hundreds of gigabytes of software, documents, sounds, images, and reference catalogues. A number of autonomous agents (software)exist for remote database searches/browsing.

Because of the growth in network database retrieval (e.g. by scientists), the business community has begun to show great interest in the location, retrieval, and analysis of on-line information. In the past four years, a number of resource discovery tools, such as *Gopher*, *NetScape*, and *Mosaic* etc., have been introduced to help such users locate and retrieve relevant information available on the networks. Some useful network access tools are listed in Appendix 1.

2.5.3 Development of On-line Information Sources

Full text access to business information remains one of the fastest growing arenas for on-line information searching. *Dow Jones Text-search Services* and *DowQuest*, offered by Dow Jones News/Retrieval, represent impressive methods for searching business texts, when one wants access to a broad array of databases or a simple entry into current literature. Many commercial services offer a range of options for searching newspapers, trade journals, newswires, newsletters, and other types of electronic document.

Examples of current online information resources are listed in Appendix 2. The databases considered applicable to fraud detection in the insurance industry are listed in Appendix 3.

2.5.4 Hypothesis - the Missing Link

Contemporary tools and access mechanisms for information retrieval generally presume the existence of sufficient knowledge by the user (manager) of all facets of the query or investigation in progress. Hypothesis as the overarching construct for successful problem solving is typically ignored in investigative search.

The synthesis and use of appropriate knowledge structures for hypothesis generation, resolution and validation involve levels of complexity in knowledge representation and reasoning that have not easily been justified by former architects of DSS or EIS. These issues are now addressed.

2.6 Conceptual Structure for MINTS

In section 2.5 the growth of corporate access to online information sources was highlighted, together with the consequential opportunities for the development of wide-ranging management intelligence systems. A new paradigm for pro-active management was foreshadowed!

The requirements for any MINTS to accommodate high-level management hypotheses in pro-active management poses new research issues into the design and use of appropriate conceptual (knowledge) structures. A detailed review of knowledge representation schemes is now presented, embracing declarative and procedural knowledge. A conceptual graph based knowledge representation scheme is then proposed to represent hypotheses, containing both declarative and procedural knowledge.

2.6.1 Knowledge Representation

Newell defined knowledge as "whatever can be ascribed to an agent, such that its behaviour can be computed to the principle of rationality [109]". The artificial intelligence community have generally acknowledged that Knowledge Representation is the key issue in artificial intelligence research, because complex problem solving requires large amounts of knowledge. With an appropriate knowledge representation scheme, the complexity involved in manipulating knowledge can be reduced[92]. Knowledge representation can be summarised as a form of data structure used to organise the knowledge required for problem representation and solution[59]

Knowledge representation could also be described as the "glue" that binds artificial intelligence together[98]. The main purpose of any knowledge representation formalism is to organise the required information into a form that enables an artificial intelligence solution to be applied.

Various formalisms for representing knowledge have been developed over the last decades. They can be divided into two major categories[160]:

- Declarative knowledge representation,
- Procedural knowledge representation.

Rich[135] has identified some properties essential for a good knowledge representation system:

- Representational adequacy,
- Inferential adequacy,
- Inferential efficiency, and
- Acquisitional efficiency.

Woods [164], on the other hand, has introduced two key aspects of the problem of knowledge representation. As reported by Woods, they are:

- *expressive adequacy, which includes the distinctions of a representation it can make and the distinction it can leave unspecified to express partial knowledge.*
- *notational efficiency which deals with the actual shape and structure of the representation; as well as the impact this structure has on the system's operation.*

Not all knowledge representation schemes meet all of the above criteria. Most of them have varying degrees of compliance with the ideal scheme[95].

The knowledge required for a management intelligence system includes objects, processes and common sense knowledge, as well as the capability to accommodate goals, motivation, causality, time, actions, etc.

2.6.2 Declarative Knowledge Representation Formalisms

Most existing knowledge representation formalisms like *Frames* by Minsky[102], *Conceptual Dependency* by Schank[141] and *Semantic Networks* by Brachman[10] provide a sound framework for the representation of declarative knowledge.

Lukose [95] has provided a comprehensive survey of knowledge representation formalisms. There are six fundamentally different knowledge representation formalisms that could be utilized to represent declarative knowledge. They are:

- Predicate logic,
- Frames,
- Scripts,
- Semantic nets,
- Conceptual Dependency, and
- Conceptual graphs.

Conceptual graphs have been adopted as the core knowledge representation scheme in this thesis, and the theory of conceptual graph of Sowa [151] will be covered in 2.6.4. Conceptual graphs incorporate both the entity-relationship model and the semantic network scheme.

2.6.3 Procedural Knowledge Representation Formalisms

In procedural knowledge representation formalisms, the knowledge base is usually a collection of procedures. The modification of the knowledge base takes place when subroutine addition/subtraction/modification or access requirements change.

The artificial intelligence community uses the term "*ACTOR*", while the term "*OBJECT*" is used by the *Smalltalk* community[122], to describe a software engineering paradigm known as "anthropomorphic programming"[95]. Substantial work on the actor paradigm was conducted by Carl Hewitt in the early 1970's at MIT. Evolution towards a true Actor Model of computation comes from the development of Smalltalk by the learning research group at Xerox PARC, the *PLANNER* system and from the *ACT* family of languages by Hewitt.

PLANNER [71] is an artificial intelligence programming language addressing both representation and control information. In the *PLANNER*, the problem and their solutions can be stated in a modular, flexible style, similar to logic. Hewitt has described the logic of *PLANNER* as a combination of classical logic, intuitional logic and function required.

2.6.4 An Overview of the Conceptual Graph Formalism

The theory of conceptual graphs is largely a network representation formalism [151]. Two basic and important types of entities in this framework for representation are "*concept*" and "*conceptual relation*". A *concept* corresponds to a type of physical or abstract entity in the domain of knowledge that is represented as a graph. A *conceptual relation* represents the (semantic) links between concepts. A conceptual graph is therefore a directed, bipartite graph of connected concepts and conceptual relations.

2.6.4.1 Advantages of Using Conceptual Graph

Contemporary knowledge representation formalisms, such as Frame [102], Conceptual Dependency [141], *Semantic Network* [10], enable the representation of declarative knowledge. On the other hand, the procedural attachment in the *Frame*, *Script* [144]and Production Rules enable a way of representing procedural knowledge. The lack of formal mapping between operations in the formalisms and the corresponding functions in first order logic [95][156], however, is a problem. Conceptual graph formalism, however, provides such a mapping to first order logic. It is a semantically rich formalism for the study of modal logic paradigms[18]. Conceptual graphs have also been extensively used as an intermediate language for bridging between natural language and computer representation of logic equivalence [57] [152], as well as language generation from conceptual graphs[11].

Apart from representing declarative knowledge, conceptual graphs have been demonstrated as capable of representing procedural knowledge[95]. Dataflow graph[151] is a form of abstraction to represent operators. Hartley[67] has also used conceptual graphs as a high level natural programming language. Lui[94] has further demonstrated the ability to utilise conceptual graphs to represent causal rules for representing procedural knowledge. With the development of an Extendible Graph Processor, Garner and Tsui have demonstrated the ability to represent a control script as a type of procedural knowledge[58]. Furthermore, Garner and Lukose implemented intelligent control scripts[95] as a procedural paradigm with the *Extendible Graph Processor.*

In addition to the distinguished features of conceptual graphs that are mentioned by Sowa[151] and Clancey[23], conceptual graphs offer three distinct advantages over other knowledge representation formalisms[156]:

1. A subset of conceptual graphs can be directly mapped into first order logic. Sowa provided such a mapping[151]. Conceptual graphs can also act as an intermediate language (or notation), bridging between natural language statements and machine representation of logic equivalents.

There is active research on the decoding of natural language sentences into conceptual graphs as well as on language generation from conceptual graphs.

2. Control knowledge (or heuristics) can be embedded into a conceptual graphs, hence enabling the use of graphs as control structures in a knowledge-based system. Actor graphs are defined in Sowa[151], Hartley[67]. Sowa's actor [loc. cit.] graphs are similar to dataflow graphs in database systems, whereas Hartley's system [loc. cit.] uses conceptual graphs as a high level natural programming language for simulation and program development purposes.

3. Research results on modal logic reasoning[18] confirm that conceptual graphs serve as an excellent framework for developing sophisticated and elegant reasoning algorithms. Recursive modal logic resolution principles are defined and implemented for components of a graph instead of an entire graph.

The knowledge representation, processing, and reasoning potential of conceptual graphs has been realised with the development of the Extendible Graph Processor. It was developed at Deakin University over the last seven years as a domain-independent, knowledge engineering tool for advanced knowledge engineering research.

2.6.4.2 Conceptual Graph for Representing Declarative Knowledge

In the conceptual graph formalism, *concepts* correspond to physical or abstract entities, while *conceptual relations* provide the semantic links between concepts within a particular domain of knowledge. A type of hierarchy table is maintained to represent the type-sub-type associations between the concepts. A concept can have many referents, and a conformity table is thus maintained to enforce the canonical constraints. The conformity relation relates type labels to individual markers.

There are eleven types of concepts that can be utilized to represent declarative knowledge, as described by J. Sowa[151], E. Tsui[156] and D. Lukose[95]. They are:

1. Generic concepts:

   these are concepts that do not correspond to any particular instance of the knowledge being represented;

2. Individual concept:

   these are concepts that correspond to a particular (unique) instance of a modelled entity;

3. Empty set concept:

   there does not exist any instance of the concept;

4. Generic set concept:

   specifies that there exists an un-identified set of elements as the referent to the concepts;

   generic set is denoted by "{*}";

5. Individual set concepts:

   these concepts correspond to having a set of individuals as referents;

6. Partially specified set concept:

   these are concepts resulting from the union of an individual set concept and a generic concept;

7. Distributive set concept:

   these concepts specify that the interpretation of the concept has to be repeated for each element in the set;

8. Respective set concept:

   this concept requires that the interpretation of the concept has to be repeated for each pair of corresponding referents;

9. Nested graph concept:

   these concepts have a graph as a referent;

   these concepts are interpreted with respect to the referent graph;

10. Nested sets of graphs:

these concepts have a set of graphs as a referent;

11. Variable concept:

    these concepts specify that the referent is a variable to be instantiated later; the variables are denoted by "x" where x is a symbolic tag;

12. Rule package concept:

    these concepts may have a rule package as a referent representing strategic knowledge.

Conceptual graph formalism also provides us with another class of structure. J. Sowa, E. Tsui and D. Lukose define this class of structure as an abstraction. The five types of abstraction for representing declarative knowledge are:

1. Type definition:

   all concept type labels are denoted by a canonical graph which consists of the necessary and essential properties of the concept;

2. Relation definition:

   all relation type labels are also defined by canonical graphs representing all necessary and essential properties of the relation;

3. Schema:

   these are canonical graphs that represent the occasional and/or accidental properties of a concept type label;

   these canonical graphs show the typical ways in which a concept may be used;

4. Composite Individuals:

   this is the type definition graph (i.e., canonical graph) with the generic concept instantiated;

5. Prototype:

   these are canonical graphs that show a typical instance of use of a concept.

2.6.4.3 Conceptual Graph for Representing Procedural Knowledge

As summarised by D. Lukose, there are four methods to represent procedural knowledge utilising conceptual graphs. A brief overview of four formalisms is described below:

1. Dataflow Graph[151]:

   The actor nodes are attached to the conceptual graph to form a dataflow graph. *Control Marks* on the graphs are used to trigger the actors and computer referents for generic concepts. This dataflow graph formalism resembles Petri Net [121]with two kinds of token: *assertion marks* are propagated forward like the token in Petri Nets, but *request marks* are propagated backwards.

2. Actor Type[67]:

   R. Hartley has attempted to elevate procedural knowledge to the same level as declarative knowledge. He defines actor type in the same way as concept type and relation type, and introduced a set of actor inputs and outputs. Hartley's extension to conceptual graphs allow actors to be much like "active concepts", which accept states as preconditions, and events as trigger. An assertion mechanism is utilized to activate the actors which are used to express causality, involving states and events, and inferences, involving propositions.

3. Conceptual Rules[94]:

   D. Lui on the other hand has extended the production rule formalism with conceptual graphs. He has implemented the Rule Acquisition System for Conceptual structures, that encodes causal rules.

4. Actor Graph[95]:

   The actor graph incorporates the declarative knowledge representation of conceptual graphs, the procedural knowledge representation of *intelligent Control Scripts*, and the object-oriented properties of the *active Agent Paradigm*. Executing an actor graph involves sending an appropriate message to the *actor* component of the actor graph. The main methods in actor graphs are designed to initially check whether the pre-conditions are satisfactory.

2.6.5 Executable Knowledge Structure

Knowledge re-organisation is a process of re-arranging knowledge structures within a knowledge base. Schank has identified memory models that can automatically reorganise their knowledge structures as dynamic memory.

There has been active development of executable knowledge structures in recent years. Both artificial intelligence and software engineering practitioners have developed the executable transformation schema, as summarised by D. Lukose[95].

2.6.5.1 Control Script and Intelligent Control Script

Garner and Tsui have introduced the term *Control Script*, which is used to specify a skeletal set of Canonical Graph Processor operation required for the implementation of a particular knowledge structure[58]. The Canonical Graph Processor system is an interactive knowledge engineering tool that is utilized by knowledge engineers to perform conceptual graph processing. *Control Scripts* can be utilized to reorganise knowledge structures, even though they suffer from a series of fundamental limitations, which prevent them from being used for any serious knowledge engineering applications. The main limitations are:

1. static control script,
2. single script language,
3. no control structure, and
4. no variable binding.

To overcome these limitations, D. Lukose has made a number of extensions and realised the implementation of intelligent control scripts. In the intelligent control script, a *Control Transfer Mechanism* was introduced to enable the use of multiple script languages in writing a particular script. Secondly, the control structures are utilized to incorporate a high level control structure to enable interaction with users. Thirdly, the concept of a *self-*

organising script is introduced to enable re-organisation of relevant sequences of script at run time, usually based on the run-time state information.

2.6.5.2 Active Agent

The artificial intelligence community uses the term *ACTOR* while the term *OBJECT* is used by the Smalltalk[1] community to describe a software engineering paradigm known as anthropomorphic programming [55]. An Actor is a small processor defined solely by its behaviours, and characterised by its response to messages. The behaviour of an Actor System highlights its capabilities as well as its level of implementation. Various definitions of Actor are encountered in the literature. A few of the definitions listed below outline the functions envisaged by various researchers:

- a *process* that responds to messages by performing some service and then generating messages that it passes to other actors[151],
- a *capsule* of knowledge with behavior including self invocation, reproduction, self introduction and communication[106].
- an *active agent* which plays a role on cue according to a script[72].

Actors endowed with the properties of procedures and data since they perform computations and save local state information. Communication between actors is through *message passing* which is a form of indirect procedure call. When an actor receives a message, it will first determine whether it recognises the message for response purposes. If it does, the associated script, method or procedure is evaluated, and a response is relayed back to the sender of the original request.

In an actor model of computation, actors are divided into two major categories: classes and instances. A *"class"* is analogous to *"type"* in a procedural language. An *"instance"* is an actor that is not a *"class"*. The method and structure of an *"instance"* is determined by its *"class"*. Most actor systems support two types of variables: class variable and instance variables. *"Class variables"* are used to hold information shared by all instances of the

1. Smalltalk is an early example of an object oriented language.

"class", while *"instance variable"* contains information specific to a particular instance.

Inheritance enables specialisation. Specialisation uses class inheritance to elicit information. Actors may be created dynamically for various functions without the need of storing an impossibly large number. Polymorphism extends downwards in the inheritance network, because subclasses inherit protocols from parent actors. The conclusions from D. Lukose research into the characteristics of an actor are as follows:

1. the maintenance of a local body of knowledge;
2. it can provide operations which allow other actors to interrogate and/or update its local body of knowledge;
3. it owns its own local body of knowledge;
4. it decides if and when other actors can access it; and
5. it decides when other actors should know of its existence.

2.6.5.3 Problem Map

A problem map is made up of a number of actor graphs organised in a certain sequence. The sequence of the actor graphs is determined by pre-condition and post-condition of each actor graph. Similar to the type definition, schema, prototype and other forms of abstraction defined by Sowa[151], problem map is a new type of abstraction:

1. Problem map is an executable knowledge structure;
2. Problem map is a highly nested conceptual graph;
3. Sequence of messages to be sent to execute the problem map is stored in a message list;
4. Information about similar/identical concepts in the problem map is stored in a binding list;
5. Each problem map has a set of initial states under which it could be executed; and
6. When a problem map successfully completes execution, it will produce

a final state.

This review of progress during the past decade in knowledge representation has identified a source of rich abstractions and analogs for the design of conceptual structures for management intelligence requirements. A better understanding of user missions (goal) in management environments was also essential, however, to proceed further in the specification of generic MINTS reasoning capabilities and explanation requirements. The results of this study of typical MINTS application environments are now summarised:

2.7 Generic Requirements for the MINTS Paradigm

The justification for a new MINTS paradigm rests largely on three premises:

1. Complexity of contemporary business problems due to global competition, compliance with laws of evidence and the fragmentation (distribution) of information sources;
1. Emphasis on proactive management, including adaptive planning requirements in risk management; and
2. maturity of some knowledge base technologies in meeting requirements for cognitive models and diverse reasoning styles.

In the following section (2.7), four key requirements are identified in elaborating criteria of success for the MINTS paradigm.

2.7.1 Detecting Anomalies

A major opportunity for a generic MINTS is in systems integration, which involves connection to massive databases. For the user of MINTS, it would involve workstations which have global access to other professionals and are connectable to an enormous diversity of information data-banks. Global connection makes it possible to scan environmental issues and to monitor competitors' activities.

To deal with the massive explosion of data, one effective way is to identify anomalous phenomena for further investigation. The anomaly can be catalogued into two groups, anomalous values and anomalous relations, as revealed in this research.

The existence of anomalous values has been recognised for a long time, and a lot of researchers have sought to explain individual exceptions since 1755 [3][84][93][97]. In such work, there was seldom any consideration of the existence of anomalous relations, however, between data, yet there is ample evidence that anomalous relations often exist in behavioural situations with consequential implications for human conduct. The search for anomalous relations among data, or rarely encountered patterns of behaviour is a difficult problem for management.

Much of the existing work on anomaly detection is highly intuitive and takes no account of the nature of the working environment and the need for new hypotheses. For example, when concerned with an apparent anomaly in a set of independent data, it is natural and appealing to look for outliers using scatter diagrams or regression analysis. However, qualitative behaviours cannot readily be examined on a statistical basis, and the use of artificial intelligence to define appropriate reasoning processes has a great deal of appeal.

2.7.2 Generating a Suitable Hypothesis

In management intelligence systems, the most important function is carried out by an hypothesis generation model, typically linked to case histories.

Other than in game theory or in the exploitation of deterministic patterns of prior behaviour (e.g. case based reasoning), hypothesis generation has attracted singularly little research by the artificial intelligence community.

In our research, hypothesis generation is viewed essentially as the process of re-explaining observed phenomena, and resolving conflict among the possible explanations. This description derives from the notions of human cognition, where human beings are always attempting to explain observed behaviour. When two explanations conflict, creative thinking may result. For example, in renaissance physics, the behaviour of light could be partially explained by postulating that it was a stream of particles, and partially, by considering it as a form of electromagnetic radiation. The conflict energised a great deal of creative research into physics, and it is the authors' opinion that new hypotheses about modern physics were triggered by the process of data explanation and conflict resolution.

2.7.3 From Information to Understanding

When seeking to provide a competitive perspective to the company's strategic planning process, managers often discover that their knowledge of competitors is incomplete, widely scattered throughout the corporation, and generally not coordinated. What is perhaps even more frustrating is that the available internal assessments and opinions about competitors are frequently in conflict, unsubstantiated by documented facts, and often based on assumptions and intuitive hunches that are partially right, unrelated in context, and often out of date [139].

A basic, but often misunderstood, reality is that intelligence is not equal to information. Information is the raw material of the intelligence process. It is contextualised data derived from every possible information source, such as financial statements, trade show gossip, union newsletters, marketplace rumours, product brochures, executive speeches, and so on. Such pieces of competitor information flow by in a constant stream, maybe true or false, relevant or irrelevant, confirmed or unconfirmed, positive or negative, deceptive or insightful. In its undigested state, these voluminous items of competitor information, most of which is publicly available, may be vaguely interesting and occasionally intriguing, but however glittering, it is essentially an unusable and potentially dangerous resource [140].

Intelligence is produced by an analytical process that transforms the disorganised, confused, and sometimes contradictory stream of competitor information into relevant, accurate, and usable strategic knowledge about competitor's position, performance, capabilities, and intensions. Intelligence is the product resulting from the collection, evaluation, integration and interpretation of all available information, which concerns one or more aspects of competition, or of their areas of operations, and which is immediately or potentially significant to strategic planning.

The most important phase of the intelligence process is to transfer the raw information (knowledge) into understanding. Before the collected information can be upgraded to the category of competitor intelligence, every relevant piece of collected information must be critically assessed and then fitted into a large, more meaningful unity. The nature of the intelligence work at this stage is the intellectual activity of sifting diverse, often conflicting strands of competitor information to find the meaningful pattern within the stream of available data.

Transformation from information to intelligence requires elaboration of the link between knowledge and understanding. To deal with such a task, it is natural to categorise those knowledge structures suited to complex, unstructured problems.

2.7.4 Human Computer Interaction

Since the user of MINTS is, in most situations, expert in the domain, strategic pro-active management requires the creation of an environment conducive to the problem solving process. The key requirement of MINTS is to provide a cooperative environment between the users and the computer system to recognise and deal with the new situations. Human beings are good at using common sense, rule of thumb and conjecture, whereas computers are effective and efficient in assisting with problems requiring repetitive analysis and involving a large amount of data. Cooperative problem solving enables the strengths of both partners to be exploited to the full!

The problem of designing a Human-Computer Interface to mediate between the users, who require an understanding of the problem, and the computer-based information resources, poses a number of research issues. The intermediary and information resources together constitute the *Intelligence Provision Mechanism*. At present, such systems require a human intermediary. The desired human-computer interaction of *MINTS* will need to perform at least some of the functions which human intermediaries perform; such an interaction would perforce be intelligent.

The typical situation in management intelligence systems, and indeed in many decision support systems, is that the users are unable to specify the requirements for intelligence or information needed for managing a problem situation. The task of such an interface thus requires:

- adaptive model of dialogue control
- information selection for improved user understanding of the problem
- associated explanation capabilities
- anomaly detection and hypothesis generation in mediated learning

2.8 Summary

Justification for a new paradigm for MINTS has focussed on the complexity, regulatory and distributed information issues that are seen to pose insuperable difficulties for managers, unless a convergence is achieved between distributed problem knowledge and understanding. the competitive advantages of MINTS have also raised expectations of their capabilities (e.g. data mining). The maturity of contemporary research in knowledge based systems has similarly generated expectations regarding the new management opportunities for problem diagnosis and proactive planning. However, the extent to which current technologies can deliver the promised benefits is itself an issue deserving objective research, as there are patently few methodologies of generic value in management intelligence. Much of the current research is fragmented and unfocused in relation to the intelligence mission.

In chapter 3, a novel and original framework for MINTS is elaborated, and appropriate criteria of success are developed further.

CHAPTER 3

Framework for Management Intelligence

3.1 Management Intelligence Systems: Business View

F. Kochen described how a *Management Intelligence System* might look to a system developer and to a user (as shown in FIGURE 3.1). Raw data flows into the scanning box of the system in response to environmental changes. Two basic types of analysis are shown in box 1. One is monitoring indicators and another is searching for novel patterns. The indicator monitoring is applied to the incoming data stream, and the pattern searching requires comparing and correlating incoming data with what has been accumulated, which is no longer raw data. Hence, it must be screened on the basis of estimated reliability, utility, precision, clarity and novelty.

The screening function can be partially automated with the help of an expert system if criteria for data evaluation can be specified and the judgements of experts can be expressed in the form of rules. In current business practice, the functions in box 1 are generally performed by human experts, whose capabilities for dealing with vast and diverse data and knowledge streams are limited.

Monitoring well-defined indicators is easily automated, but the search for novel or rarely encountered patterns is a challenge. The basic idea for discovering new patterns is to screen out forms of behaviour that fail to correspond to any known patterns or are unusual variations of known patterns. The human expert may be very good at noticing the unusual or unfamiliar, and the computer, in symbiotic partnership with human judgement, could display data in various ways — using methods of *Exploratory Visualisation* — to enhance this ability. Moreover, much of the data in intelligence analysis is qualitative, in the form of unformatted text and graphics, and this would have to be transformed into a canonical representation, perhaps one resembling the form of influence graphs [80].

The most important function in the architecture of a MINTS is represented by box 2, in which the intelligence analysis process really begins. It starts in response to an alert or a stimulus that motivates hypothesis formation or generation. This motivation usually stems from external data. But it could also stem from the reflections or meditations of a human analyst.

At stage 3 (box 3) the analyst seeks to mine knowledge from the data using detected anomalies as cues for possible questions. In seeking to automate this process it has been noted [46] that domain-independent algorithms for asking good questions are particularly difficult to construct. The more specialised the domain the easier it is for an expert to ask profound questions! This problem is reflected in the need for domain restriction in box 2, so that non-trivial questions may be selected (box 4). The questions generated actually imply or define the strategy of the investigation to be conducted by the intelligence analyst.

In seeking answers to questions posed in box 4, conventional approaches depend on information retrieval strategies (IR) that may be informed by artificial intelligence (e.g. deductive database). Typically, such questions are transformed in box 5 into boolean combinations of search terms and would then be directed to one — or more on-line bibliographic databases. The latter consist of indexed references to documents that might contain relevant information. The role of AI or expert systems, at this first stage, may be solely to advise the investigator about which database(s) to use. The questioner (currently a human investigator, in future, perhaps an automated analysis assistant) is presented with a number of articles that are retrieved from such database, indexed with the specified terms, so that he

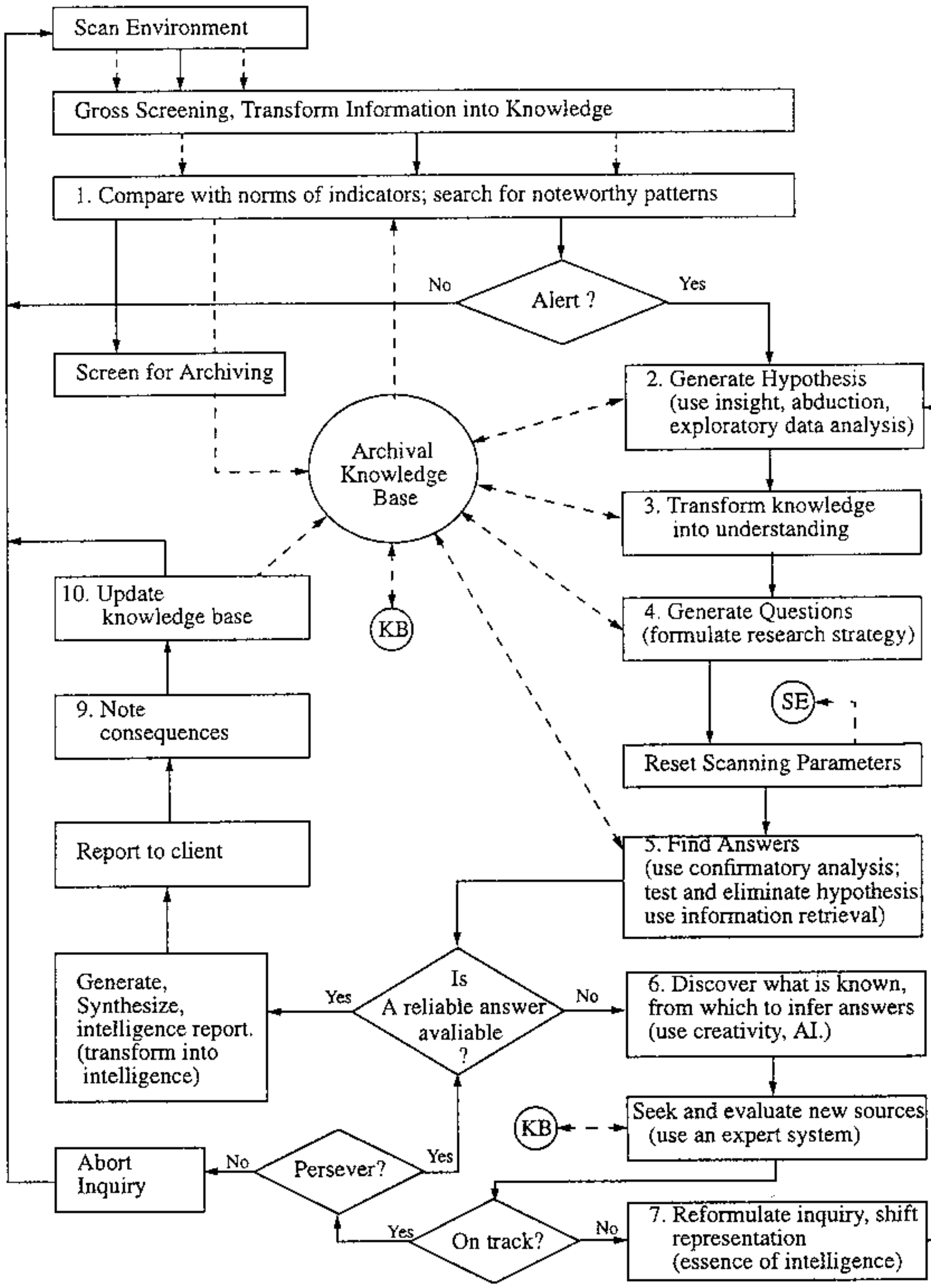


FIGURE 3.1 Architecture of a MINTS (Adapted from Kochen 1989)

can revise the search strategy. During the first run, only titles or abstracts may be retrieved. Finally, full bibliographic information would be printed out about articles matching the revised search specification. Manual searching of these articles would normally permit a useful intelligence assessment to be made.

Of great benefit are those situations where the questions may be posed in a query language that can automatically search a database or a knowledge base. For example, if he wishes to know for the past two years the number of motor vehicle theft claims in Kensington, a suburb of Sydney, there may be a database management system that can provide this data. In response to these opportunities for automation of the query, directory services are being implemented to identify the existing databases and their access paths.

Many database management systems (DBMS) are directly coupled with a *Statistical Analysis Package* (SAP), so that confirmatory statistical analysis can be done on a continual basis. Ideally, a DBMS and a SAP should also be integrated with simulation systems or modeling packages, such as an *Interactive Financial Programming Language* (IFPL), and also with tutorial systems, so that the investigator can get on-line guidance about which methods, programs or languages to use under different circumstances.

When it is not possible to obtain useful answers to the questions posed, artificial intelligence may be used (box 6) to retrieve units of knowledge from which better IR strategies may be formulated. For highly restricted domains of discourse, automatic question-answering algorithms for English-like questions have been developed. Should such discourse fail to provide the breakthrough needed, external knowledge may be sought off-line! Ultimately, if the enquiries prove to be fruitless, the process of investigation is abandoned, and replaced by a new one that is expected to do better. This is done in box 7 by zooming back to the general domain and selecting a different set of specialized domains or some other representation shift procedure.

The remaining two processes (box 9 and 10), complete a learning loop. Learning by the MINTS is necessary, if only to enable it to keep up with a changing environment. For it to improve, it must learn faster than required by the changing environment.

A MINTS can be regarded as a combination of several subsystems. Each subsystem has its own knowledge base, question askers, question answerers, and expert system (box 2 — 6). But there are also hypothesis generators at the system level. The subsystems are further organized into more specialized sub-subsystems, in which there is increasing expertise. It is the integration of all these subsystems that makes the production of intelligence possible.

A concluding statement in Kochen's work is worth considering:

> Many assumptions about human problem solving and decision making are again under challenge as researchers develop a paradigm to fit in with this perceived demand. Contemporary research into management systems reflect this research for new support paradigms.

3.2 Management Intelligence System: Artificial Intelligence Viewpoint

From Kochen's architecture, there are several stages in which an hypothesis is needed leading to the conclusion that hypothesis generation should play a key role in any management intelligence system. In our proposed structure for a management intelligence system, the central role of a knowledge based system, supported by an hypothesis generation paradigm, is the key to new search strategies and learning functions.

Thus, the structure depicted in FIGURE 3.2 can implement most functions of MINTS specified by Kochen and is based on an hypothesis generation model, *HG*, an hypothesis space, *HS*, and an anomaly detection model, *ADM*. The six (6) stages of the problem solving cycle of the process used by our management intelligence system are outline briefly below:

1. *ADM* scans the environment such as databases, and on-line information sources through the back-end interface.
2. When some anomalies are detected by *ADM*, the attributes will be sent to *HS*, and relevant components in the system will be activated.

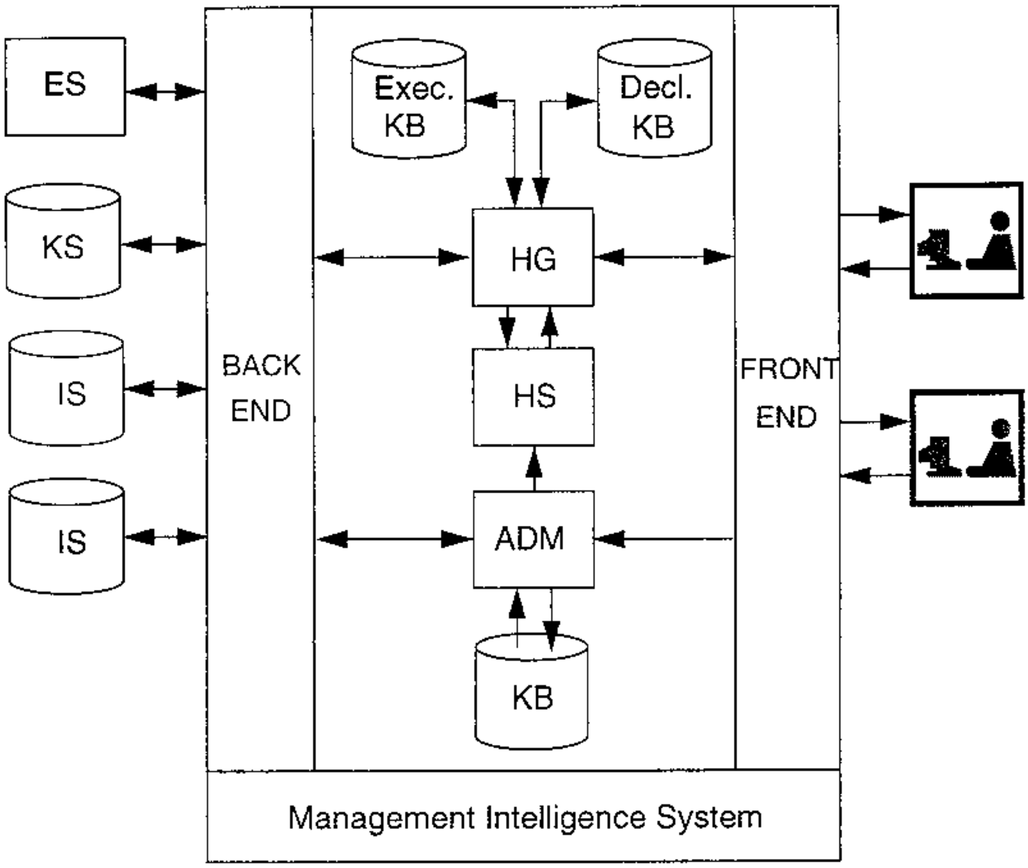


FIGURE 3.2 Structure of Management Intelligence System

3. Based on the active components of *HS*, *HG* will generate a candidate hypothesis abstract, which will be instantiated by domain knowledge. The hypothesis generation process also involves the evaluation process for hypothesis verification.

4. After the hypothesis is generated, it can be used to solve the problem by execution of the proposed hypothesis. One example of hypothesis execution is to explain anomalies detected. This explanation may be supported by information retrieved from various sources.

5. The process of hypothesis execution is a feedback control process. The information gathered from hypothesis execution will be used to test this hypothesis. The credibility of an hypothesis will often change in the pro-

cess of verification.

6. The final stage is to generate an intelligence report.

At any one of these stages, the user can interrupt systems process by assigning another scanning strategy, the system explanation may be overridden by attaching a new explanation, and a generated hypothesis may be manually verified or rejected (i.e. "fastpath" facility).

All the processes can be represented using the flow chart shown in FIGURE 3.3.

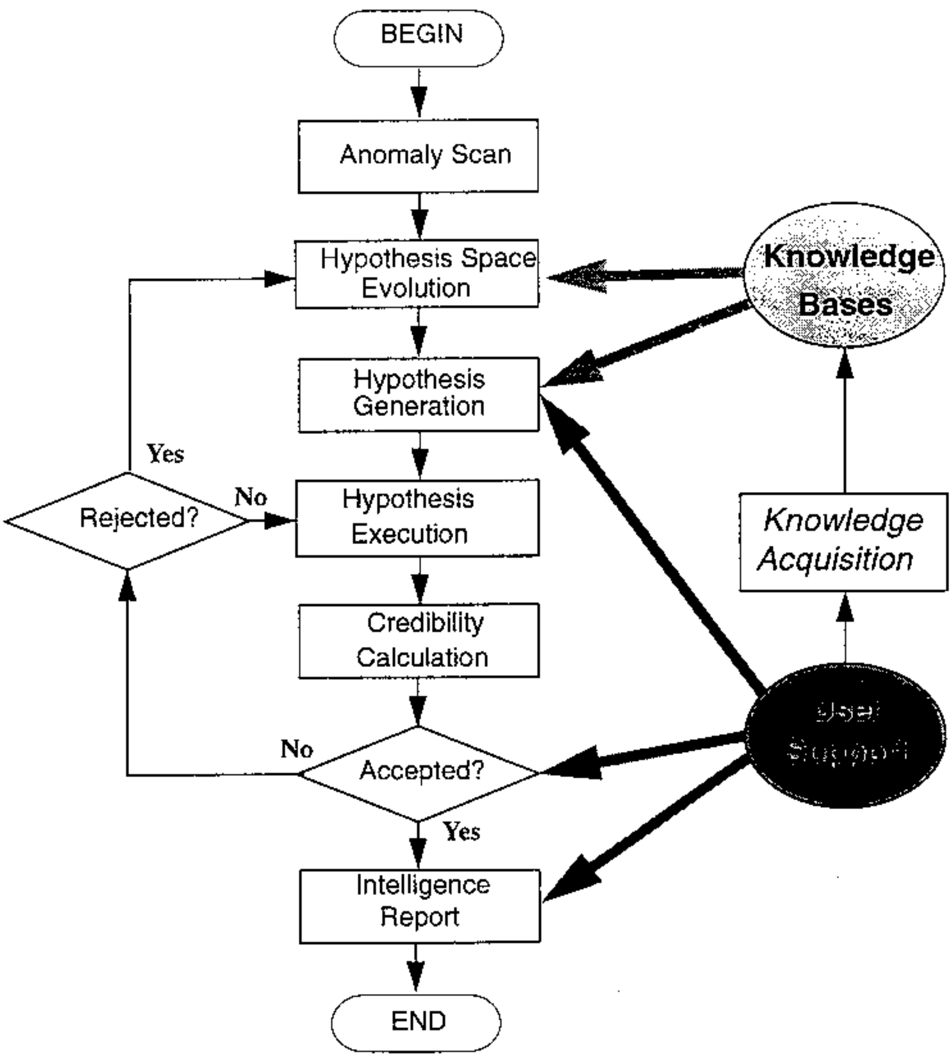


FIGURE 3.3 Problem Solving based on Hypothesis Generation

3.3 Hypothesis and its Representation

The knowledge required for a management intelligence system includes declarative and procedural knowledge, such as about objects, processes, as well as about goals, motivation, causality, time, actions, etc.

3.3.1 Limitations of Current Approaches

In order to overcome the limitation caused by treating the knowledge differently from the data/information, collaborative research conducted by University of Technology, Sydney and the CSIRO Division of Information Technology developed an unifying formalism called '*objects*', in which data, information and knowledge are all described by one formalism[28]. Their work focuses on three issues:

> First, the development of an unified framework for conceptual modelling in which the 'data', 'information' and 'knowledge' in the application can all be represented entirely in a single formalism. Second, the development of classes of constraints for knowledge that can protect the knowledge base effectively against the introduction of update anomalies. Third, the derivation of a single principle of normalisation (loc cit).

Furthermore, the distinction between declarative and procedural knowledge governs the flexibility of knowledge system's application in profound ways.

Since 1985, the *Knowledge Engineering Group* in *Deakin University* has built a number of knowledge engineering tools based on *Canonical Graph Model*. This project has progressed from the initial *Canonical Graph Processor* to an *Extendible Graph Processor*[156], and subsequently, included an *Actor Paradigm*[55] and *Problem Map*[95].

Even though *Canonical Graph Model* was an improvement in comparison to other models for knowledge engineering, it was unsuited to rapid prototyping or to general problem solving by domain experts wishing to build hybrid systems.

The Deakin design of a *Problem Map*, notably the executable feature of the actor mechanism, has successfully realised a knowledge acquisition process for complementary use of declarative and procedural knowledge in goal interpretation [95]. Strategic knowledge is a key goal of this knowledge acquisition process [65].

Our research into knowledge management and knowledge reusability has, however, identified significant limitations in use of the Problem Map design in management intelligence applications. The principal limitations in relation to our requirements are summarized below:

1. Problem map is a static knowledge structure.
   Dr. Lukose uses goal interpretation mechanism to explicate planning knowledge to construct problem maps[95]. Once the problem map is built up, it is fixed, and is difficult to change. Thus, the problem map lacks the ability to modify and maintain itself in dynamic environments.
2. There is a lack of knowledge sharing among different problem maps.
   Problem maps are stored separately, and there is no connection among them. Strategic/planning knowledge is not shared. Due to this limitation, a new problem map will be built from scratch whenever a new situation is encountered.
3. The explanation capability is inadequate.
   Description of the problem solving routes or the sequences of actor execution are the only explanation sources available. Explanations based on those sources are often unconvincing at strategic levels.

Removal of those limitations was necessary in the elaboration of a structure for representing hypotheses. A number of modifications and extensions have been made to the base Problem Map leading to significant improvements to the generic model:

1. Introducing *Hypothesis Space* as a *node-relation* structure to represent the expertise of the expert's domain. This knowledge structure will be used to generate a specific hypothesis by the hypothesis generation model.
2. Hypothesis space is a dynamic knowledge structure which will expand in the process of knowledge acquisition and will shrink in the process of knowledge re-organization (fusion).
3. All hypotheses are concealed in the hypothesis space, and an hypothesis will only become active when its triggers(evidences) are activated. The hypotheses are interconnected and overlap in the hypothesis space.
4. In the hypothesis generation process, a segment of hypothesis space is *projected* by the executable components (actors), thus producing an instantiated *hypothesis*.
5. An hypothesis can be executed if its pre-conditions (observed evidences) are satisfied. On completion of execution, an explanation will usually be produced.
6. From the topological viewpoint, an hypothesis is also a highly nested conceptual graph and is, thus, a complex abstraction.

3.3.2 Hypothesis Representation

An intuitive concept of hypothesis may be illustrated as follows: two attributes a_i and a_j are said to be related by a chain of qualitative connection in the simplest hypothesis shown in FIGURE 3.4. The qualitative connection included in this chain, when assumed to be active, would invoke the explanation of a causal relation (r_i) between a_i and a_j.

The hypothesis (H_i) is formally defined as follows:

$$H_i \equiv < ID_i, E_i, XP_i, KP_i >$$
$$KP_i \equiv < DK_i, ACTOR_i >$$

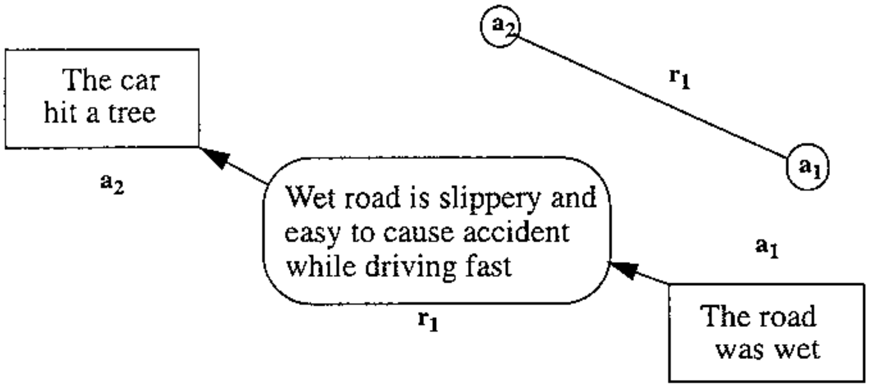


FIGURE 3.4 Simplest Hypothesis Structure

where i ∈ I and I is a set of positive integers;

> ID_i is the hypothesis' identification number;
>
> E_i is a set of nodes representing evidence or attributes;
>
> XP_i is a set of relations representing explanation sources;
>
> KP_i is executable knowledge (expressed in set form) which can be used to verify the anomalies;
>
> DK_i is domain knowledge (declarative knowledge packet);
>
> $ACTOR_i$ is a set of actors (executable knowledge packet).

A simplified definition of hypothesis can be expressed as:

$$H = (A, R)$$

For a given set of nodes A (that is attributes which represent observed or unobserved evidence, facts, and conclusions), and a set of relations R, an hypothesis, $H = (A, R)$, is that subgraph of hypothesis space HS, formed by connecting the separated nodes with relations, that obeys the following two constraints:

1. each node has at least one qualitative relation connected to another node;
2. there is no conflict among the relations in an hypothesis.

Hypothesis is a type of knowledge of unusual complexity and scope. The following context may be used for its application:

- Hypothesis is an executable knowledge structure;
- Hypothesis is a highly nested conceptual graph;
- The hypothesis contains actors connected in a certain sequence. Execution of these actors will cause the state of the hypothesis space to change. Hypothesis verification (or not) follows from the execution path that the evidence supports.
- Each actor in an hypothesis is activated by pre-conditions of the actor;
- The sequence of executing actors in the hypothesis is determined during hypothesis generation, and could be revised according to the current execution state;
- When an hypothesis successfully completes execution, it will produce a final state (goal).

3.3.3 Data Structure of Hypothesis

The hypothesis structure definition has been elaborated steadily during the course of this research to enable the computer to identify and utilise hypotheses to solve domain specific problems, Prolog data structures were selected, as the knowledge engineering workbench at Deakin is based on the Prolog Programming Language. The data structure needs to represent the name of the hypothesis, the conceptual graph representing the hypothesis, the concept binding list to propagate new information during the execution of the hypothesis, and some other details particular to our implementation.

A Prolog fact with eight (8) arity called hypoindex/8 is used to represent the definition of an Hypothesis. The data structure is shown below:

hypoindex (<hyponame>, <hypo cg>, <hypo cg id>, <main concept name>,
<main concept id>, <binding list>, <initial state cgs>, <hypo id>)

where:

<hyponame>	-	the name of the hypothesis;
<hypo cg >	-	the name of the conceptual graph that represents the hypothesis;
<hypo cg id>	-	the identifiers of the conceptual graph that represent the hypothesis;
<main concept name>	-	the name of main concept in conceptual graph;
<main concept id>	-	the identifier of the main concept in conceptual graph;
<binding list>	-	nested list containing the concept identifiers of identical concepts in conceptual graph;
<initial state cgs>	-	list of skeletal initial state graph identifiers representing the initial state required to activate the hypothesis
<hypo id>	-	the unique identifier representing the hypothesis

The following is an example of such a definition:

hypoindex(staged_accident, stage_a, 810460, hypo, 765483,

[810448, 810460, 810501], [810 541, 810194, 810 325], 810101)

In the hypothesis structure (FIGURE 3.5), *k-package* points to an executable knowledge packet, which is represented as a nested graph, and describes the knowledge base and procedures (actors) to apply the knowledge.

Evidences are connected by an *explanation package,* represented as a nested conceptual graph. A simplified structure for an hypothesis is shown in FIGURE 3.5.

3.4 Improving Explanation by Combining Planning Heuristics with Explanation Based Learning

The need for better explanations in knowledge based systems has been recognized for a long time. Domain knowledge is typically incomplete for many classes of problem, necessitating the use of heuristic knowledge in association with procedural knowledge. Being heuristic, they do not guarantee optimal solutions: in fact, they do not guarantee any solution at all. The best that can be said for a useful heuristic is that it offers solutions that are good enough most of the time.

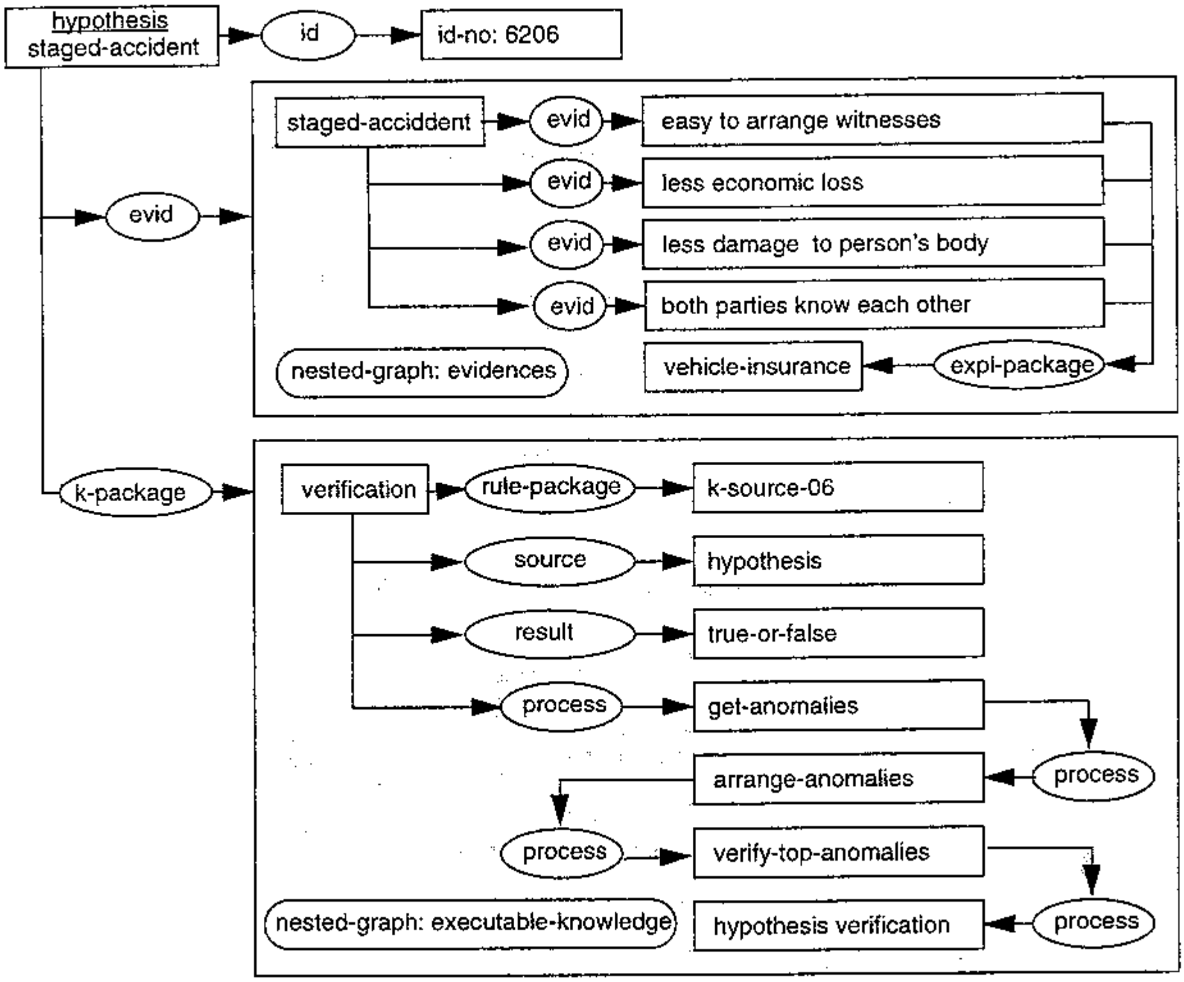


FIGURE 3.5 Simplified Structure of Hypothesis

For a knowledge based system to be trusted, it must be able to provide explanations on its reasoning, justification and conclusion. Good explanations can make an heuristic system more believable and can improve confidence in the system's advice. On the other hand, if a knowledge based system has been pushed beyond the limits of its expertise, an explanation facility can also make it obvious to the user, thus warning the user that the system's advice is based on tentative conclusions and may be erroneous.

Based on direct tracing of the rules fired in solving a problem, first generation systems produce explanations by paraphrasing the rules or execution traces into natural language using templates [146]. A major advantage of this approach lies in its simplicity: once the system is working, explanations may be generated fairly easily.

However, the limitations of this approach of "*paraphrasing-the-rules-to-explanation*" have been well documented [25]. Swartout and Moore, while working on the requirements for second generation expert systems[154], concluded that the first generation systems' failure to generate a good explanation lies in three limitations: the use of a representation (rules) that is too low-level, the failure to capture all the information needed for explanation, and an inability to distinguish the roles that different kinds of knowledge play. Fundamentally, all these limitations derive from the same origin.

On the other hand, case-based explainers, provide explanation from prior (known) explanations stored in memory. Case-based explanation ignores the causal information which may exist in a knowledge base, and depends on anomaly detection, case index, and case modification techniques to solve the problem[145]. The attraction of case-based explanation lies in the power of indexing and modification methods. But this method appears to ignore the effective guidance provided by tracing rules or causal relations in knowledge bases.

Explanation generation is a process of constructing a consistent narrative that relates the evidence to be explained and the conclusions that the user understands and accepts. First generation systems assumed that satisfactory stories could be produced with clever techniques for traversing, pruning, and translating the system's execution trace. But, as we have seen, this approach proved unsatisfactory.

The main aim, in developing explanation capabilities involving hypothesis structures, is to develop a knowledge representation which is suitable for the explanation generated. Generating a good explanation is a complex problem solving process requiring its own expertise.

The design advocated here involves separation of causal relations from explanation knowledge, while retaining use of causal relations for guidance of the case based explanation mechanism. Explanation requires its own body of knowledge, in addition to the knowledge used by the conventional knowledge based systems.

Many of the explanation failures of early knowledge based systems can be attributed to restrictions on structures in representing knowledge that is required to support explanation. Researchers now have a better understanding of what kinds of knowledge support good explanations. Second generation expert system architectures have been developed to represent that knowledge and to make it available for explanation[154]. The explanation generation is a problem-solving activity in its own right, worthy of its own problem-solving architecture.

Conceptual graphs, as used in our hypothesis structure, provide the ability to represent both declarative knowledge and procedural knowledge, and facilitate the combination of planning heuristics with explanation based learning. The limitations of explanations provided in first generation systems, however, can be resolved more efficiently when explanation based learning provides strategic knowledge.

Thus, to support better explanations, the knowledge structure needs to represent two general kinds of information[154]:

1. It must represent concepts, methods, facts and terms that are potentially familiar to the user and that underlie the system's actions. These are the starting points for an explanation. They are the points where the explanation connects with the user's current knowledge,
2. It must represent the inferential linkages between the starting points and the item to be explained. The linkages show the object to be explained relates back to things the user understands. The linkages provide an abstract "story line" for the explanation, based on strategic knowledge for representation of the inference strategy.

One of the promising features of our hypothesis structure is that the system is able to support feedback from the user about the suitability of its explanations. This capability is crucial for two reasons. Firstly, studies of human computer interactions show that experts and novices must negotiate the problem to be solved as well as agree on a solution that the novice understands and accepts. Secondly, the relevance and accuracy of user models cannot be guaranteed in practice. Thus, unless systems can compensate for incorrect or incomplete user models, the preferable explanation is difficult to generate for a wide range of skill levels.

Our proposed hypothesis structure provides support for better explanations through the use of high-level specifications to allow the representation of extensive knowledge of the domain, its principles and terminology. Procedural and heuristic knowledge are separated and can be represented at various levels of abstraction. A knowledge base at several levels increases the number of possible "starting points" for an explanation and enhances understandability.

3.5 Anomaly Detection Modeling

Anomaly detection is a vital step in indexing and generating hypotheses. In this section, we present an anomaly detection model, the potential value of which has been demonstrated in the domain of fraud detection[50] and by other authors, for market surveillance [63]. This model drives the hypothesis guided problem solving mechanism, which pays attention not only to anomalous data, but also to anomalous relations among data, and also involves abductive processes for positing virtual relations.

3.5.1 Introduction to Anomalies

What are anomalies and what is anomalous evidence? As defined by most researchers[3][97], an anomaly is a subjective, post-data manifestation. In observing a set of observations in some practical situation, an observation may stand out in relation to other observations, usually as an extreme value. In other words an anomaly is one that appears to be an exception to other members of the sample in which it occurs.

The existence of anomalous values has been recognized for a long time, and a lot of researchers have sought to explain individual exceptions since 1755 [3][84][93][97]. In such work, there was seldom any consideration of the existence of anomalous relations between data, yet there is ample evidence that anomalous relations often exist in behavioural situations, with consequential conclusions for the explanation of human conduct. Research by the author in conjunction with a major insurance company has demonstrated that the discovery of anomalous relations is extremely important in fraud detection, as discussed in Chapter 9 (Case study).

Much of the existing work on anomaly detection is highly intuitive and takes no account of the nature of the working environment and the need for new hypotheses. For example, when concerned with an apparent anomaly in a set of independent data, it is natural and appealing to look for outliers using scatter diagrams or regression analysis. However, qualitative behaviours cannot readily be examined on a statistical basis, and the use of artificial intelligence to define appropriate reasoning processes has a great deal of appeal. Story understanding programs, in particular, have been particularly effective in using expectation as the basis of anomaly detection. The *SWALE* project [145], for example, detects anomalies in the stories it reads and explains them by retrieving and revising old explanations. The new explanations are then stored in memory for future use.

3.5.2 The Basis for Modelling Anomaly Detection

An *Anomaly Detection Model*, as defined and reported in 1994[50], has been used to scan and judge insurance claims on the basis of:

1. Anomalous data (outliers) defined both statistically and using rules of classification. For example, in vehicle insurance claims, anomalous data could be such items as;
   - events (e.g. accident happened at midnight)
   - attributes (e.g. very large losses; aged vehicle)
2. Anomalous relationships, which are typically unsuspected at the time of the claim and are indicative of suspicious scenarios. For example, the vehicle involved in the accident has previous claims; two parties to the accident/claim are known to each other!

To deal with these two types of anomaly, the detection process is separated into two steps, each of them utilizing different reasoning processes. The first stage is statistics/rule based processing for the detection of anomalous data. The next stage is hypothesis guided processing for anomalous relationship detection. The strategy of the *Anomaly Detection Model* is to narrow the scope of the data, and then to thoroughly investigate possible anomalies according to the hypothesis. Occasionally, it will be necessary to perform a data mining search for the discovery of virtual relations (referenced in chapter nine).

Anomaly (data) detection is initially simplified to identify any data which appears to deviate considerably from other data in the samples. There are various methods of detection [3][97][84]. In order to overcome the limitation of pure statistical methods and to provide a flexible process, a rule-based strategy was found to be necessary and has been incorporated in the detection model for the insurance domain.

The results of this anomalous data detection are the claims with fraud scores over a defined threshold, together with the qualitative evidence provided (summarised). Usually the evidence available is not enough for a firm conclusion for fraud. The claims which should be investigated have, however, now been limited.

The next stage involves hypothesis guided detection, which attempts to establish new relationships in the anomalous data. The respective heuristics are activated by initial indications of anomalous behaviour based on evidence from:

- historical pattern of suspicious events (e.g. fraud claims known to insurers);
- linkage with other (suspicious) circumstances;
- facts determined in the course of attempting to verify claimants stories;
- inconsistencies, as determined by reference to other agencies database; and
- model-directed discovery of new methods for verifying the information provided.

Using the knowledge implied in the respective hypothesis/explanation and extracted from online information sources, such as the *Australian Securities Commission* (ASC) and *Australian Electoral Roll*, the process for examining anomalous relations will, if successful, throw up some novel evidence and further leads for the investigation. Field studies may be necessary under some circumstances.

3.5.3 Structure of the Anomaly Detection Model

The structure of our *Anomaly Detection Model* is shown as FIGURE 3.6. In this model, there are two types of knowledge involved in anomaly detection.

1. Knowledge represented as rules, which includes the strategy to analyse statistical results, and threshold rules for determination of anomalous data. This type of knowledge is also used to detect inconsistent data.

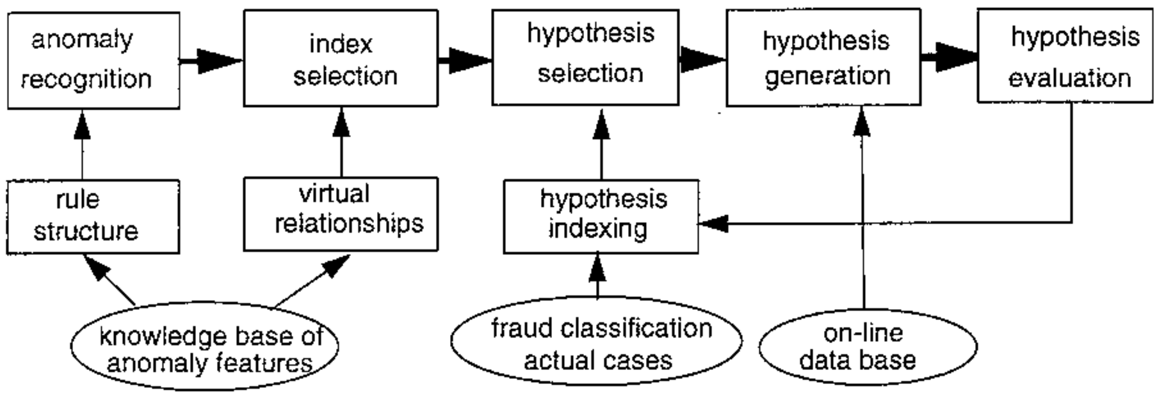


FIGURE 3.6 Anomaly Detection Model

2. Knowledge assisted detection of anomalous relations, based on hypothesis guidance. The knowledge required for use of an hypothesis is derived from fraud indicators, from verification sources and from information retrieval techniques. Use of a particular hypothesis implies the availability of the following knowledge:

- Description of events affecting or relating the claims;
- Historical Pattern of events (fraud claims);

- Knowledge about evidential anomalies;
- Information sources to verify the hypothesis; and
- Methods of using on-line information to detect anomalous relations.

As shown in figure FIGURE 3.5, the hypothesis structures are based on highly nested conceptual graphs[151]. The representation has advantages in using the conceptual graph formalism as well as the actor formalism [54]. The proposed data structure representing the hypotheses has to accommodate state change information and methods, and also handle incoming and out-going messages, together with the capability to transfer control to various entities during graph execution [54][95].

3.5.4 Reasoning in the Anomaly Detection Model

In the knowledge acquisition conducted by the author in the *Special Claims Unit*, fraud investigators have demonstrated highly developed abductive reasoning skills. Broadly speaking, abductive reasoning is any reasoning process which derives the best explanation(s) for a given set of problem features/evidence [120]. Like deduction, abduction requires that we find pertinent facts and apply them to infer a new fact. However, unlike deduction, ambiguous answers can arise in abductive reasoning [19].

Explanation plays a vital role in cognitive processes used in this anomaly detection model. In the anomaly understanding process, this role ranges from self-evident connections to the generation of complex relations between any nodes in the reasoning network.

The definition of an explanation is defined as:

$$\text{explanation} = \mathit{Explanation}(\text{node}_1, \text{node}_2)$$

or

$$\text{explanation} = \mathit{Relation}\ (\text{node}_1, \text{node}_2)$$

This latter explanation is defined as explaining the relation linking the node_1 to node_2.

Such explanation definitions are primarily based on the specific domain knowledge embodied in the anomaly detection model. Note, however, that we concerned with the provability of an hypothesis rather than an explanation of the behavior of the model itself.

The purpose of such explanations is to construct an hypothesis, which enables better understanding of anomalous evidence. The effectiveness of explanation is mainly determined by the conceptual richness of its knowledge sources.

In the domain of insurance fraud detection, we are particularly interested in the individual conditional probabilities connected through evidence to certain type of fraud. In particular, given that partial evidence may be present, there could be many hypotheses available, and each could be mandated for investigation.The detective is interested in the most likely overall hypothesis or explanation for the occurring evidence.

There is a considerable literature on reasoning under uncertainty. Most publications appear to address one of the four (4) following theories:

- Bayesian Inference[19],
- Dempster-Shafer Theory of Evidence [30][147],
- Certainty Factor Model[14], and
- Theory of Possibility [166].

In our application to insurance fraud detection, evidence can be seen as statistically independent not only in generic fraud cases, but also in special types of fraud. We are interested in a reasonable hypothesis for a short time rather than a perfect one for a long time. The Bayesian model has, therefore, been chosen as the basis for bottom-up inference in our reasoning engine [19]. The top-down reasoning process is a straight-forward application of backward chaining for reasons of efficiency.

In fraud detection, complicated situations may arise where the reasoning process first propagates probabilities up through intermediaries to hypothesis, following which, top-down reasoning may occur. FIGURE 3.7 explains, by example, this Bottom-up and Top-down reasoning process.

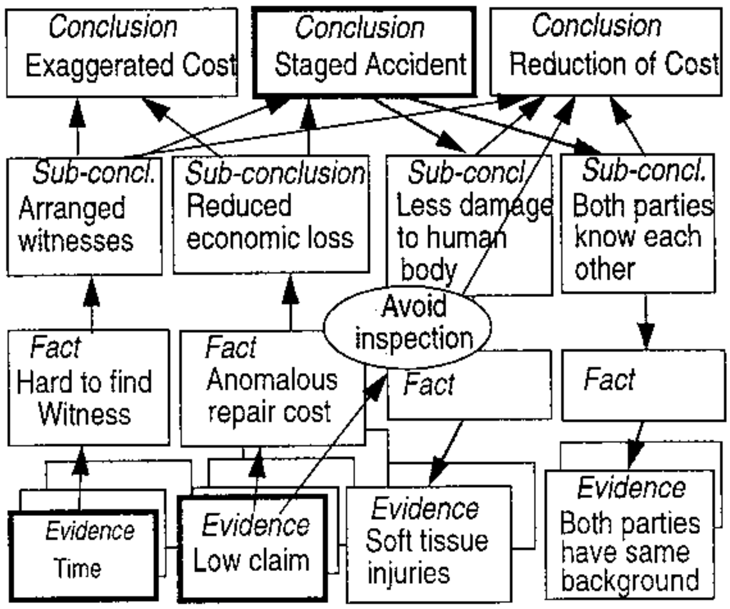


FIGURE 3.7 Reasoning Process in Action

In the anomaly detection model, the probabilities are not directly propagated up to the *conclusion* but are directed through some intermediate steps. This structure provides a flexible strategy to control the reasoning process, and requires only simple semantics to explain relations between neighboring nodes. As shown in FIGURE 3.7, the *facts* are evaluated first, and finally the *conclusion*s.

3.6 Hypothesis Based Problem Solving Cycles

Through research conducted at Deakin University, we conclude that an hypothesis based problem solving process can be described as one involving Abduction, Resolution, Verification, and Deduction, as shown in FIGURE 3.8. This result is analogous to the human cognitive model in a complex problem solving process.

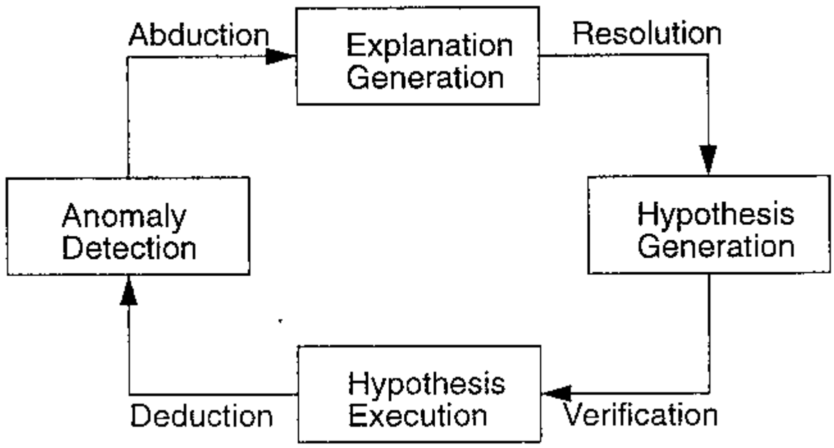


FIGURE 3.8 Hypothesis Based Problem Solving Cycle

Abduction is used to generate explanations, i.e., abducible sentence, whereas resolution is used to synthesize the conflict explanation. Deduction is normally used for testing derivability. In a deduction step, each consequence is obtained by applying a logically correct inferential rule, so the deduced formulae are a logical consequence of the theory under consideration.

In our experiment, once the hypothesis is defined, the problem solving simply becomes hypothesis verification. Thus, the mechanism for generating an hypothesis must have the following features:

1. control capability for inference using strategies implied in the hypothesis structure (that is executable relations);
2. improved communications between the user and expert systems in the process of problem solving; and
3. to focus the system resources on interesting aspects of the problem solving, and thereby avoid exhaustive search.

The strength of hypothesis based problem solving comes from the evaluation of complex hypotheses. We are also investigating the possibilities of controlling the inference with novel control strategies derived from hypothesis structures. This view of problem solving is different from the classical methods based on a single trace of reasoning, or from newer approaches using multiple threads of reasoning, which provide the problem solver with consistent alternatives, but make no attempt to evaluate their respective credibility.

3.7 Summary

In this chapter, the prototype specifications of a management intelligence system capable of meeting the requirement of an unstructured environment were analyzed. The conceptual framework advanced has three subsystems: anomaly detection model, hypothesis generation model, and hypothesis execution system. Our framework for a management intelligence system draws on the contributions made by Kochen [80] in a business management context.

A management intelligence system is ideally an extension of the domain expert's cognitive model. Its map of concepts and relationships extends the user's cognitive space, pushing back cognitive limits and expanding knowledge levels. Its knowledge processing capabilities increase the user's skills, by overcoming cognitive limits on the speed and capacity of knowledge processing.

The management intelligence system can also be seen as a knowledge based system supported by an hypothesis generation paradigm. The kernel of our framework is hypothesis generation modelling. Here, we have proposed a knowledge structure for hypothesis representation after briefly reviewing the limitations of current knowledge representation schemes. An original approach to explanation generation is offered by combining planning heuristics with explanation based learning, and by using a high-level specification for the representation of knowledge about the domain, its principles and terminology.

A vital step in hypothesis generation and indexing is the process of detecting anomalies. In this chapter, we briefly introduced an anomaly detection model, which is based on the hypothesis guided problem solving mechanism. An important feature of this anomaly detection model is its ability to detect not only anomalous data, but also anomalous relations.

Once the hypothesis is defined, the problem solving is, in effect, simplified as a process of hypothesis verification. The inference control capability has been used to focus the system resources on interesting aspects, thereby improving human-computer interaction.

In the next chapter, the rationale and significance of hypothesis generation modelling are explored in depth. The contribution made by abductive reasoning to this paradigm is explained in chapter 5.

CHAPTER 4

Hypothesis Generation Modelling

4.1 Introduction

All problem solvers are believed [90] to generate hypotheses in developing a solution framework. Hypothesis has been variously construed as a cognitive structure, as meta-knowledge or as 'possible worlds' knowledge. Our computational model for representing and using hypothesis to enhance reasoning skills reflects the need, identified in complex problem solving environments such as management intelligence, for knowledge integration and re-organisation. The paradigm proposed for hypothesis space elaboration and reasoning is original in supporting the diverse knowledge types and levels for problem explication and explanatory functions.

Quite recently, researchers in Artificial Intelligence have signalled the importance of hypothesis generation [29][33][51][99][101][108]. Such research has typically been limited to the assembly of an hypothesis rather than hypothesis generation [76]. One of the fundamental differences between hypothesis assembly and generation is the ability to deal with new situations, to discover new interpretations, and to form new theories that explain the observed data.

Case Based Explainer has theoretically solved the problem of building new explanations from old ones relying on having past explanations readily available in memory[145]. One obvious problem is what to do when no existing explanations relate to the facts at hand. This kind of problem will arise when the knowledge gaps between the current knowledge base and the real world become too big. We propose a framework which is based on *Case Based Interaction* principles [35][49] to overcome the limitations encountered in current *Explanation Generation* mechanisms [145].

4.2 Peirce's Early Contributions

Philosopher C. Peirce described a logic 'for the future' in his seminal work on abduction as source of new inferences[118]. Abductive reasoning is a form of logical inference, distinguished from induction and deduction, the traditional forms of reasoning. Abduction derives plausible explanations for a given set of observations and proceeds, in effect, by generating hypotheses, which may be combined with the respective domain information to 'explain' the given observation[39][115].

Peirce's classification is thus different from the traditional one, because it includes a novel type of inference, in addition to induction and deduction. Peirce's classification of inference is as follows:

Explicative (analytic or deductive)

Inference

Abductive

Ampliative (synthetic)

Inductive

According to Peirce's definition,

Deduction: an analytic process based on the application of general rules to particular cases, with the inference of a result.

Induction: a synthetic reasoning which infers the rule from the case and the result.

Abduction: another form of synthetic inference, which infers the case from a rule and a result.

Peirce further described abduction as the "probational adoption of an hypothesis" in explaining observed facts (results). But he also pointed out that it was, however, a kind of weak inference, because we could not assert the truth of the explanation, only that it might be true!

Broadly speaking, abduction happens in all the processes by which theories and conceptions are engendered. These process operations are best illustrated in scientific hypothesis. Peirce thought this process was essentially inferential. "Although it is very little hampered by logical rules, nevertheless it is logical inference, asserting its conclusion only problematically or conjecturally, it is true, but nevertheless having a perfectly defined logical form"[118]. Abductive reasoning may be illustrated as follows:

The surprising fact C is observed,
But if A were true, C would be a matter of course;
Hence, there is reason to suspect that A is true[39].

Such a process is inferential because the hypothesis "is adopted for some reason, good or bad, and that reason is regarded as lending the hypothesis some plausibility"[118].

Hypothesis is where we find some surprising fact which would be explained by supposing that it was a case of a certain general rule, and thereupon adopt that supposition[39]. This sort of inference is called "making an hypothesis". In this kind of inference, it should be noted that "When we adopt a certain hypothesis, it is not alone, because it will explain the observed facts, but also because the contrary hypothesis would probably lead to results contrary to those observed"[118].

Although explanatory hypotheses may vary widely, Peirce identifies at least three kinds:

1. The kind which refers to facts unobserved when hypotheses are made, but which are capable of being observed. For example, upon entering a room I find many bags containing different kinds of beans. On a table

there is a heap of white beans; I may adopt the hypothesis that the heap was taken out of a bag which contained white beans only.

2. There are hypotheses which are incapable of modern observation. This is true of historical situations.

3. Thirdly, hypotheses may refer to entities which in the present state of knowledge are both factually and theoretically unobservable.

In order that the process of making an hypothesis should lead to a probable result, Peirce lists three rules which must be followed [39]:

1. The hypothesis should be distinctly put as a question, before making the observations which are to test its truth. In other words, we must try to see what the result of predictions from the hypothesis will be.

2. The respect in regard to which the resemblance is noted must be taken at random. We must not take a particular kind of prediction for which the hypothesis is known to be good.

3. The failure as well as the success of the predictions must be honestly noted. The whole proceeding must be fair and unbiased.

In one sense, proposing an hypothesis is no problem at all. But of the hundreds of hypotheses that might be suggested, only one is true. The problem of constructing a good hypothesis is analogous to the problem of choosing a good hypothesis. The two questions, in practice, merge together [39].

What does abduction consist of? Is it the logic of constructing an hypothesis, or the logic of selecting an hypothesis from among many possible ones? At the outlet these seem to be too entirely different questions, but, in practice they are comparable questions. The central problem of abduction is to analyze the conditions or the criteria for the best hypothesis [39]. Peirce names three main considerations that should guide our choice of an hypothesis [118]:

1. An hypothesis must explain the facts at hand (effectiveness).

2. An hypothesis must be capable of being subjected to experimental confirmation.
3. An hypothesis must be guided by economic consideration (efficiency).

The first two requirements, however, are only the condition of a good hypothesis. The third consideration, in which efficiency plays a considerable role, is a very important element in Peirce's theory of abduction. Since the number of possible hypotheses satisfying the first conditions may be very great, we are faced with the problem of deciding which one should be tested first.

4.3 Early Approaches to Hypothesis Generation

Early philosophers believed that inductive inference was a suitable mechanism to generate hypothesis[2]. Induction is the inference of the rule (major premise) from the case (minor premise) and results (conclusion). From induction, we can generalize from a number of cases of which something is true, and infer that the same thing is true of the whole case. Also, we can find a certain thing to be true of a certain proportion of cases and infer that it is true of the same proportion of the whole class.

After several year's research, Barker concluded, however, that induction is unable to account for the confirmation of hypotheses implying the existence of unobserved things, and so, he has been examining ways in which such hypotheses supposedly might be dispensed with[2]. Abductive reasoning is the inference of case from a rule and a result. Abduction is a form of logical inference that attempts to derive plausible "explanations" for some observed evidence. In fact, abduction is sometimes referred to as "inference to the best explanation"[117]. Essentially, an abductive inference proceeds by generating hypotheses which, when considered together with certain domain knowledge, would account for or "explain" the given data[115].

The most important function of hypothesis generation is the formation of a new "theory" that explains the observed data. Abductive reasoning, which is, in a general form, seen as the process behind insight and is the key mechanism that introduces new ideas, is expected to play a significant role in hypothesis generation.

The earliest concept of hypothesis generation in the artificial intelligence community was proposed by Morgan in 1971. In his work[104], Morgan showed how a complete set of truth-preserving rules for generating theorems could be turned into a complete set of falsehood-preserving rules for generating hypothesis. Morgan's method is based on deductive inference and it faces the same difficulties regarding inductive completeness.

4.4 Hypothesis Generation Modelling

In this section, we will propose a conceptual framework for hypothesis generation, involving the anomaly detection model, case based interaction, and conflict resolution strategies. The framework also makes it possible to integrate qualitative knowledge with the primary probabilities for use in abductive reasoning.

Domain knowledge consists of the qualitative information about the structural relations and their evidential attributes, and the quantitative information about the uncertainty associated with these relationships. It also contains the procedural knowledge which can be used to change state when executed.

4.4.1 Semantic Description of Hypothesis Generation

Two main components of an hypothesis are the domain attributes (such as facts, evidence, and conclusions) and the relations among subsets of these attributes. We denote an hypothesis as a collection of attributes, A, and relations, R, among them:

$$H = \{A, R\}$$

where: $A = \{a_1, a_2, \ldots, a_3\}$ is a finite non-empty set of attributes, which include evidences, facts and conclusions; $R \subseteq A \times A$ is a set of non-empty relations among the attributes A.

The attributes are constructed from three types of entities, such as:

1. concepts,

2. statements,
3. conceptual graphs.

Based on their function in hypothesis space, the attributes can be classified into three groups;

1. **Evidential attributes**, which usually consists of the factual knowledge of hypothesis space, (e.g. references to anomalous evidence,) and are activated by the anomaly detection model. They may also be used to index events;
2. **Factual attributes**, which are the middle component of hypothesis space, are connected directly by evidential attributes, and are usually a type of generalized evidence, or an intermediate state of evolution from evidence to conclusion;
3. **Conclusive attributes**, which are the higher levels of hypothesis space and are usually the respective interpretations of anomalous evidence.

In the initial version, relations were characterised as a type of explanatory relation and were used to explain the causal link between two attributes. This relation can be a simple conceptual relation as defined in the theory of conceptual structures[151], such as 'is a', 'is a type of', and 'is evidence of'. The relation, in more complex situations, can be a complex conceptual graph or nested conceptual graph.

In an hypothesis structure, the inference about underlying evidence (attributes) is achieved with the help of the relations that link the observed evidences to proposed factual attributes, and on subsequent analysis, to Conclusive attributes. The nature of this hypothesis generation paradigm, therefore, critically depends on the reliability of the explanatory relations, and on their ability to assimilate a new relation into hypothesis R without causing conflicts.

Considering a set of relations R (a presented theory), and a set of evidences E (observation) the hypothesis generation process can be characterized to a first approximation as the problem of finding a set of explanations X (abductive reasoning) that satisfy the following two conditions:

- $R \cup X \rightarrow E$,
- $R \cup X$ is consistent.

This characterization of hypothesis generation is independent of the knowledge representation in which R, E, and X are formulated. The proposed requirement denoted by "$R \cup X$ is consistent" is not explicit in Peirce's more informal characterization of abduction, but it is a natural further condition.

In fact, these two conditions are too loose to completely capture Peirce's notion. In particular, additional restrictions on X are necessary to distinguish abductive explanations from inductive generalizations. Moreover, we also need to restrict X. We do not want to explain one effect in terms of another effect, but only in terms of some cause. For both reasons, explanations are often restricted to belong to a special pre-specified, domain-specific class of sentences.

Theoretically, there are two ways to deal with the problem. One is to only generate a suitable explanation at the beginning. Even if possible, it may, however, be more time consuming during the generation process and would certainly require a more complex representation for the hypothesis space. An alternative way to deal with this problem is fast-prototyping by separating the hypothesis generation process into two steps; the abductive reasoning for generating plausible explanations, and the hypothesis synthesis for conflict resolution among the explanations.

Hypothesis Generation = Abductive Reasoning + Conflict Resolution

Our implementation is based on the second approach, in which an explanation is generated first, followed by the detection and resolution of conflicts.

4.4.2 Extending the Framework for Hypothesis Generation

The original framework for hypothesis generation contained only explanatory relations. This framework was unsuited, however, to the automated generation of hypotheses due to lack of procedural knowledge. For example, in the process of generating the hypothesis - *staged accident*, use was made of several pieces of evidence; namely

- *the time when the accident happened,*
- *the location where the accident happened*, and
- *both parties have same background.*

To make the hypothesis more acceptable, other evidence such as '*the driver/passenger has only soft tissue injuries*' should be examined, since people are usually unwilling to risk bodily injury. In this situation, it is useful to have a procedure by which a database containing medical reports is opened, and the information relevant to the driver/passenger is retrieved. If the driver/passenger has not sustained serious bodily injury, the credibility of the hypothesis is increased.

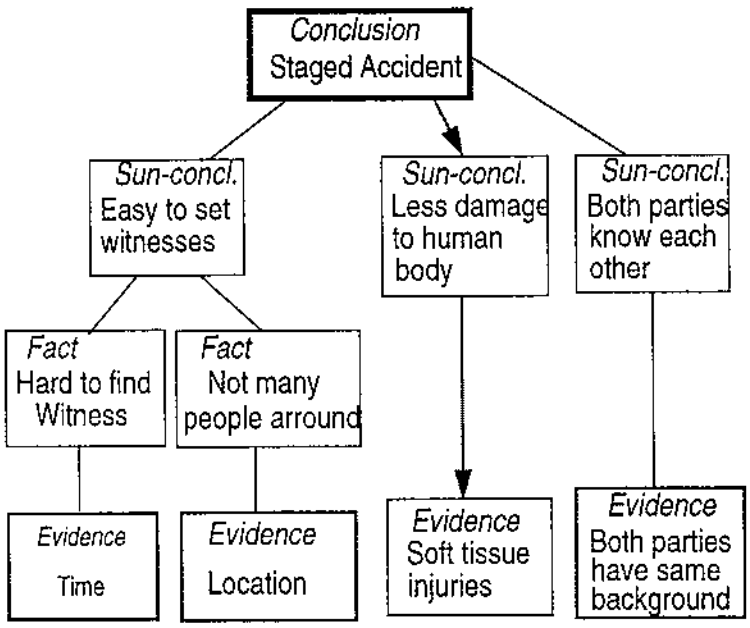


FIGURE 4.1 Example of Hypothesis Generation

To improve effectiveness in generating and verifying hypotheses, both declarative and procedural knowledge have been incorporated into the hypothesis, as different types of relations. The relations involved in the extended hypothesis generation model are categorized as two types:

1. **explanatory relation**

   This type of relation is used to explain the causal relation between two attributes. A simple explanatory relation can be in the form of *'is a'*, *'is a type of'*, and *'is an attribute of'*, while a complicated relation (explanation) is represented by a conceptual graph.

2. **executable relation**

   This type of relation is used to change from one state to another. A simple executable relation can be a command, while a more complicated one can be a software procedure, or even a pop up window containing another application.

To keep the hypothesis structures simple, the executable relations only serve as index features, the actual procedures are stored in memory while activated. The executable relations will be executed when the evidence connected is crucial to verify the hypothesis.

4.4.3 Virtual Representation of Hypothesis

In our experimental prototype of the hypothesis generation paradigm, the attributes A and relational explanation R are loosely and separately stored, and they are connected to each other only after being activated. In this way, the working space of the system can be reduced significantly, and the speed of response is also significantly improved.

FIGURE 4.2 describes a connected hypothesis which is activated by evidential attributes a_1, a_2, and a_4 (The circles connected by solid lines are activated). From the definition given in section 4.4.1 (page 82), the generated hypothesis can be seen as the mapping on the space $A \times R$ with the evidential attributes a_1, a_2, and a_4.

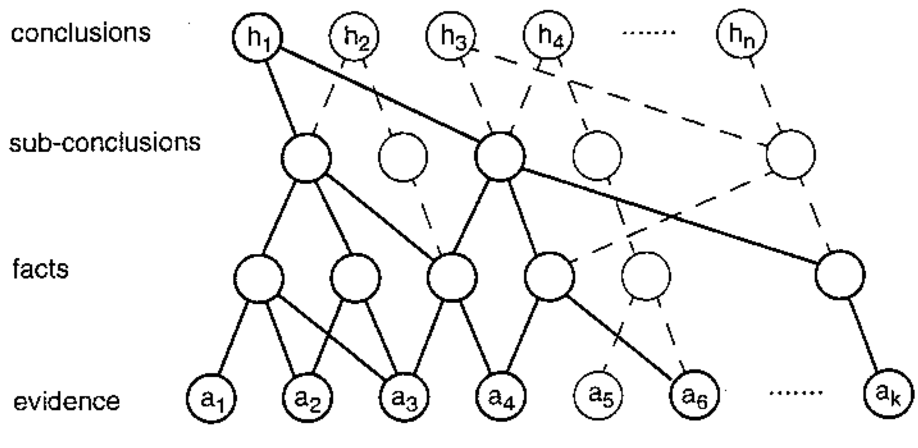


FIGURE 4.2 Virtual Representation of Hypothesis and Hypothesis Space

4.5 A Dynamic Hypothesis Space

A fundamental difference between our approach and the case-based reasoning paradigm lies in the difference in memory organization. Cases stored in memory are separated, and they are indexed by case features. There is no connection between the different cases. In our hypothesis generation model, the hypotheses (or cases) are dispersed in hypothesis space, in which the different hypotheses may be interconnected, and may share some common components. Hypotheses are implied in hypothesis space in an idle state. An hypothesis will emerge only when its relevant attributes are activated by anomaly detection model. In this way, we can say that the hypothesis space is a cluster of hypotheses about a specified domain.

$$HS = \Sigma H$$

The main advantage of introducing the notion of an hypothesis space is the facilitation of new ways to generate hypotheses, or a new explanation. This improvement is achieved by the knowledge sharing in the domain. Furthermore, the extent of memory required for storage of cases is reduced.

The hypothesis space is a dynamic knowledge base, in which part of system's memory is allowed to change. The working memory of a hypothesis generation model consists of all the *attribute-relation-attribute* relationships that are established during the knowledge acquisition process and will keep changing during the process of knowledge acquisition and problem solving.

4.5.1 The Evolution of Hypothesis Space

Goal interpretation has been developed as a mechanism for eliciting planning knowledge[95]. Subsequently extensions of this work on knowledge acquisition focussed on knowledge capture and construction of dynamic Canonical Graph Model that utilizes problem maps to represent the elicited planning knowledge[95].

Goal interpretation as a knowledge acquisition mechanism solves the problem of hypothesis space construction to some degree. The main intention of the goal interpretation mechanism for knowledge acquisition is to assist domain experts utilize, build and evaluate their own hypothesis space.

A dynamic hypothesis space has the following features:

1. Only relevant part is activated. Usually it is much smaller than the total hypothesis space;
2. Hypothesis space will expand as the knowledge acquisition proceeds;
3. Hypothesis space will shrink with knowledge fusion, and
4. During the problem solving process, the active hypothesis space will expand or contract depending on the intermediate results.

In this implementation, the hypothesis space consists of attributes and relations which are stored separately in attribute and relation tables. Two approaches are employed for controlling the hypothesis space. They are *Parsimony Covering Theory* and *Forward-backward Propagation.*

A number of different control criteria have been identified and used in related research work such as Single-disorder Restriction, Minimality, Irredundancy, and Relevancy. These criteria could also be applied to control the hypothesis space.

4.5.1.1 Restricting hypothesis space by parsimony covering

Parsimony covering theory, which is based on a formalization of causal associative knowledge[119], has been proved to be an effective approach in diagnostic problems, where all causal relationships are well known and can be easily represented by a function. Although it is potentially inefficient when new evidences or new relationships appear, it has been used very effectively to restrict the active extent of hypothesis space.

In our research, we have expanded the scope of parsimony covering theory to our hypothesis generation model. Based on parsimony covering theory, an active hypothesis space must cover all the initial attributes (evidence) in order to accommodate all evidences in E. On the other hand, not all covers of E are equally plausible for generating hypotheses for a given problem. The principle of parsimony is adopted as a criterion of plausibility: a 'simple' cover is preferable to a 'complex' one. Therefore, a restricted hypothesis space is defined as a parsimonious cover of that set of attributes that both covers the anomalies and satisfies the notion of being parsimonious or 'simple'. There is, in general, more than one possible cover for initial attributes, and users are often interested in all plausible hypotheses. In this situation, the set of all covers is defined to be the hypothesis space of a given problem.

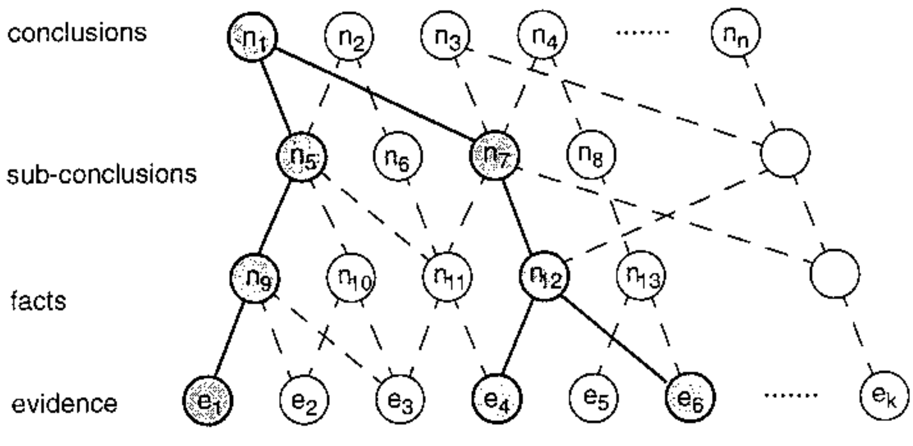


FIGURE 4.3 Restricting Hypothesis Space by Parsimony Covering

FIGURE 4.3 graphically illustrates the parsimony covering process in a partial hypothesis space. Let initial attributes $E = \{e_1, e_4, e_6\}$ be the initial evidence detected. Then the active space $A_0 = \{n_1, n_5, n_7, n_9, n_{12}, E\}$ is a minimal cover of E because it alone covers all evidences $\{e_1, e_4, e_6\}$. The space $A_1 = \{n_2, n_3, n_5, n_7, n_9, n_{12}, E\}$ is not redundant but not minimal because these attributes cannot cover the evidences $\{e_1, e_4, e_6\}$. The space $A_2 = \{n_1, n_2, n_3, n_5, n_7, n_9, n_{12}, E\}$ is relevant but redundant because A_0 is a subset of A_2. Finally $A3 = \{n_n, n_1, n_2, n_3, n_5, n_7, n_9, n_{12}, E\}$ is an irrelevant cover of E because there is no evidence in E causing n_n.

The determination of restricted hypothesis space is the first step of inference, and the type of relations is not considered at this stage. The initial (active) hypothesis space (working hypothesis space) will increase progressively in the reasoning process, until finally the working hypothesis space becomes a plausible hypothesis abstraction or schema.

Algorithm for hypothesis space restriction using Parsimony Covering

```
Input: An hypothesis space HS, initial evidence
Output: active hypothesis space AHS
Initialize: open = [initial evidence]; closed = []; AHS = []; n = 0;
while open <> [] do
    remove the leftmost evidence from open, called it X;
    if X can be found in HS
        then
        if X is conclusion node
        then put X in closed
            else
            generate all higher level nodes Y for X;
            put Y on the right end of open;
        end
    end
end
while closed <> [] do
    remove the leftmost conclusion node from closed, called C;
    generate all the nodes A under this conclusion C;
    counter the number of initial evidence N involved in A;
    if N > n then
        n = N
        AHS = A
    end
end
```

This algorithm is not strictly parsimonious for general graph searched. It achieves a very good result in our prescribed hypothesis space and also provides the ability to cope with newly found evidence.

An alternative approach to determining the plausibility of an hypothesis space is to objectively calculate its probability using formal probability theory. Peng and Reggia [119] integrated formal probability theory into the framework of parsimonious covering theory. According to their approach, a prior probability p_i is associated with each disease (conclusion). A causal strength is associated with each causal association representing how frequently a disease causes a symptom (evidence).

Although Peng and Reggia's approach solves the problem of multi-membership classification, and provides a formal method for hypothesis likelihood calculation, the assumption that conclusions (i.e. disease) are independent is not valid in all domains.

4.5.1.1 Restricting hypothesis space by propagation

Another simple and useful approach is based upon forward-backward propagation. An example of a restricted hypothesis space based on this approach is shown as FIGURE 4.4, where the initial detected evidences $E = \{e_1, e_3\}$, and the restricted hypothesis space (A_0) $= \{n_1, n_2, n_5, n_6, n_9, n_{10}, n_{11}, e_1, e_2, e_3, e_4\}$

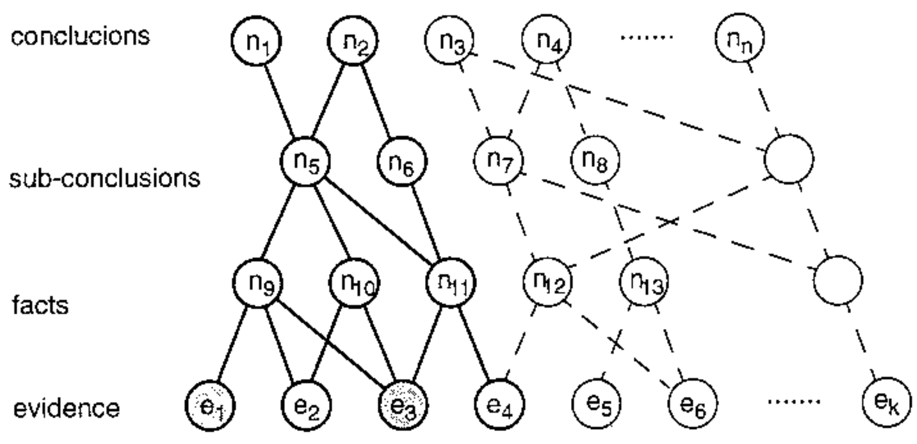


FIGURE 4.4 Restricting Hypothesis Space by Propagation

The scope of working hypothesis space based on forward-backward propagation is usually larger than one based on parsimony covering theory. The initial hypothesis space will decrease gradually in the reasoning process, until finally it becomes a plausible hypothesis.

Algorithm for forward-backward space restriction

```
Input: An hypothesis space HS, initial evidence
Output: active hypothesis space AHS
Initialize: open = [initial evidence]; closed = []; AHS = [];
while open <> [] do
    begin
    remove the leftmost evidence from open, called it X;
    if X can be found in HS
    then
        if X is conclusion node *
        then put X in closed
        else
            generate all higher level nodes Y for X;
            put Y on the right end of open;
        end
    end
end
while closed <> [] do
    remove the leftmost conclusion node from closed, called C;
    generate all the nodes A under this conclusion C;
    eliminate any elements of A already on AHS;
    put the remaining elements on AHS;
end
```

In comparing the two approaches, our research shows that the active space generated by parsimony covering based approach is smaller than that from forward-backward propagation. The first approach is more efficient in generating a plausible hypothesis while the latter is more effective.

Researchers in the machine learning field are also interested in hypothesis space and its restriction[99]. They have argued that in many cases the meta-knowledge can be extracted directly from the raw data and can be used to restrict the scale of hypothesis space. Due to different domains and a fundamentally different approach, it is not appropriate to compare the results.

4.5.2 The Evolution of Hypothesis Space

Research has indicated that this hypothesis generation paradigm, empowered by abductive reasoning, has the ability to acquire new explanations from existing evidence, and to cause the hypothesis space to expand. The relevant process reflected in hypothesis space is their evolution; more specifically, the generation of a new attribute or relation.

Attribute generation is simpler than relation generation. The process of generating an attribute is as follows:

1. selecting an attribute which has the shortest semantic distance from the evidence node,
2. modifying the label of the attribute by automatic modification strategies such as concept substitution, generalization, and specification,

The generated attribute inherits the properties from its relative, i.e. the attribute upon which the new one is based. The relations inherited from others will be deleted based on the following strategies:

1. all the executable relations are deleted,
2. virtual relation is checked with concept hierarchy table, and if in conflict, it is deleted; and
3. explanation relation is ranked by an explanation credibility measurement. It will be deleted if it's credibility is below the threshold.

Generating an explanatory relation is based on knowledge from the concept hierarchy table and is supported by the generalization/specification algorithms. Suppose a relation is required to link two attributes labeled a_1 and a_2. The system searches for corresponding concepts A_1 and A_2 from the concept hierarchy table. A_1 is the concept in the concept hierarch table, which has the shortest distance to a_1. A similar relationship also exists between A_2 and a_2. The relation between A_1 and a_1 (A_2 and a_2) may be equivalent, generalized, or specified. If the search is successful, the relation between A_1 and A_2 in hierarchy table will be adopted as a virtual relation through some simple revision. With the support of the concept hierarchy, virtual relations in hypothesis space are well maintained.

1. Generating a new explanatory relation is basically a case based approach. If failure occurs in generating a virtual relation between a_1 and a_2, the system turns to generating an explanatory relation. In our case based algorithm, a_1 and a_2 will be used as the indexes to retrieve the most suitable explanation. After some necessary explanation modifications, this explanation could be adopted as a new explanatory relation.

2. Case based explanation often fails when faced with a new situation. Currently, we are using abductive reasoning (a knowledge based strategy) to control the inference, and only restricted types of explanations are invoked and processed. In this way, the redundancy and irrelevancy of explanation, which is generated by abduction, will be reduced considerably.

3. Generating a new executable relation is the process of actor assembly. Every actor has its pre-condition and post-condition. To build an executable relation between two attributes a_1 and a_2, a typical breath-first search algorithm is utilized, and if successful, a sequence of actors, $\{actor_1, actor_2,..., actor_n\}$ is assembled, such that the pre-condition of $actor_1$ matches a_1 and post-condition of $actor_n$ matches a_2. This executable relation has the capability to change the state *of* a_1 into that of a_2.

4. It is not necessary to have an executable relation between most attributes. In insurance fraud detection, the executable relation between two attributes labeled "hard to find witness" and "accident time" is actually the procedure to open a database, read a file, and get the time the accident happened. The procedural knowledge, together with declara-

tive knowledge embedded in hypothesis space, make it possible to generate an hypothesis which can solve the problem without the support of massive knowledge bases.

4.5.3 Knowledge Fusion in Hypothesis Space

Knowledge fusion is the process of producing a simplified knowledge structure by the combination of two or more separate data/knowledge structures[56]. The hypothesis space is formed during the knowledge acquisition process and evolves in the hypothesis generation process. Without proper management, the hypothesis space will become more and more complex and inefficient. The effectiveness of indexing and retrieving knowledge from the hypothesis space is fundamental to the efficiency of the knowledge based system. Knowledge fusion is used to maintain effectiveness.

Knowledge fusion aims to increase the efficiency of knowledge management by simplifying the structure of hypothesis space. The fusion process is triggered by a housekeeping program when certain problem solving cycles are finished. At the macro level, the fusion processing replaces a set of relations {Relation(*type*, a_1, a_2, p_{12}), Relation(*type*, a_2, a_3, p_{23}),..., Relation(*type*, a_4, a_6, p_{46})} with a simpler one Relation (*type*, a_1, a_6, p_{16}) as shown in FIGURE 4.5. It usually contains executable knowledge which may change the state of a_1 into a_6, and an explanation, which interprets the relationship between a_1 and a_6; occasionally as a virtual relation.

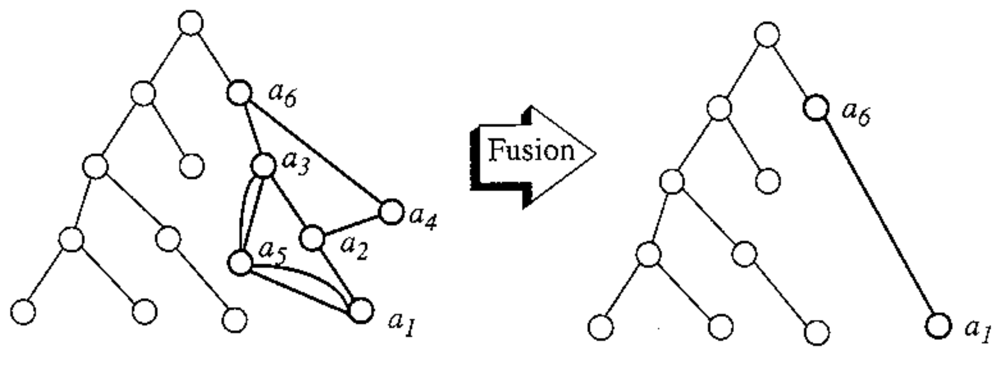


FIGURE 4.5 An Example of Knowledge Fusion in Hypothesis Space

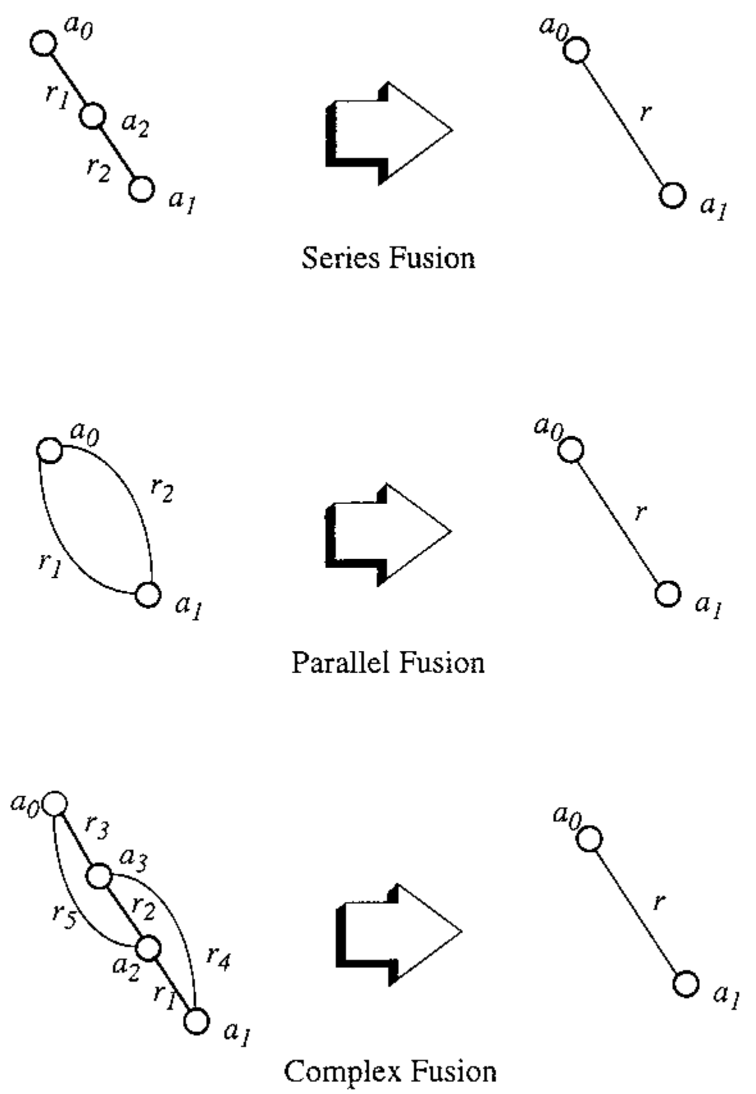


FIGURE 4.6 Three types of Basic Fusion Process

Although complicated, the fusion process can be broken down into three basic fusion processes: parallel, series and complex relation fusion. A graphical description for these processes is shown in FIGURE 4.6

The basic principle in a fusion process is that the simpler relation is preferred to a complicated one, and the executable relation is favored most. Under this principle, a virtual relation is better than an explanatory relation, a simpler explanation is better than a complex one, and an executable relation is better than all others.

Algorithm for series fusion processing:

1. delete attribute a_2,
2. if relations r_1 and r_2 are virtual or explanatory relations; then $r = r_1 + r_2$.
   The process of "$r = r_1 + r_2$" is a maximal-join operation with the conceptual graphs related to r_1 and r_2. If r_1 or r_2 is a virtual relation, a simple graph is constructed to perform maximal-join.
3. if relations r_1 and r_2 are executable relations, then $r = r_1$ & r_2.
   The process of "$r = r_1$ & r_2" is a merging process with:
   pre-condition(r) = pre-condition(r_1),
   post-condition(r) = post-condition(r_2),
   actor(r) = actor(r_1)+actor(r_2) (i.e. execute actor(r_1) followed by actor(r_2)).
4. if relation r_1 is virtual/explanation relation and r_2 is executable relation, then $r = r_2$ @ r_1.
   In these situations, actor(r) = actor (r_2), and the conceptual graph related to r_1 max-join with graph related to r_2 if it existed, otherwise just add to the explanation slot of executable relation.

Algorithm for parallel fusion processing:

1. if relation r_1 is virtual relation and r_2 is virtual/explanatory relation then $r = r_1$, and vice versa,
2. if relation r_1 is virtual/explanation relation and r_2 is executable relation then $r = r_2$ @ r_1,

Algorithm for complex fusion processing:

1. select one executable relation if it exists (suppose r4 is executable relation),
2. select the simplest path which links attribute a_1 to a_0 through the executable relation r_4 (that is r_4 and r_3 in FIGURE 4.6),

3. process $r = r_4 @ r_3$,
4. if there is no executable relation, the simpler path is chosen as $r = r_3 + r_4$ or $r = r_4 + r_3$.

The knowledge fusion algorithms are powerful tools to simplify hypothesis space. No matter how complicated, all lattices are composed of attributes linked by series, parallel, and complex relations. Oversimplification of hypothesis space should, however, be avoided by restricting at least three or four levels from fusion of any evidential attributes to consequent attributes.

4.6 Conceptual Graph Based Hypothesis Generation Model

The model for hypothesis generation proposed so far is a generic one, and there is no restriction on the formalism (knowledge representation). The definition (in section 4.4.1) can be easily applied to a conceptual graph based knowledge representation scheme. In our experiments with hypothesis structures, including all attributes and relations, conceptual graphs have been used.

Abduction as presented can be restricted by using integrity constraints. The concept of integrity constraints first arose in the database field and to a lesser extent, in the field of knowledge representation. The basic idea is that only certain knowledge elements are considered acceptable, and an integrity constraint is meant to enforce these restrictions. When conceptual graphs are used to perform abduction, the integrity constraints are used to restrict graph operations, and to reject unacceptable abductive explanations.

Given a set of integrity constraints of first-order closed formulae I, the second condition of the semantic definition of hypothesis generation can be represented by:

$$HS \cup X \text{ satisfies } I.$$

HS represents hypothesis space and X represents explanatory relations. There are several ways to define what it means for an explanation in hypothesis space HS, ($R \cup X$ in our case) to satisfy an integrity constraint I. In our prior experiments:

$HS \cup X$ satisfies I iff $HS \cup X$ is consistent.

In further experiments to explore alternative uses of I, only integrity constraints having practical value were explored, as that there are no conflicts among the explanations in hypothesis space. The absence of conflict is true for the knowledge base rather than the world modelled by the knowledge base.

4.6.1 Basic Definition and Notation

Two attributes a_i and a_j are said to have been explained by a chain of qualitative relations in an hypothesis. These qualitative relations, when active, provide the causal chain between a_i and a_j.

Consider the following definitions involved in hypothesis generation.

Attributes

The attributes in hypothesis generation are the nodes which are either a concept or a statement represented by conceptual graphs. The attribute can be used to represent evidence, states, facts or conclusions.

Evidential attributes and conclusion attributes

The evidential attributes are the lowest level attributes of any relations in hypothesis space, while the conclusion attributes are the attributes appearing at the top level.

Relations

A *relation* from an attribute a_i to another attribute a_j in an hypothesis is defined as an ordered sequence $\{a_i, a_{i+1}, a_{i+2},\ldots a_{j-1}, a_j\}$, where $a_{i+1}, a_{i+2},\ldots a_{j-1}$ refer to the intermediary attributes, a_i refers to the evidential attribute, while a_j is a consequent attribute.

An *explanatory relation* between a_i and a_j in an hypothesis in either direction constitutes an explanation for the causality between the evidential attribute and its consequent. Assuming this path of qualitative causal relationships to be active, then the two events would occur as a matter of course.

Another way of understanding two attributes a_i and a_j is to hypothesise a shared common ancestor attribute for them. If there is an attribute a_o such that there is a path from attribute a_o to the attribute a_i and another from the attribute a_o to the attribute a_j, then, the attribute a_o is hypothesised as being the cause for both the events.

An *executable relation* between a_i and a_j in an hypothesis can be used to invoke a procedure which can change the initial state a_i to consequential state a_j. The pre-condition of execution is decided by the priority of the consequential states when verifying the hypothesis.

The index format of relations in *Prolog* is as following:

$$Relation(type,\ a_i,\ a_j, p_{ij})$$

In this notation, the relations are indexed by the attributes, evidences/consequences (or initial/consequential state), where a_i and a_j, are attributes in hypothesis. Here *type* in a relation notation can be explanatory or executable.

In relation notation, p_{ij} is an indicator representing the credibility measurement of explanation between a_i and a_j, it can also be represented by the prior probability of $P(a_i \mid a_j)$. This indicator will be used to control the reasoning process in an hypothesis generation.

Note that relation $Relation(type,\ a_1,\ a_2,\ p_{12})$ is not equal to the $Relation(type,\ a_2,\ a_1,\ p_{21})$.

Hypothesis Space

Let $HS = [A^*,\ R^*]$ be an hypothesis space. The hypothesis space HS is defined as a lattice which consists of a set of nodes A^* (attributes) connected by a set of relations R^*. The generic space is instantiated for a pre-specified application domain.

Hypothesis

Given a set of attributes A, and a set of relations R, an hypothesis $H = (A,\ R)$, is that subset of hypothesis space HS formed by connecting the separated attributes with relational explanations while obeying the following constraints:

1. $A \in A^*$, $R \in R^*$, and $H \in HS$;

2. each attribute has at least one qualitative causal relation connected to another attribute;
3. there are no conflicts among the explanations in an hypothesis.

Hypothesis Generation

Given a set of attributes A, and a set of relations R: For a set of attributes A', $A' \subseteq U$, U is domain knowledge and A' is part of the domain knowledge. Hypothesis generation is the process to generate a set of relational explanations for connecting A' and A. If there is no suitable explanation or attribute, the process will create a new one. An hypothesis generation problem is thus defined to be

$$H = [A, R]$$

where $A = \{a_1, a_2, \ldots a_n\}$ is a finite non-empty set consisting of $n = |A|$ attributes, including *evidence, facts, sub-conclusion* and *conclusion*. R is a class of non-empty relations, and includes *explanatory* and *executable relations*.

Based on this hypothesis space, the hypothesis generation problem is to select a suitable subset of hypothesis space *HS*. The subset of hypothesis space must include the observed attributes if they exist in hypothesis space. Otherwise, it is necessary to incorporate the new attributes in the hypothesis space and to generate a corresponding relation.

In order to qualify as an hypothesis, by which an inference about a set of evidences can be made, the hypothesis generated should include all the observed evidential attributes and at least one conclusion attribute. It should include explanations for all evidences taken a pair at a time. Of course, there is no conflict among the explanations in generating the hypothesis.

In our application, A' represents the evidence detected. A and R represent the existing knowledge base. In the simplest situation: $A' \subseteq A$, the evidence detected is already defined in the existing hypothesis space, and therefore, the existing explanations can be used to explain them. This is a special case of hypothesis based problem solving. The system works in a similar way to a case base reasoning process.

The hypothesis generation paradigm provided here is mainly concerned with situations in which not all elements in A' can be found in A. In such situations, a new (observed) piece of evidence and a new explanation will be generated to expand the hypothesis space.

Conflict

In this hypothesis generation paradigm, conflict can theoretically be divided into two types; namely explanation conflicts and execution conflicts. In our research, only the explanation conflicts are considered, based on the following reasons:

- the executable relations in our application domain are closely related to the evidence, and they are, in most cases, executed in parallel. So, the execution of one executable relation will not change the pre-condition of others.
- the executable relation is used to verify evidential propositions. Once execution has finished, we are only interested in the truth value of evidential propositions and the explanation between them. The executable relation is no longer useful.

Let $HS = [A, R]$ be an hypothesis space, in which a conflict is:

1. a subset C of R such that explanation cannot be intact under current observations, (contradicts observations)
2. a subset C cannot be intact among the other explanations (contradicts other explanations).

The conflict C is identified by an inference engine using the design model for hypothesis generation and from current observations. Each conflict in R is a set of components which cannot all be intact under current observations or other explanations. In other words, a conflict $C \in R$ represents the explanation that there is at least one anomaly within C.

Conflicts are usually detected when erroneous values are observed as output variables. In general, conflicts occur when two different values are predicted for a variable using same test. The definition of conflicts (above) is very loose, and several concepts require formalisation. A detailed discussion on conflict detection and conflict resolution is presented in chapter 6.

4.7 Hypothesis Generation Processing

In this section, we provide a graphical model (flow chart FIGURE 4.7) to illustrate at an abstract level the relations between the main components in the hypothesis generation model.

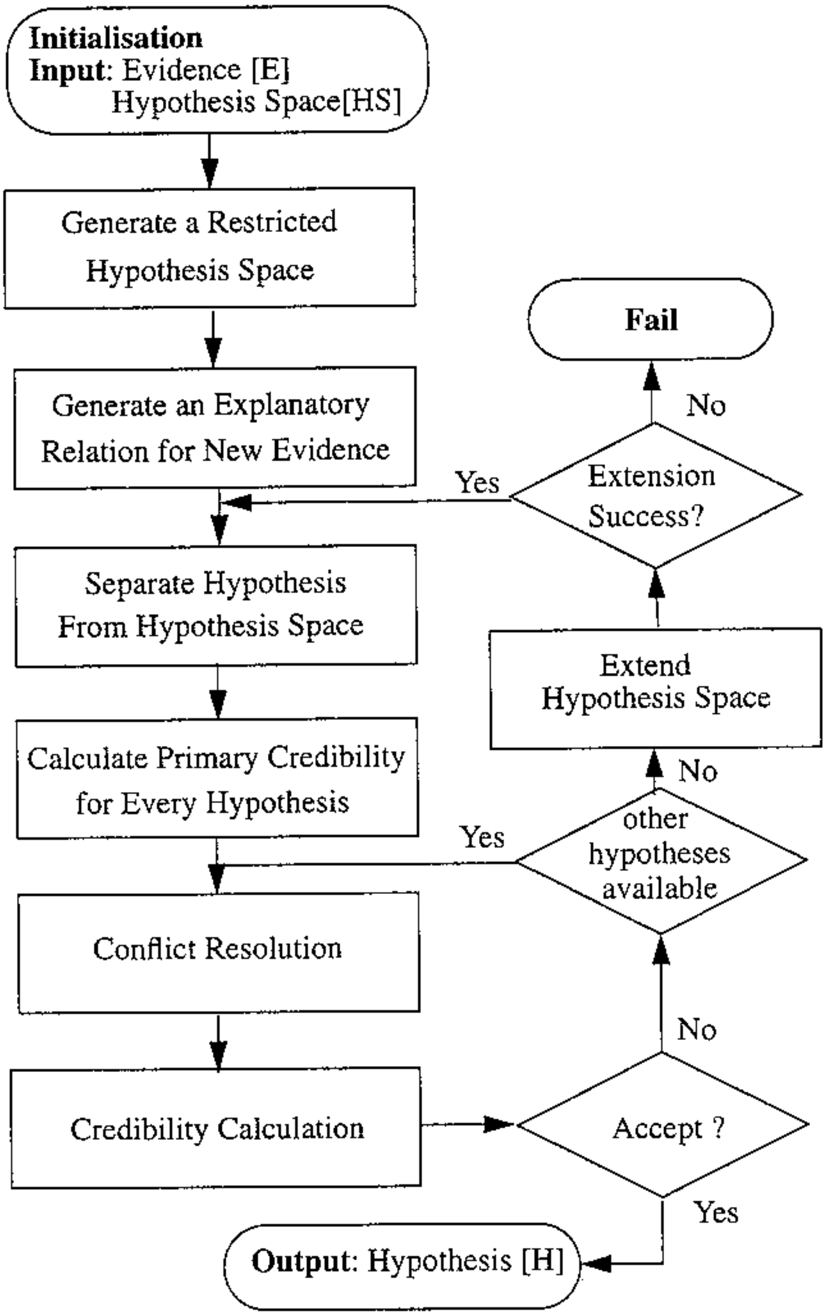


FIGURE 4.7 Flow Chart of Hypothesis Generation Processing

4.8 Summary

In this chapter, we have focussed on the issue of knowledge representation. A kernel for future management intelligence systems has been proposed based on an hypothesis generation paradigm. More specifically, the hypothesis generation model, supported by a dynamic hypothesis space, provides the capability to represent both case based explanations and novel interpretations of evidence

The contributions of Peirce on abduction and hypothesis were first reviewed, from which the potential of abduction for generating hypotheses was developed.

The knowledge representation requirements for a conceptual hypothesis structure were then addressed. The main advantage of this structure is its ability to accommodate both declarative and executable knowledge, and also its capability to imply inference control strategies in its extended version, a very important feature in dynamic problem solving.

Subsequent extension of the definition of hypothesis includes the notation of an hypothesis space. Management of the knowledge space is supported by a space restriction algorithm (which is based on either parsimony covering theory or forward-backward propagation), and a knowledge fusion algorithm. Technical advantages of this approach are evident in:

- providing a platform to actualize hypothesis generation processing,
- reducing the size of hypothesis space, and
- sharing knowledge in different hypotheses.

Finally, substantive results of implementing this approach in conceptual graphs (CG) have been reported. The generic algorithm developed for hypothesis generation modelling has provided the experimental theory for CG based knowledge guided abductive reasoning.

In summary, the experimental justification provided for hypothesis generation modelling for management intelligence has identified new research opportunities in knowledge guided abductive reasoning.

CHAPTER 5

Abductive Inference Control in Hypothesis Generation

5.1 Introduction

Abduction was frequently confused with induction until Peirce distinguished it as one of the three fundamental types of logical inference: abduction, deduction and induction. In the artificial intelligence literature, research into abductive inference was re-activated by Pople in 1973 [126] and subsequently, by Charniak and McDermott in 1985[19]. At its simplest, abduction is a plausible reasoning process, which derives the best explanation(s) for a given set of problem features (evidence).

Abduction provides hypotheses about why a statement is true, while deduction solves the problem of determining whether a given statement is true. Abduction requires a reasoning engine that can incorporate a set of new observations into a theory of the world by determining what assumptions should be added to the theory, so that it accounts for the new observations.

Like deduction, abduction requires that we find pertinent facts and apply them to infer a new fact. However, unlike deduction, we can get ambiguous answers in abduction[19].

Unfortunately, there is no obvious strategy to choose among the alternatives. The best we can do is to determine which hypothesis is the more plausible?

In the case of fraud detection, the conditional probabilities of possible frauds is of less interest than the most likely hypothesis and explanation for the evidence at hand. For instance, suppose that a detective finds that the credit limit is exceeded in one account. There might be many explanations available to explain this evidence. Our purpose is not to list all hypotheses, but to provide the best explanation, which should also accommodate other real world knowledge.

5.2 Contemporary Approaches to Abductive Reasoning

It is commonly recognized that there are many versions of abduction in the artificial intelligence literature[19][117], namely: classical approaches, set cover based approaches, logic based approaches, knowledge level approaches; etc.

Knowledge for abduction is generally presented as explanations and rules to apply to the explanations. The objective of abduction is to find a set of explanations generated by the rules as consistent with the evidence. Usually, there would be more than one set of explanations satisfying the observed evidence. The explanations are considered to be the hypotheses or assumptions underlying the observations.

5.2.1 Classical Approach

The classical way to deal with abductive reasoning is based on Bayes's Theorem:

$$P(cause|evidence) = \frac{P(cause)\,P(evidence|cause)}{P(evidence)}$$

We describe this process in the following way. Given the conditional probabilities *P(evidence|cause)* and the unconditional probabilities *P(cause)* and *P(evidence)* for the natural occurrence of causes and evidence of interest, respectively, then the probability that the observed evidence signifies a possible cause can be calculated for each cause. The best known expert systems, such as *Mycin* and *Caduceus*, utilise Bayesian statistics.

It should be noted that Bayes's theorem makes a very strong assumption about the independence of evidence, and the independence of evidence for given causes. For this reason, classical abductive reasoning has severe, practical limitations inherited from Bayes's assumptions.

5.2.2 Set-cover Based Approaches

In *set-cover based approaches*, a set of explanations are found by selecting a suitable subset from a given set of explanations[38]. This subset should best account for the observations, and is determined by coverings, parsimony, plausibility or another suitable selection criterion. Since hypotheses constructed in this way use a set of previously known candidates, this approach is also called hypothesis assembly[38].

There are some limitations that cannot be neglected. First and foremost, the computability of the mapping (covering) is crucial for the choice of possible explanations. All causal relationships that might be relevant must be encoded in the form of relations before starting the abductive process. This seems practicable only in restricted areas, e.g., diagnostic problems.

Apart from that, the domain must satisfy some further assumptions[117]. Set-cover based approaches rely heavily on previously known relations from which a super-set of the desired explanations is determined. Levesque [90] concluded that the set-cover based abduction model appears to be adequate only for diagnostic tasks or repair problems, where all causal relationships are well known and can easily be represented by function. Another disadvantage is the sensitivity of hypothesis selection to background knowledge.

5.2.3 Logic Based Approaches

The majority of research in abduction is based on logic models [117][124]. The logic based approach is widespread as it allows more flexibility, and thus seems to be adequate for a wide range of applications. The weakness of this approach is the implicit reliance on a special knowledge representation scheme. Causal relationships among the facts and evidences must be determined, since they are responsible for choosing the right explanations.

As noted by Levesque, however, logic-based abduction is defined over global logical properties, such as consistency and derivability, and this seems to be a limitation. In addition, the knowledge about causal relationships that are used for selecting abductive explanations is represented implicitly in the global theory[117]. The disadvantage of this approach lies in the limiting specification of reasoning into global properties of the logic, such as consistency and implication. Different reasoning abilities, deductive or abductive, will then require different notions of implication or consistency[90].

5.2.4 Knowledge Based Approaches

Levesque gives an account of abduction at the knowledge level[90]. The knowledge based approach is based on a model of belief. It goes one step further and defines a model for abduction independent of a belief type. Levesque shows that the knowledge level approach subsumes the logic based model for implicit belief. Hence, further generality has been gained.

Nevertheless, it remains necessary to represent causality at the computational level in explicit terms. So, in order to gain further insight into the scope of abduction for this investigation, the role of causality should be incorporated into the relationship between abduction and induction, thereby making explicit the close connection stressed by Peirce.

5.2.5 Conceptual Graph Based Approaches

The conceptual graph based approaches demonstrate the feasibility of the implementation. Tsui has suggested that maximal-join operation could be a useful tool for plausible reasoning[156]. Hartley and Coombs [68] proposed an operator which is composed of the two primitives, *cover* and *maximal-join* for abductive reasoning.

$$Abduction = Cover + Maximal\text{-}join$$

The function of *cover* is to choose an appropriate subset of stored graphs, which cover all of the concepts in a given subset of graphs extracted from those inputs. The function of producing an explanatory hypothesis is based on the *Maximal-join* operation[151][156]. This approach is limited by the capability of choosing the most suitable graphs, as there may be many that satisfy the above condition. Furthermore, a satisfactory algorithm for cover is hard to define.

On the other hand, Pagnucco proposed an approach which was also based on conceptual graphs[115]. In his approach, he utilizes the *Sheet Of Assertion*, and in particular, the graphs that belong to it are deemed to represent the domain knowledge. Abductive inference can be viewed as the process of determining a set of graphs to add to the sheet of assertion in order to prove the given data, and which are also consistent with the graphs on the sheet of assertion. As Pagnucco admitted, there are no criteria for selecting the best abduction from those explanations derived, and there are also some syntactic restrictions on the representation of conceptual graphs.

5.2.6 Other Approaches

The above approaches represent the main trends in development of an abductive inference engine. Other methods are no doubt under investigation. However, besides the four principal methods briefly reviewed, complementary approaches are to be found dealing with abductive inference, in some degree, as a subgoal of their main goal.

Weight abduction [74] was proposed by Hobbs Stickel et al at SRI International. It uses assigned weights and costs to make individual assumptions. The cost of an explanation is a function of the cost of the individual assumptions made in reaching the explanation. This cost is used in an effort to guide the abductive inference by favoring the intended explanations. The final choice for best explanation will be the one with lowest cost.

The main weakness of this approach, however, is the lack of clear semantics for the cost assignments. Furthermore, objective ways to assign weights and costs remain an open question.

A minor variant of weighted abduction was presented by Charniak and Shimony as cost based abduction, in which hypotheses have associated costs, and the cost of a proof is simply the sum of the costs of the hypotheses required to complete that proof [20]. The key idea to this approach is to utilise directed acyclic graphs (or weighted AND/OR directed acyclic graphs) to represent the relationships between hypotheses and the evidence to be explained. Each node represents some piece of knowledge, and the connections explicitly detail the relationships between the different pieces. Unfortunately, finding minimal cost has been shown to be very difficult [20].

5.3 Abductive Inference for Hypothesis Generation

In our research, the most important goal of abductive inference has been to generate a new hypothesis that explains the newly detected evidence. The new hypothesis means that there are some pieces of evidence which cannot be explained by existing knowledge. In other words, there is no attribute in the activated hypothesis space that can match the evidence observed, and therefore, there is no explanatory relation which can be connected to the new evidence.

Hypothesis can be seen as a set of explanations to explain observed evidence. The outcome of abduction is an explanation for this evidence. For a given collection of evidence, the abductive inference engine will generate a number of explanations. In this chapter, we defer consideration of the conflicts among the explanations and the evaluation of explanations until the next chapter.

Our abductive inference engine processes only the new evidence provided by the anomaly detection model. Where there are existing attributes which match the new evidence, existing explanations will be indexed to explain them. In this case, abductive inference is not activated.

5.3.1 Problem Descriptions

We begin by reiterating the notation for abduction in terms of conceptual graphs as defined by Pagnucco recently[115]. An abduction (in terms of conceptual graphs) is a set of graphs which, when added to a set of graphs denoting our domain knowledge, will allow us to 'account' for the given data (a graph) while maintaining the domain knowledge.

Abductive inference requires knowledge to perform the inference and rules to apply the knowledge. In our hypothesis generation model, the knowledge is the explanations in hypothesis space, (i.e., the explanatory relations,) while the rules are implied in the topological structure of hypothesis space.

Two components of an hypothesis are the domain attributes (such as facts, evidence, and conclusions) and the relations among these attributes.For simplicity, we only consider explanatory relations in hypothesis space. There are three reasons for this simplification:

1. Abductive inference is mainly concerned with generating explanations for observed evidence.
2. The function of executable relation is to subsequently verify the explanation.
3. In our application domain, most executable relations are connected to the evidence (that is the bottom nodes). The execution of one relation will not cause a state change for another relation.

For these reasons, executable relations are ignored in the abductive process. The definition of hypothesis space is now revised as a collection of attributes, A, and relational explanations, RX, among them:

$$HS = \{A, RX\}$$

where: $A = \{a_1, a_2,..., a_3\}$ is a finite non-empty set of attributes, which includes evidences, facts and conclusions; and RX is a set of non-empty relations within attributes A.

Therefore, abductive inference is defined in relation to hypothesis space as the problem of generating an explanatory relation, which can be used to connect the new evidential attribute to the hypothesis space.

If there is no existing relation connecting the observed evidence, domain knowledge would be used to establish the association between the evidence node and the explanations. Case Based Explainer [145] and knowledge level approaches [90] have dealt with this type of problem to some degree.The new explanations are derived from old explanations through appropriate modification and rely on having explanations readily available in memory and knowledge structures that package the reasoning strategy.

The two approaches described above have been combined in this research project and satisfy the requirement to absorb new explanations, generated by the abductive reasoning process, into a subset of relational explanations without causing conflict.

5.3.2 Operation for Abductive Reasoning

Existing methods for abductive inference are generally neutral to prior experience and current goals[68][115]. Candidate explanations are built from scratch by means of forward and backward chaining, without considering how similar situations were previously explained. Problems arise when applying these methods to complex problem solving. For example, the large number of possible explanations makes it difficult to reduce the cost of selecting the "best" explanation and difficult to ensure that the hypotheses generated will actually be useful.

An alternative model that addresses the weakness of standard abductive inference control by using previous knowledge and current goals to guide the inference is advocated here. Being consistent with the common view of abduction, our model also characterizes the abduction task as finding plausible explanations for observed evidence.

Maximal-join is advocated as an appropriate operation for conceptual graph based abductive reasoning. The theoretical basis for this research relies on canonical rules of formation[106]:

By formation rules the resultant graphs from canonical graphs are canonical.

A conceptual graph is a combination of concept nodes and relation nodes where every arc of every conceptual relation is linked to a concept. But not all such combinations make sense. For example, some of them include absurd combinations such as:

$$[\text{Sleep}] \rightarrow (\text{AGNT}) \rightarrow [\text{Idea}] \rightarrow (\text{COLR}) \rightarrow [\text{Green}]$$

This is an odd, unusual, or perhaps meaningless graph[151].

To distinguish the meaningful graphs that represent real or possible situations in the real world, certain graphs are declared to be *canonical*. Based on Sowa's theory[151], new conceptual graphs are canonical if they are derived from canonical graphs by canonical formation rules. The rules can be classified into:

- Equivalent rules (*Copy* and *Simplify*)
- Specialization rules (*Restrict* and *Join*)
- Generalization rules (*Unrestrict* and *Detach*)

The *Join* operation is to perform the integration of two graphs. Two concepts are "joinable" if and only if they have the same type label and the same referent. In the *Max-join* all corresponding compatible concepts and relations between the two original graphs are unified together [151].

We believe that *Max-join* holds the key to success in conceptual graph based abductive inference and can provide a set of possible explanations for further development [151][156]. Conceptual graph based abduction is defined here as:

$$\text{Explanation} = \text{Background\_knowledge} \otimes \text{Evidence}$$

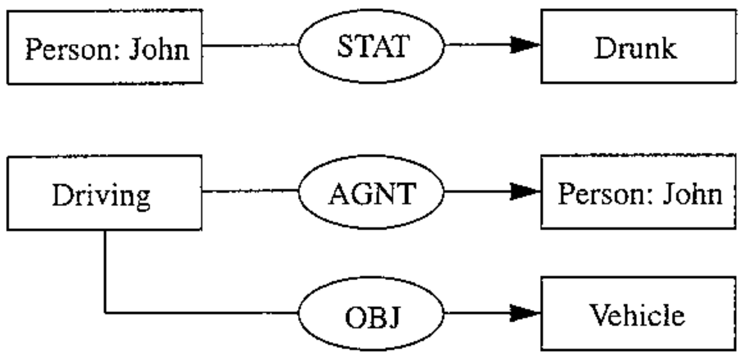
The operator $\otimes$ employed here is used to represent the *Maximal-Join* of conceptual graphs. Minimum requirements for a *Maximal-Join* operation to be successful is that there exist two "*joinable*" concepts.

In abduction based on the *Maximal-Join* operation, the resultant graphs can be seen as hypothetical in nature. Additionally, however, constraints stemming from canonicity and conformity increase the likelihood of successful (plausible) inference.

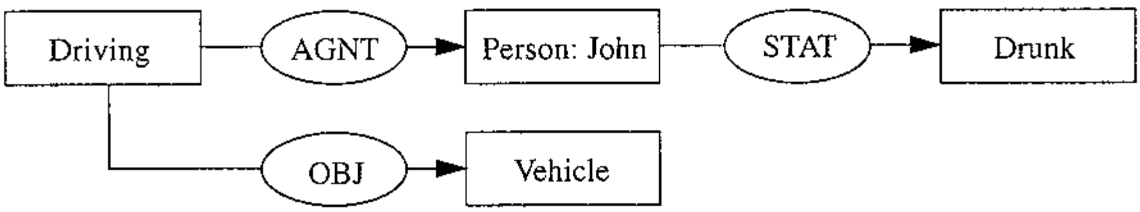
In the above notation, the pragmatics of the problem are significant, because there are too many explanatory graphs that could be chosen. It is a difficult job to select an appropriate subset of graphs and, furthermore, to restrict the number of conceptual graphs for abduction. Unfortunately, there is no solution that is totally satisfactory for the evaluation of selected graphs.

5.3.2.1 Algorithm for Maximal-Join

In Tsui's research[156], two algorithms for plausible reasoning with conceptual graphs are investigated: the *Join* and the *Maximal-join* operations. The Join operation is the simpler of the two, and it refers to the integration of two graphs that share a common concept. Two concepts are common, if and only if, they have identical type labels and identical referents. For example, consider the following graphs:

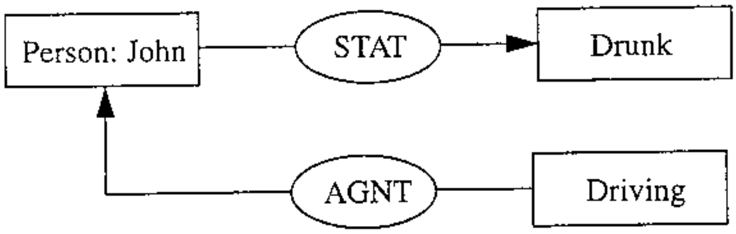


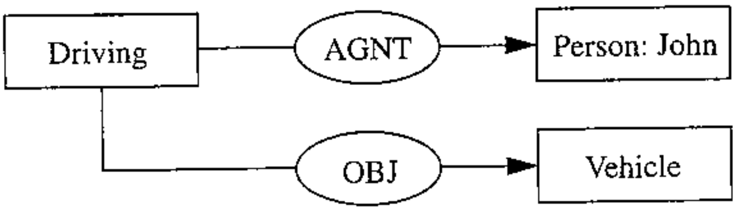
The resultant graph of a Join operation on the common concept [PERSON:John] of the above graphs is the following graph.



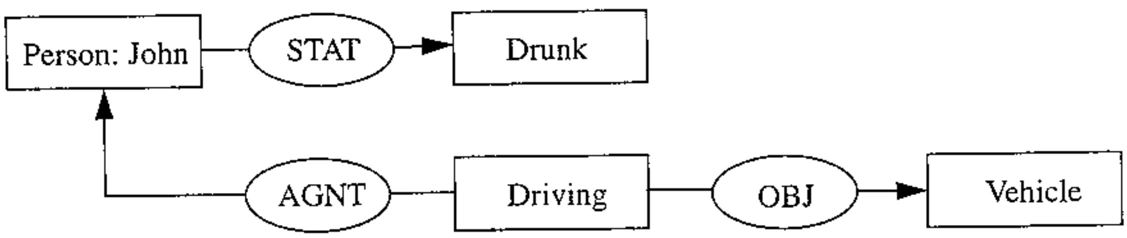
However, the *Join* operation creates a problem when the graphs to be joined have more than one concept that can be merged. In this case, if one simply Joins the graphs on a common concept, then there exist duplicate relations and concepts in the resultant graph. As a result, Join cannot be used on graphs that share more than one concept in common. The *Maximal-join* operation is used in such situations.

The *Maximal-join* algorithm between two graphs essentially provides abductive inference. *Maximal-join* ensures maximum connectivity in the resultant graph, merging as many concepts between the two original graphs as possible. For example, consider the following graphs:





The resultant graph of a *Maximal-Join* operation on the above graphs is the following.



Tsui implemented his algorithm in MU-Prolog, and it proved to be very successful. Currently, some researchers are working on other algorithms[68][105].

For my purposes, it was necessary to extend the definition of "*joinable*" concepts in order to perform abduction more flexibly. In our *Maximal-join* algorithm, the concept "*joinable*" is looser! The two concepts u and v are still "*joinable*" if they meet one of the following conditions:

- *u* and *v* have a common supertype,
- *u* is the subtype of *v*, or
- *v* is the subtype of *u*.

In essence, the resultant graphs produced by *Maximal-join* can be seen as abductive inference from the facts and those definitions, causal or Aristotelian, that cover them [68]. The result is hypothetical in nature because the maximal common subtype restriction of two types leads to the same unsound inference rule to that derived from logical abduction.

5.3.2.2 The Role of Hypothesis Space in Abduction

In this research, all potential explanatory graphs have been represented as explanatory relations which are used to connect two different attributes. This theoretical notion permits any two graphs sharing a common concept to participate in maximal-join operations. However, if an evidence graph is imported into the *maximal-join* operation the ensuing number of resultant graphs will be large. Strategies for restricting the number of graphs to which *maximal-join* is applied are essential.

Before we discuss the strategies to implement inference control for *Maximal-Join* based abductive reasoning, there are three questions to be answered, while noting that the hypothesis generation model is based on hypothesis space:

1. how to restrict the explanatory graphs;
2. how to select suitable explanatory graphs; and
3. how to measure the relevance between evidence and its explanation.

It has been noted that the specific *desires* for knowledge have a clear role in the focus of attention during natural language processing, and in directing machine learning programs [131]. We believe that the method of restricting realm of experience and background knowledge for inference should be context-based. The dynamic hypothesis space has an innate advantage in providing an appropriate set of explanation graphs.

The explanatory relations in hypothesis space are used to index the explanatory graphs. My research indicates that restricting hypothesis space is a very effective way to reduce the candidate explanatory graphs. This is achieved using the feature of dynamic hypothesis space that requires only part of the space to be active.

In assuming that a set of evidence will require a set of related explanations, it is quite likely that explanations that are currently active in hypothesis space will also be relevant to explanations of successive evidence to that causing the initial activation. As such, it is appropriate to consider these active explanatory graphs as prime candidates for the abduction process with the new evidence.

As discussed in Chapter Four, several approaches have been researched for hypothesis space restriction. They are Parsimony Covering Theory and Forward-backward Propagation based approaches. The experimental results indicate that active hypothesis space generated by the parsimony covering theory based approach is smaller than that from the forward-backward propagation based approach. In abductive inference, the number of candidate explanatory graphs restricted by the latter approach is larger. It is difficult to tell however, which approach is preferred in terms of the semantic requirements of explanations.

5.4 Example of Abductive Inference

In this section, an example of abductive inference is used to generate a "suitable" explanation. Supposing that the evidences detected by *Anomaly Detection Model* are "*Accident happens at 11 pm*", "*Driver has only soft tissue injury*", "*Repair cost is low*" and "*Claimant has claim history*". FIGURE 5.1 shows the evidence graph representing the above evidence (facts):

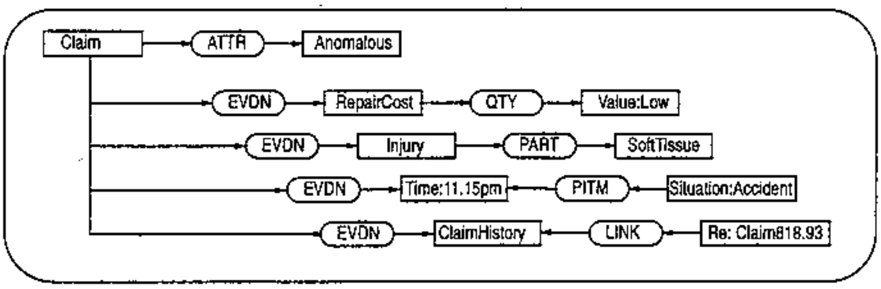


FIGURE 5.1 The Evidence Graph

The first step of abductive reasoning involves decomposing the evidence graph into four separate graphs, which are used to activate the evidence nodes in hypothesis space. The result is a limited hypothesis space, as shown in FIGURE 5.2, merged through the activation of evidence derived from the evidence graph. The hypothesis, with the conclusion of "*Exaggeration of Economic Loss*" and "*Staged Accident*" are associated automatically in the limited hypothesis space. At the same time, it should be noticed that there is no matching of evidence for the "*Repair cost is low*".

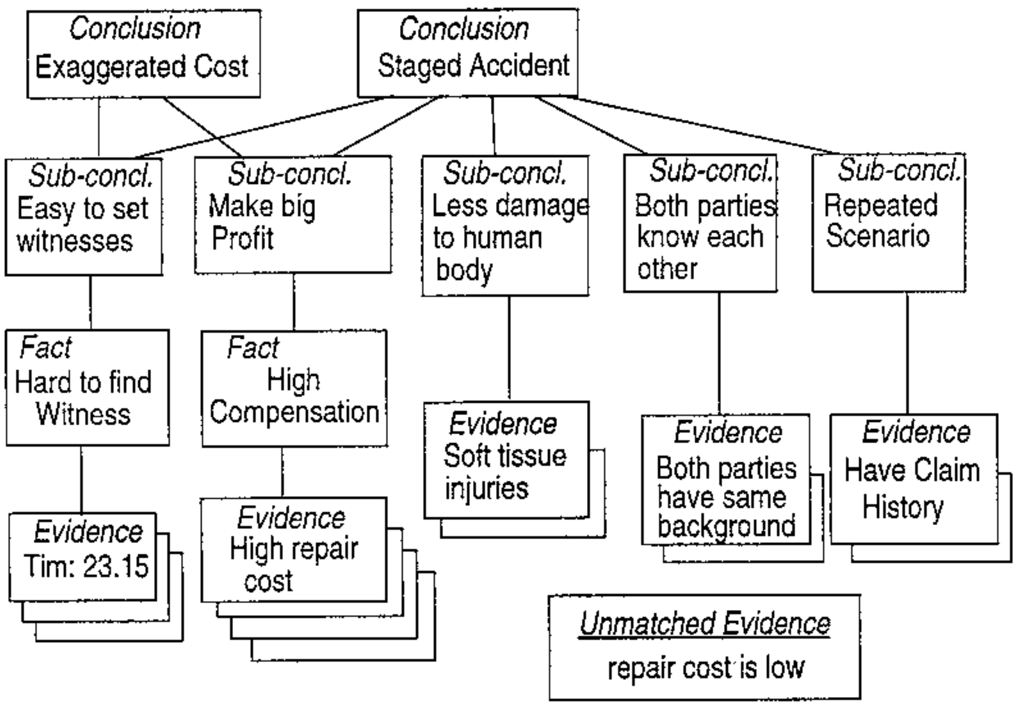


FIGURE 5.2 Activated Hypothesis Space (Abstracted)

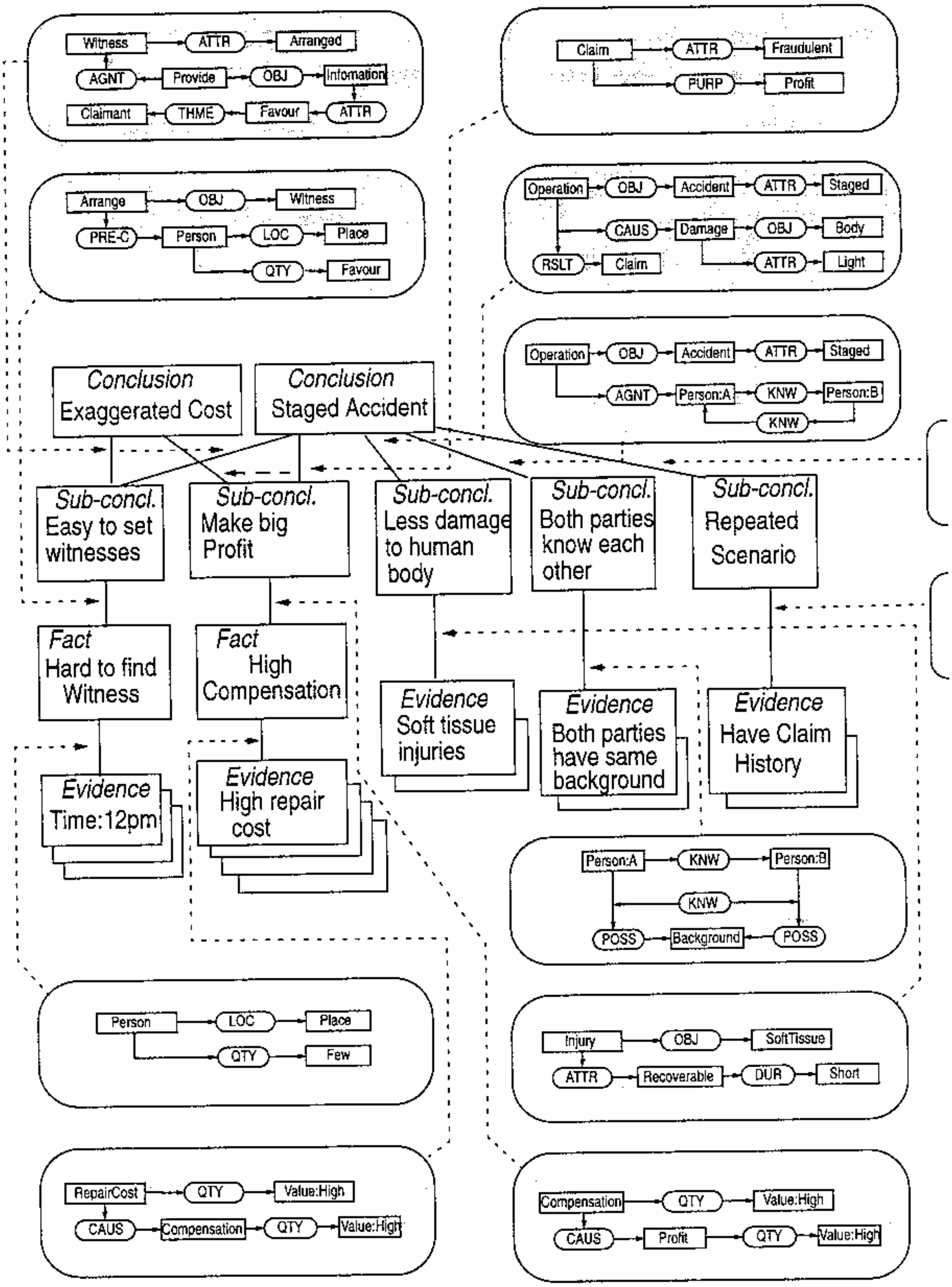


FIGURE 5.3 Hypothesis Space with Explanatory Relations

In FIGURE 5.3, the detailed hypothesis space with explanation relations is displayed, and is equivalent to the hypothesis space in FIGURE 5.2. By utilizing this activated hypothesis space, the number of explanations which potentially could be used in abductive inference is greatly reduced to twelve (ten relations are shown). In the twelve restricted explanation graphs, only three graphs, which have at least one concept *"joinable"* to unmatched evidence, will actually be used for the abductive inference process. FIGURE 5.4 displays these three candidate explanation graphs.

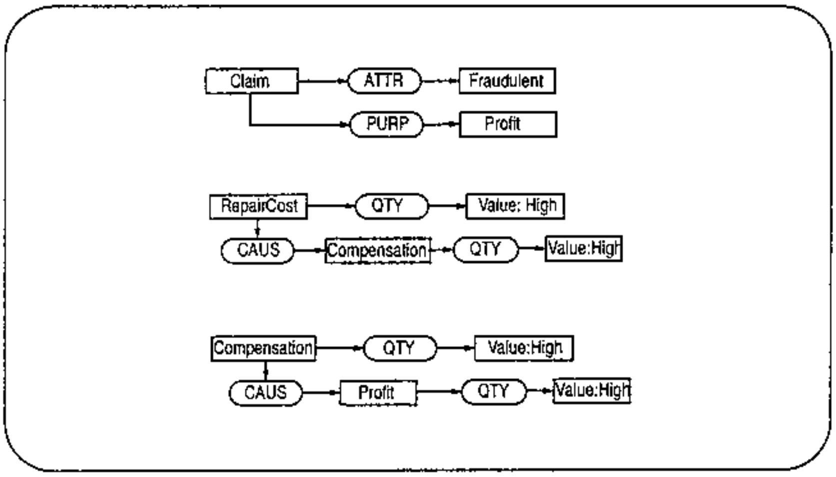


FIGURE 5.4 Candidate Explanation Graphs

The new evidence graph which is separated from matched evidence graphs is shown in FIGURE 5.5. The abductive inference engine will *'maximal-join'* it with three candidate explanation graphs respectively. The resultant graphs generated from the *maximal-join* operation for the new evidence *"Repair cost is low"* are shown in FIGURE 5.6.

As we know, one of the weaknesses of abduction is redundancy in its resolutions. It is necessary to assess these three graphs and choose a "best" one. This is carried out by the explanation evaluation and conflict resolution process, which are discussed in the next chapter (6). Following explanation evaluation and conflict resolution, the final explanation for the new evidence is shown in FIGURE 5.7; namely: 'the low repair cost results in low compensation.

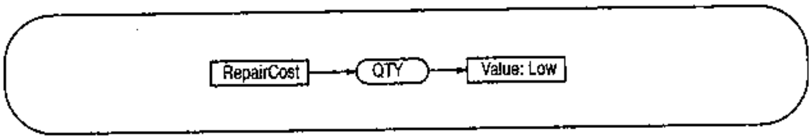


FIGURE 5.5 New Evidence Graph

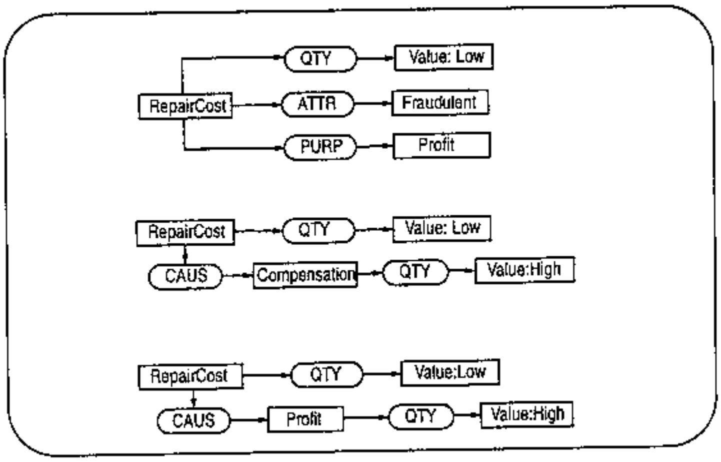


FIGURE 5.6 Resultant Graphs of Abductive Inference

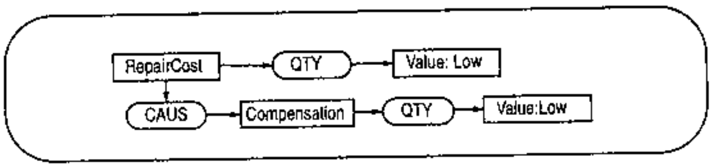


FIGURE 5.7 Final Explanation Graph for the New Evidence

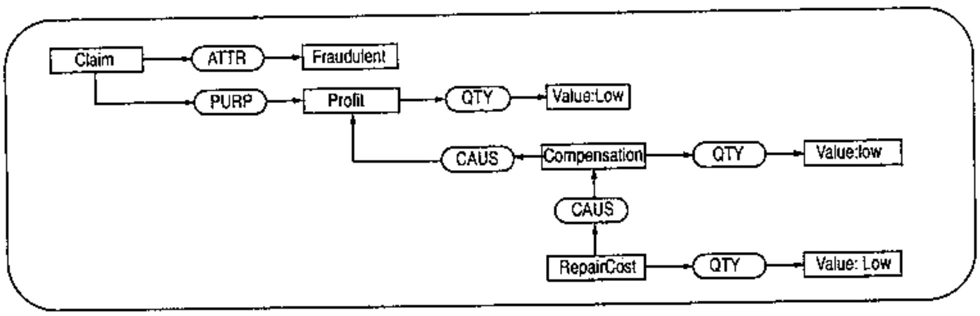


FIGURE 5.8 Combined Explanation Graph

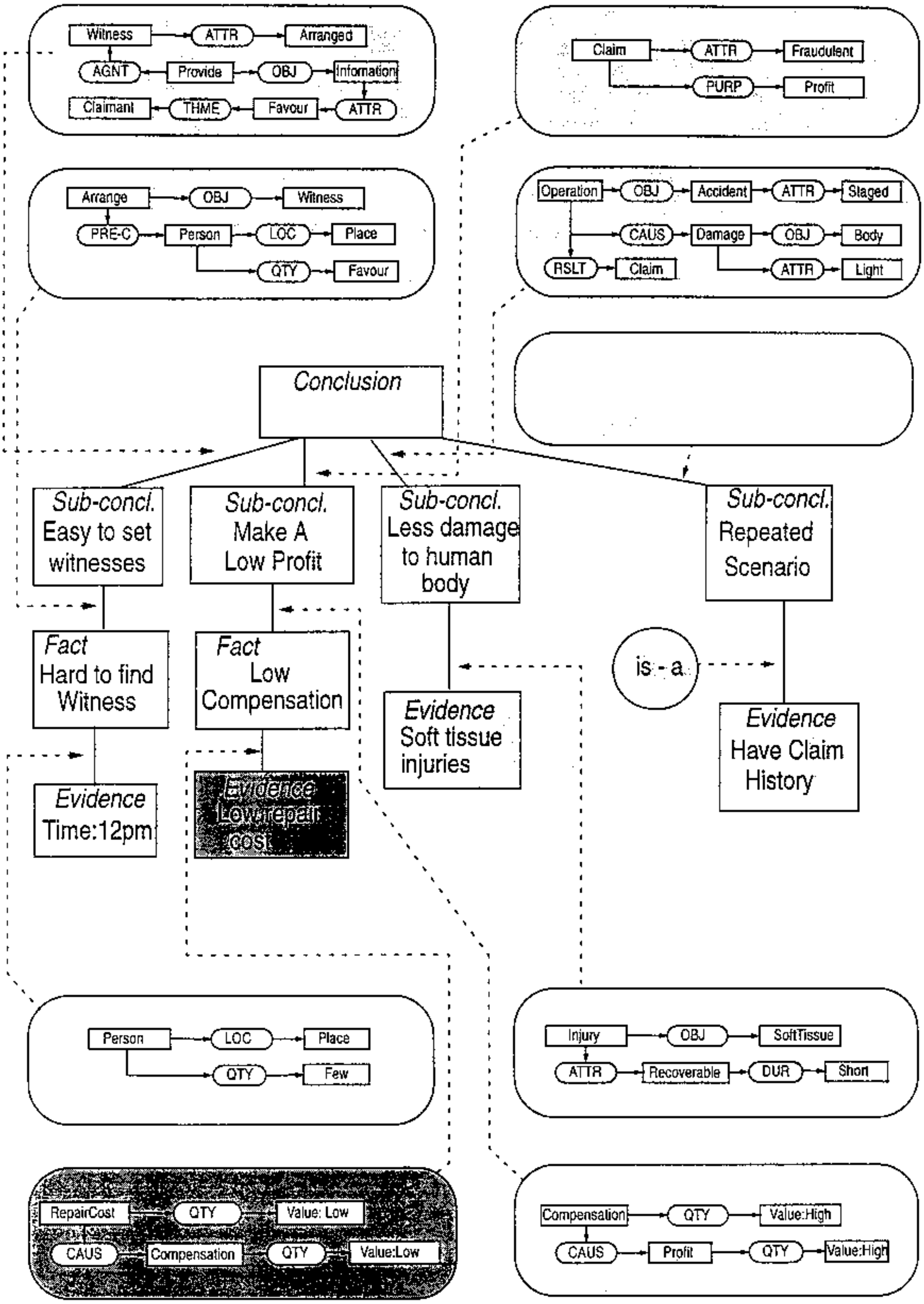


FIGURE 5.9 Candidate Hypothesis

The final explanation graph, which explains the relation between the new evidence and conclusion, is generated by a *max-join* operation. The label *high* of concept *value*, which is connected from *profit* by *QTY* is substituted by *low* with the strategy used in local conflict resolution. This method is actually an equivalent to the *Twist* technique introduced by Leake and Schank[145].

The conceptual graph displayed in FIGURE 5.8 is the result of *Maximal-join* of the explanatory graph from the evidence, facts, sub-conclusion and conclusion.

The current state of the hypothesis, which is still under development, is shown in FIGURE 5.9. Because the evidence "*two parties involved in accident know each other*" cannot be verified, the conclusion of *staged accident* is abandoned. This hypothesis is reasonable in logic, but it did not have a suitably strong conclusion to support it.

The hypothesis space will continue to extend through the incorporation of the new evidence nodes and new explanatory nodes. As the result of this extension, it is obvious that the old conclusion is no longer suitable, and a new conclusion is necessary. There are theoretically three types of methods for providing a conclusion. They are based on extending hypothesis space, analyzing the knowledge in rule bases and eliciting the knowledge from users. The last method is based on the principle of hypothesis based interaction and will be discussed later.

To extend the hypothesis space, a broader set of explanations will be used for abduction. We still use the unmatched evidence graph, as shown in FIGURE 5.5, as an index to activate hypothesis space. FIGURE 5.10 displays the *Extended Hypothesis Space* with two relevant explanations.

In the extended hypothesis space, two relevant explanations are activated, in which concepts share a common supertype with concepts in the unmatched evidence graph. This first one connects to the evidence "*photocopies from altered document*" and another to "*repairing existing damage*". Both of them involve further investigation for the purpose of "*check original document*" and "*check damage consistency*".

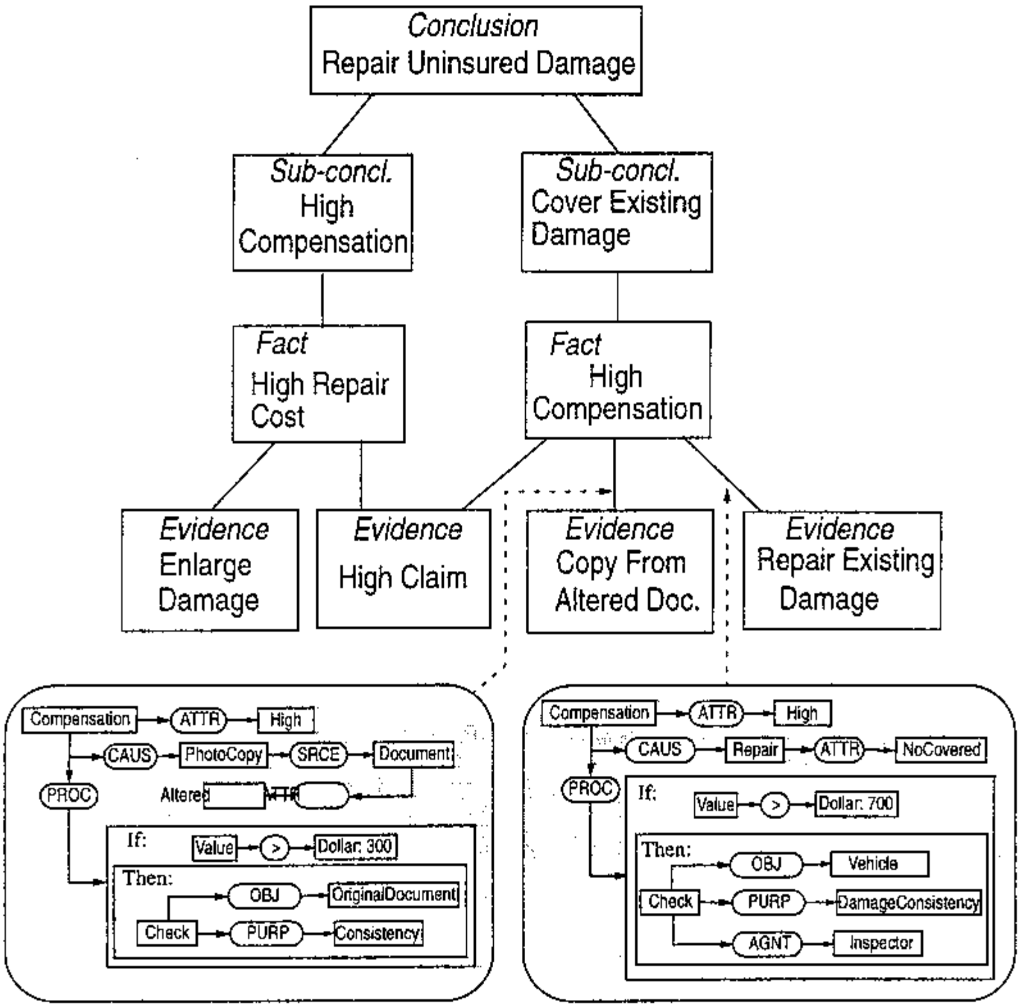


FIGURE 5.10 The Extended Hypothesis Space

As an example, consider the second explanation "*check damage consistency*". The resultant graph from *Maximal-join* operation shows that "*Repair cost is low*" will not cause "*Check damage consistency*" by an inspector. Based on such information, the explanation: "*Low repair cost will avoid damage consistency check by inspector*" will naturally follow. Although a fruitful outcome (conclusion) from such a process cannot be guaranteed, the hint provided by the machine does provide guidance for user responses. At the end of the dialogue between user and system, the new explanatory graph is formed after a conflict resolution process, such as simplification, deletion, or etc.

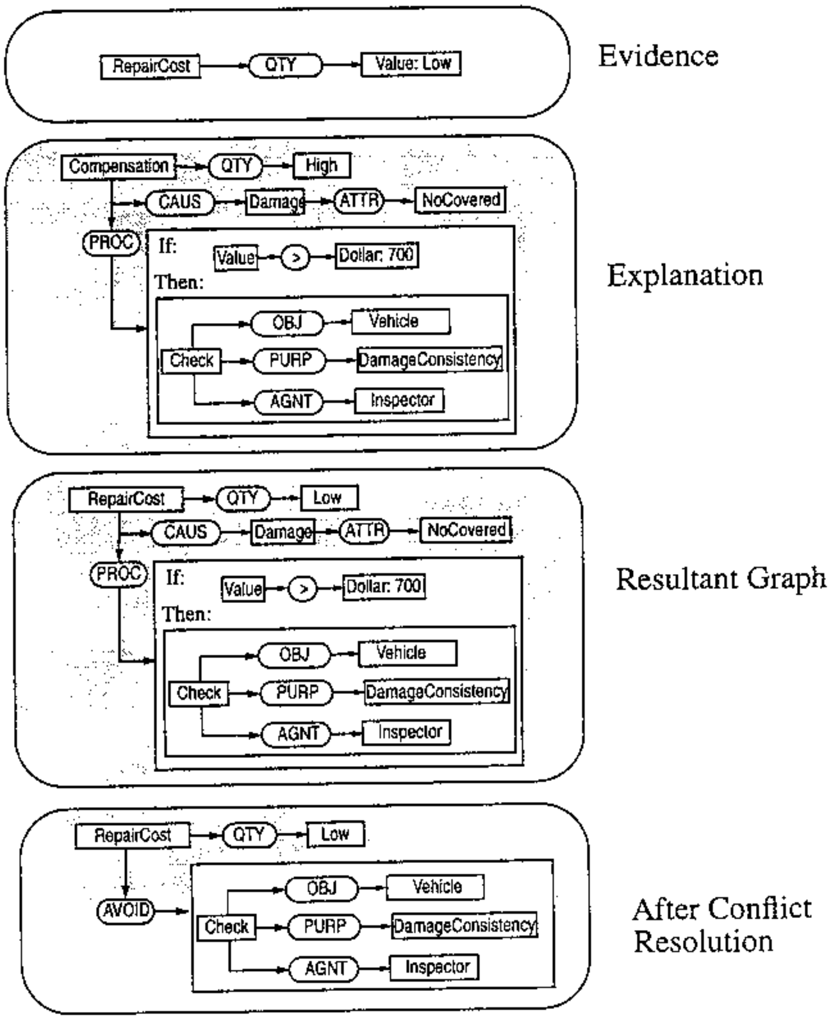


FIGURE 5.11 Another Explanation Generated

FIGURE 5.11 displays the process of generating a new explanation from the resultant graph by *max-join*. We incorporate the new explanation into the candidate hypothesis (FIGURE 5.12). The Sub-conclusion "*Cover Existing Damage*" (see FIGURE 5.10) will be adopted as a conclusion in the hypothesis shown in FIGURE 5.12. The final consistency check for the conclusion is usually performed by human experts. At this stage, the user can overwrite the system's conclusion, and input a statement, which usually is more meaningful in a fraud context. However, note that the explanation process has been packaged for future use by the machine.

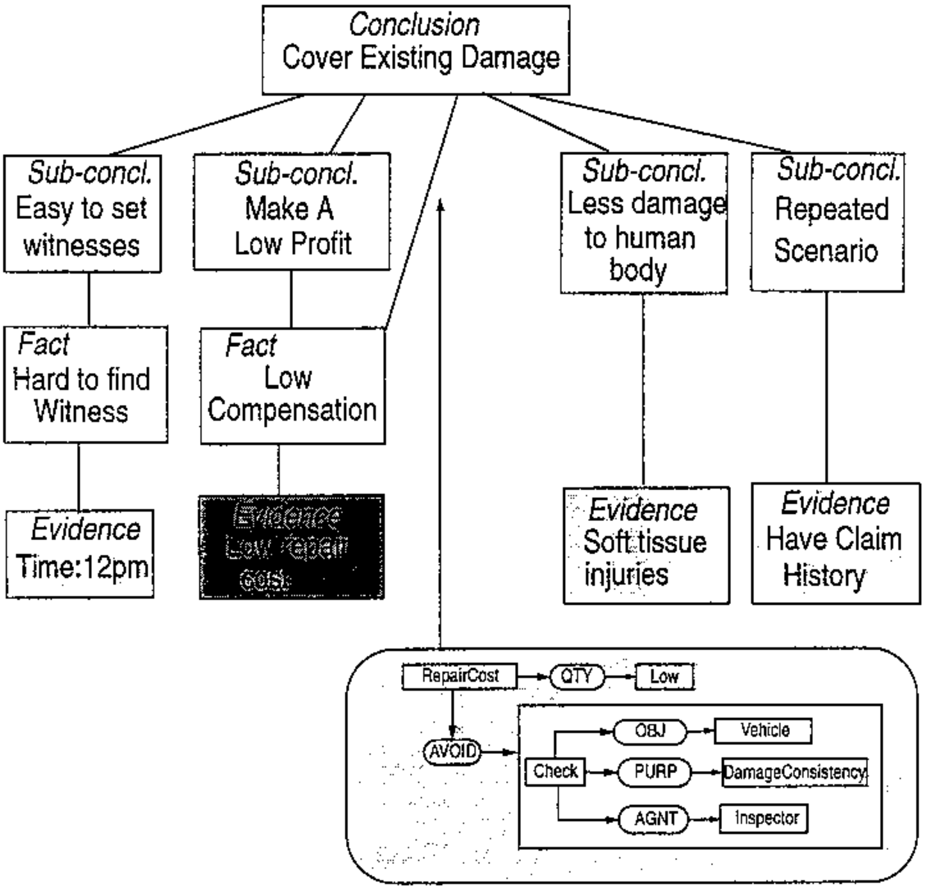


FIGURE 5.12 Hypothesis under Development

In FIGURE 5.13, we present a final version of the new hypothesis, formed after simplification using the knowledge fusion process.

In this example of abductive inference, originality has been demonstrated in hypothesis space activation, explanatory relation selection, maximal-join operation, hypothesis space extension, conflict resolution, explanation evaluation, and knowledge fusion. Naturally, the human expert should be able to overwrite the system's conclusions and explanations at every stage of the intelligence acquisition process.

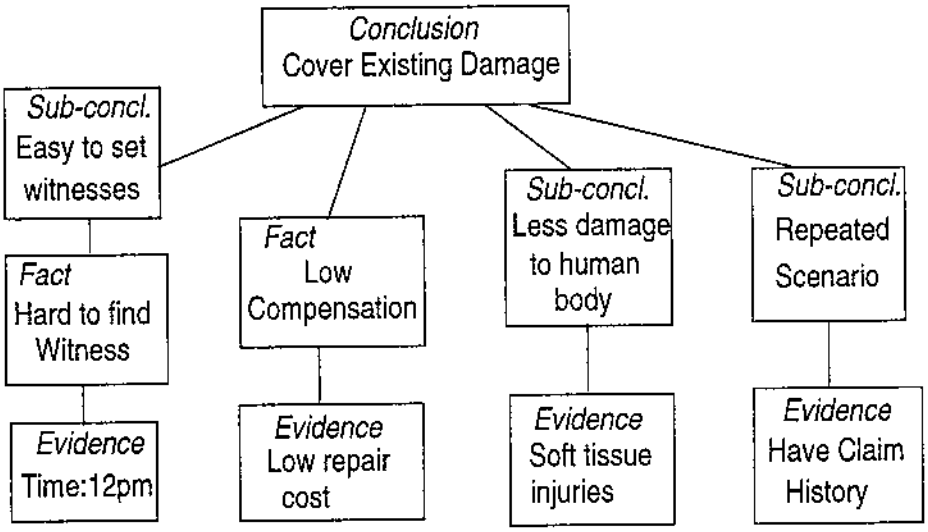


FIGURE 5.13 A New Hypothesis Generated

5.5 Summary

The abductive inference is a weak mechanism for generating explanation and hypothesis. A practical conclusion cannot be guaranteed, and sometimes, results will conflict with other existing explanations. Strategies for conflict resolution and hypothesis evaluation will be addressed in the next chapter.

The research indicates that inspiration provided by abductive inference is sometimes more beneficial than a complete hypothesis. Inevitably, real world knowledge and suspicions voiced by other parties may require modification of machine intelligence results. However, there is inestimable value in fully exploiting the formalised facts by machine and in explaining how a result was arrived at.

While abductive inference has not been able to produce a complete hypothesis in some situations, the inspiration provided by the machine is very attractive for complex problem solving and for the associated human computer cooperation.

CHAPTER 6

Conflict Resolution and Hypothesis Evaluation

6.1 Introduction

In artificial intelligence, the problem of evaluating explanations has been investigated in three main areas[88]. Research into explanations provided by expert systems has concentrated on the question of the explanation's goodness for explaining system behavior, for the benefit of the system users[25]; research into explanations for story understanding has concentrated on how to select valid explanations from a range of hypotheses[145]; and research into explanation based learning has been primarily concerned with the problem of determining an explanation' goodness for improving performance of concept recognition and search[36].

Many theories of explanation evaluation are logically based on context independent criteria. Such theories either restrict their consideration of explanation to fixed goals, or assume that all valid explanations are equivalent, in which case evaluation criteria would be neutral to the hypotheses underlying the explanation task. Context-dependent criteria evaluate hypotheses from the point of view of domain tasks that the system is trying to solve. A fraud detective, for example, might need to build very different explanations for such facts as "the driver had decamped from the scene of accident" to the common hypothesis of "Joy driving"?

This chapter addresses the problem of evaluating the "goodness" of explanation and hypotheses in the context of a working domain. The focus on explanatory hypotheses (causal chains) attempts to verify a given anomalous fact in terms of causal relations within the underlying hypotheses.

6.2 Conflicts in Hypothesis

An hypothesis consists of a set of attributes which is connected by relations. For the reasons given before, the relations, relevant to conflict resolution, are explanatory ones, which explain the relations between the attributes. One of the major problems produced by the hypothesis generation model, more specifically by abductive inference, is the existence of redundant and contradictory explanations. The origin of conflict in hypothesis is determined by the nature of abduction, which provides "plausible" explanations for the given evidence. In insurance fraud detection domain, if the evidence is "*Driver has no obvious injuries*", the two explanations: "*Accident in which the person is involved is not serious*" and "*Person is not in car when accident happened*" are contradictory and only one can be valid when the hypothesis is verified.

The contradictory explanations, which are commonly called conflicts, can often be modified or removed by further imposing some constraints to the hypothesis. A single candidate hypothesis may contain different kinds of conflicts, and each conflict, in turn, may be solved by a particular set of constraints.

The process of hypothesis generation consists of three steps: anomaly detection, hypothesis generation, and hypothesis verification. No matter which way the hypotheses are generated, evaluation of the derived hypotheses is crucial. Ram and Leake categorize evaluation criteria into two categories: syntax-based and task-based criteria[129]. Syntax-based criteria rely on structural or syntactic properties of the causal chain to evaluate hypotheses. A "goodness" measure for each hypothesis can be computed based on the length of the causal chain, the number of abductive assumptions, or other structural properties [74][82][129]. Task-based criteria select hypotheses according to requirements arising either from the system's intended use, or from an explanation, such as those required for predictions about future events[129].

Conflict resolution in hypothesis generation is a complex computational process. In this chapter, we will introduce a measure for explanation and hypothesis. Our method can be used to evaluate the "quality" of explanation and hypothesis, and could also be used to detect indirectly, possible conflicts in a set of explanations. The abductive inference engine and hypothesis generation model are, therefore, consistent with the use of such measures, elaborated below.

6.3 Semantic Distance of Concepts

The notion of semantic distance as a useful measure of concept similarity has been exploited by Tsui [156]. Firstly, however, the notion of *type hierarchy* is introduced as a partial ordering defined over the set of concept labels [151]. Consider the concept type vehicle. The immediate *subtype* of *vehicle* includes *ambulance, automobile, bicycle, buck-board, bus, carriage, cart*, etc. Immediate subtypes of *automobile* would be *coach, con-vertible, coupe, hot-rod, jalopy, sedan*, etc. Immediate *subtypes* of *sedan* include *brougham, limousine* and *saloon.*

Graphically, the above type hierarchy is shown in FIGURE 6.1

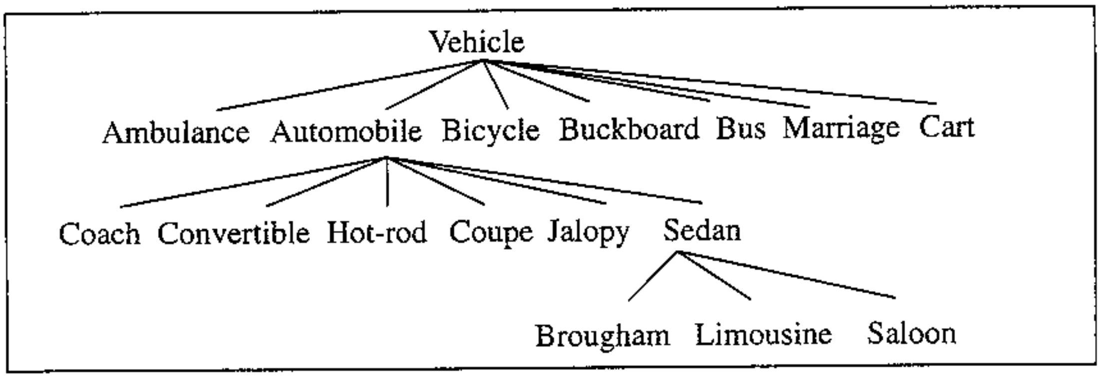


FIGURE 6.1 Concept Type Hierarchy for Type Vehicle (adapted from[156])

The *semantic distance* (SD) function is designed to correlate the relevancy of two valid concept labels[47][151], and represents the similarity of two concept labels. The link between two adjacent type labels in the type hierarchy is assigned an arbitrary quantitative value called the "*semantic distance between two type labels*"[156]. For any concept labels p and q, the semantic distance between them, *SD(p,q)*, may be defined as the number of arcs in the shortest path through the type lattice from p to q. For example:

$SD(convertible, saloon) = 3$,

$SD(sedan, vehicle) = 2$,

$SD(coupe, automobile) = 1$, and

$SD(saloon, cart) = 4$.

It may be inferred that two concepts which have the smallest semantic distance (*SD*) are most relevant and "similar". It should also be noted in support of the *SD* notion that semantic distance has played an important role in the *Semantic Interpreter*, a memory based natural language parser[57].

6.4 Difference Measurement between Conceptual Graphs

Semantic distance can be utilized to assist in calculating the difference between two conceptual graphs. Wuwongse and Niyomthai use matching score, which is defined as the summation of *Importance Values* [165] minus *Semantic Distance* of each concept in their minimal common generalization. The importance value is a score for each concept type to denote its relevance to the essence of the graph in which it is contained. It can be fixed or redefined with each concept type according to the domain.

Other proposals for matching conceptual graphs are based on *Ten Position Vector* [107] which is used in matching conceptual graphs and is useful in classification and comparison, common generalization[101], and specific algorithms[105].

The algorithm proposed by Wuwongse and Niyomthai sets important values manually and pays no attention to similarities in macro-structure of conceptual graphs. The ten position vector algorithm is also too time consuming for efficient human computer interaction.

Definition of Graph Difference-1:

Given two conceptual graphs U and V, such that $u_i \in U$ ($0 < i \leq n$), $v_j \in V$ ($0 < j \leq m$) and $n \leq m$, then the difference (GD) of two conceptual graphs is defined as:

$$GD = \sum_{i=1}^{n} SD(u_i, v_j)$$

$$j = i \; (j \leq n)$$

The difference measurement is an extension of the semantic distance, and is the sum of the semantic distance of corresponding concepts in two conceptual graphs. If the difference is zero it means two graphs are identical while a difference of infinity means they belong to different domains.

The *Definition-1* is simple to use and is efficient in computation. But, there are two limitations to this definition; namely the difference in relations and the need to specify corresponding concepts. This definition is effective only when two graphs have a similar structure. In our experiment, all evidences or attributes have similar structures and the *Definition-1* is shown to be effective and efficient.

Definition of minimal common generation:

Let conceptual graphs *u* and *v* have a common generalization *w*. *w* is said to be a ***minimal common generalization*** of *u* and *v* if the following conditions are true:

- for each concept *c* in *w*, it is a minimal common super type of the corresponding concepts in *u* and *v*,
- *w* is maximally extended (i.e. no further extension is possible).

Definition of Graph Difference-2:

Given two conceptual graphs *u* and *v* that have *w* as their minimal common generalization, the graph difference (*GD*), of *u* and *v* is defined by the following formula:

$$GD = N_u + N_v - 2N_w + \sum_{i=1}^{n} q_i \times SD(u_i, v_i)$$

Note:

N_u is the number of concepts in *u*,

N_v is the number of concepts in *v*,

N_w is the number of concept in *w*,

q_i is the weight of the corresponding concept in the conceptual graph,

$SD\ (u_i, v_i)$ is the semantic distance between u_i and v_i.

The minimal *difference* value is reached when one conceptual graph is identical with another graph or is a subgraph of another graph.

The Definition-2 is a more general method for measuring the *Difference* for graphs. This definition also embraces the different structures that may exist, and can assign a different weight to the concepts in graphs. By this method, the key concepts play an important role in graph matching, and some concepts will be relatively unimportant.

In our model, hypothesis space is activated by matching the graphs with evidences detected and with evidence nodes in the hypothesis space.

In the process of hypothesis space activation, we can treat the problem of graph matching as:

> Given evidence represented as a conceptual graph u, given an hypothesis space HS, and given v_i is the evidence node in V of the hypothesis space HS ($v_i \in V$, and $V \subset HS$), *then* the activated evidence node v_i will have the minimal difference D with u, and $D > d$ (the threshold value).

$$D = Min\ \{\ SD\ (\ v_i\ ,\ u)\ \}$$
$$(v_i \in V)$$

If, $D \geq d$ then the evidence node v_i is activated by u.

Semantic distance based measurement is computationally expensive. A good method for deciding weights for attributes may be hard to define. The attractiveness of this approach lies in the capability to deal with inexact matches of features and in the representation of '*fuzzy*' concepts and situational knowledge.

6.5 Credibility Measurement

In our hypothesis generation model, the candidate explanations are the results of an abductive reasoning process. In order to measure the quality of explanations, we introduce the concept of *credibility*. The purpose of a *credibility* measurement is to select the "most suitable" explanation. The degree of conflict identified with any given (generated) explanation is an inverse measure of its likely credibility.

Conflicts may be divided into different categories according to the different feature classifications. Our principal concern is with two kinds of conflicts according to their local or global significance.

1. Explanation conflicts (local conflicts)
   Conflicts occur when explanations are activated by same evidence. This kind of conflict happens in the process of abductive reasoning, and is due to the set of candidate graphs selected. The strategy for explanation conflict resolution is to select that one which "best" explains the evidence under certain selection criteria.

2. Hypothesis conflicts (global conflicts)
   After the process of hypothesis generation, there may still be some conflicts among the candidate hypotheses. Such conflicts are caused by introduction of a new explanation to the hypothesis space. It is necessary to identify and resolve such conflicts (between explanations) and to redefine the hypothesis, if necessary.

A problem of implementation often arises, however, as the boundary between conflict resolution and hypothesis evaluation is often indistinct. Minor modifications may be necessary to the methods used to evaluate hypotheses.

6.5.1 Explanation Credibility

To measure the credibility for an explanation in relation to particular evidence, we use the notation:

$$p' = p_0'(explanation,\ evidence)$$

In an activated hypothesis space, an explanatory relation is activated by at least one attribute. The credibility for an existing explanation derived from a connected evidence node is very high, because conflicts are resolved during the generation process.

To explain new evidence, there are two ways; namely use of an existing explanation, or generation of a new explanation.

The computational cost is very high in calculation of the credibility between explanation and evidence directly and even worse in conflict resolution. In the first situation, we can avoid the difficulty by calculating the difference between attributes in hypothesis space and the evidence discovered. Therefore, we can simplify the credibility computation through measurement of the difference between the evidence at hand and the existing attributes, because the value of credibility p is inversely proportional to that of graph difference GD.

$$p' = p'(attribute, evidence) = \frac{c_1}{GD(g_1, g_2)}$$

where g_1 is the evidential graph nested in *evidence-node*, g_2 is the conceptual graph of new evidence and c_1 is a constant.

According to this definition, the difference GD ranges from zero to infinity$[0, +\infty)$ and the credibility will be range from infinity to zero $(+\infty, 0]$. For the purposes of standardization, it is more sensible to normalise the range between $[0, 1]$. The following formula will fulfil this mapping:

$$p = \frac{p'}{p' + c_2} = \frac{c_1}{c_1 + c_2 GD(g_1, g_2)}$$

where c_2 is also a constant, and therefore:

$$p = \frac{b}{b + GD(g_1, g_2)}$$

Here $b = c_1/c_2$. The definition can be simply represented as:

$$p = \frac{b}{b + GD}$$

Under this definition, the credibility p is one (1) if two evidential graphs are identical, and zero (0) if two graphs are totally different (This may be interpreted as two graphs being in different domains).

In a similar way, explanation *un-acceptability (up)* may be defined as $(1 - p)$:

$$up = \left(\frac{GD}{b + GD}\right)$$

The un-acceptability is zero (0) if the difference is zero (0) and is one (1) where the difference is infinity ($+\infty$).

This explanation's credibility and the related measure of un-acceptability, are derived from the evidences-node and from the evidence at hand. It is only suitable for measuring existing explanations. For measuring the credibility of a new explanation, we have developed a method which can be classified as task based evaluation, which will be discussed later.

6.5.2 Hypothesis Credibility

The measurement of hypothesis credibility is made by extending individual explanation credibilities. The simple way is to calculate every explanation's credibility separately, and then compute the hypothesis' credibility as the sum:

$$P = \frac{1}{n}\sum_{i=1}^{n} p_i$$

In this formula, n is the number of explanations connected to the evidence nodes. Although this approach is very simple in concept, it has two disadvantages. One is the high computational burden. Another is the lack of consideration of conflicts between the explanations.

To overcome such limitations, the credibility is calculated after abductive operation. This method reduces computation significantly, but at the cost of decreasing accuracy. In determining hypothesis credibility, we are concerned about conflicts among all the explanations indexed by all evidences. We define the hypothesis credibility as P:

$$P = P_0(Explanation, Evidence) = P_0(\Sigma explanation_i, \Sigma evidence_j)$$

The *Explanation* represents all explanations in the hypothesis and the *Evidence* represents all evidence to hand. The Σ operation can be implemented by the *Maximal-Join* operator [151][156], and, therefore

$$P_0(Explanation, Evidence) = P(G_1, G_2)$$

where: G_1 = Max-join($IDX\text{-}evidence_1$, $IDX\text{-}evidence_2$,...,$IDX\text{-}evidence_j$), and

G_2 = Max-join($Evidence_1$, $Evidence_2$,...,$Evidence_i$).

As with explanation credibility, we can compute the *hypothesis credibility* as:

$$P = \frac{a}{a + GD(G_1, G_2)}$$

In a similar way, *hypothesis un-acceptability* may be defined as:

$$UP = \frac{GD(G_1, G_2)}{a + GD(G_1, G_2)}$$

Because the hypothesis credibility and un-acceptability are measured indirectly from conceptual graph differences, it is a kind of possibility measure.

If the credibility P as calculated exceeds the threshold of credibility D, some conflicts may be presumed to exist. A trace-back of the computation process reveals a way to identify the location of the conflicts through identification of the concepts having a semantic distance over the threshold.

Semantic distance (SD) based criteria provide a relatively easy way to evaluate the "goodness" of an hypothesis. However, such criteria are sometimes not very desirable in conflict resolution with newly generated explanations. Goal based criteria (or task based) are then directly applicable to explanations, without the help of evidence nodes. Whereas the SD method relies on domain knowledge such as inference rules, cases, schemas, or other types of knowledge, goal based criteria take a strongly context-dependent view of evaluation of hypotheses. The final determinant of an explanation's "goodness" is whether it informs the reasoning process.

6.6 Conflict Resolution Methods

Conflicts threaten the reliability of hypotheses. As stated earlier, both local conflicts and global conflicts may be encountered, requiring different conflict resolution strategies.

When an hypothesis is generated to explain anomalous evidence, it is necessary to evaluate the hypothesis using criteria that are domain dependent:

1. Novelty:
   In fraud detection, we are more interested in a novel explanation instead of modifying old explanations to suit new evidence. Novelty provides a basis for judging whether a surprising event was itself caused by something surprising.

2. Simplicity:
   In most situations, the simplest explanation is often the best. This crite-

rion (Occam's razor) can also be used in hypothesis evaluation by focussing on the simplest structure and the simplest concept.

3. Specificity:
   The specified explanation is usually richer in knowledge when accumulated through the process of problem solving.

Although goal based criteria provide some indirect approaches to hypothesis evaluation, such considerations sometimes result in goal conflicts, for example, the criteria for simplicity and specificity. We cannot expect any one assessment technique to give a uniquely definitive result. The best that can be hoped for is that a number of different criteria (each incorporating different features) will give self-consistent results.

6.6.1 Explanation Conflict Resolution

The main point in explanation conflict resolution is that when there is a limited choice of explanations, increasing of the credibility of one explanation will decrease the credibility of the other explanation. Since the number of possible explanations involved in an hypothesis is finite, the absence of a new explanation strengthens the evidence for existing explanations. In order to implement these ideas, a *Priority Value* (PV) has been employed to select the most suitable one.

In this approach we extend the treatment of *Modified Belief Value* proposed by Zhang[167][168] for the purpose of explanation conflict resolution. Each hypothesis determines

1. the degree of *explanation* credibility p; and
2. the degree of *explanation un-acceptability up*

for each explanation. Modification of the degree of credibility for an explanation for the purpose of general applicability is obtained as follow:

$$ModifiedCredibility(MP) = p_1 + \sum_{i=2}^{n} \omega_i \cdot up_i \cdot (1 - p_1)$$

where p_I is the credibility of an explanation, up_i is the un-acceptability of all the explanations, and ω ($\Sigma\, \omega_i < 1$) is a weighting factor which is self-adjusted when the system is running. From this formula, we not only consider the credibility of an explanation but also the relationships to other explanations.

We determine a modified un-acceptability (MUP) of each explanation in a similar way.

$$MUP = up_1 + \sum_{i=2}^{n} \omega_i \cdot p_i \cdot (1 - up_1)$$

After an hypothesis has modified the credibility and un-acceptability value, the priority value (PV) of an explanation for given evidence is computed by using the combination function in the hypothesis generation model, and is determined by the formula as:

$$PV = MP - MUP$$

6.6.2 Hypothesis Conflict Resolution

When a set of explanations are indexed by separate evidence, some conflict may exist between the explanations. Resolution of these conflicts and refinement of the explanations is then necessary.

In order to detect conflicts in hypothesis, we calculate the *hypothesis* credibility P. If the P is below a certain threshold, minimal conflict is assumed and it is ignored. Otherwise, it would be necessary to examine every individual conflict.

To detect inconsistence (conflicts) in hypotheses, it is useful to measure the un-acceptability between one explanation and new evidence.

$$UP = UP_0(explanation, Evidence) = UP_0(explanation, \sum evidence)$$

$$UP = UP_0(g_1, G_2)$$

When an explanation with the maximal *UP (un-acceptability) is under consideration*, it is asserted the explanation is in conflict with others. A conflict resolution process will then be applied to that explanation.

It should be noted that no single solution exists to resolve all conflicts. The method provided here is a weak approach. Conflict resolution should be supported by a number of approaches, such as task based approaches.

6.7 Task Based Evaluation

What constitutes a good hypothesis? Before discussing task-based evaluation, we should classify this question. For our purposes, a good explanatory hypothesis should comprise a set of good explanations without conflict between them. However, in some domains (e.g. fraud detection) it is necessary to speculate regarding individual motives to discover the rich hypotheses and associated explanation. It may be concluded from our study that there is no sole criterion suitable for hypothesis evaluation in a complex domain. A sophisticated set of criteria must reflect the dynamics of the situation actually encountered.

The semantic distance measurement provides a basis for evaluating the explanation without considering (or discounting) other explanations, although it does provide a simple way to calculate credibility indirectly. On the other hand, the accuracy of this measure requires improvement due to the indirect measurement. To overcome such limitations, the knowledge based method has been proposed as a compensating strategy to measure the credibility of an hypothesis.

In knowledge based explanation evaluation, the system has a set of expectations in memory, such as a set of fraud schemes in our case. The method involves evaluating whether the proposed explanation "fits" a candidate fraud scheme. In our experiment, the fraud schemes are arranged in a fraud hierarchy as shown in FIGURE 6.2.

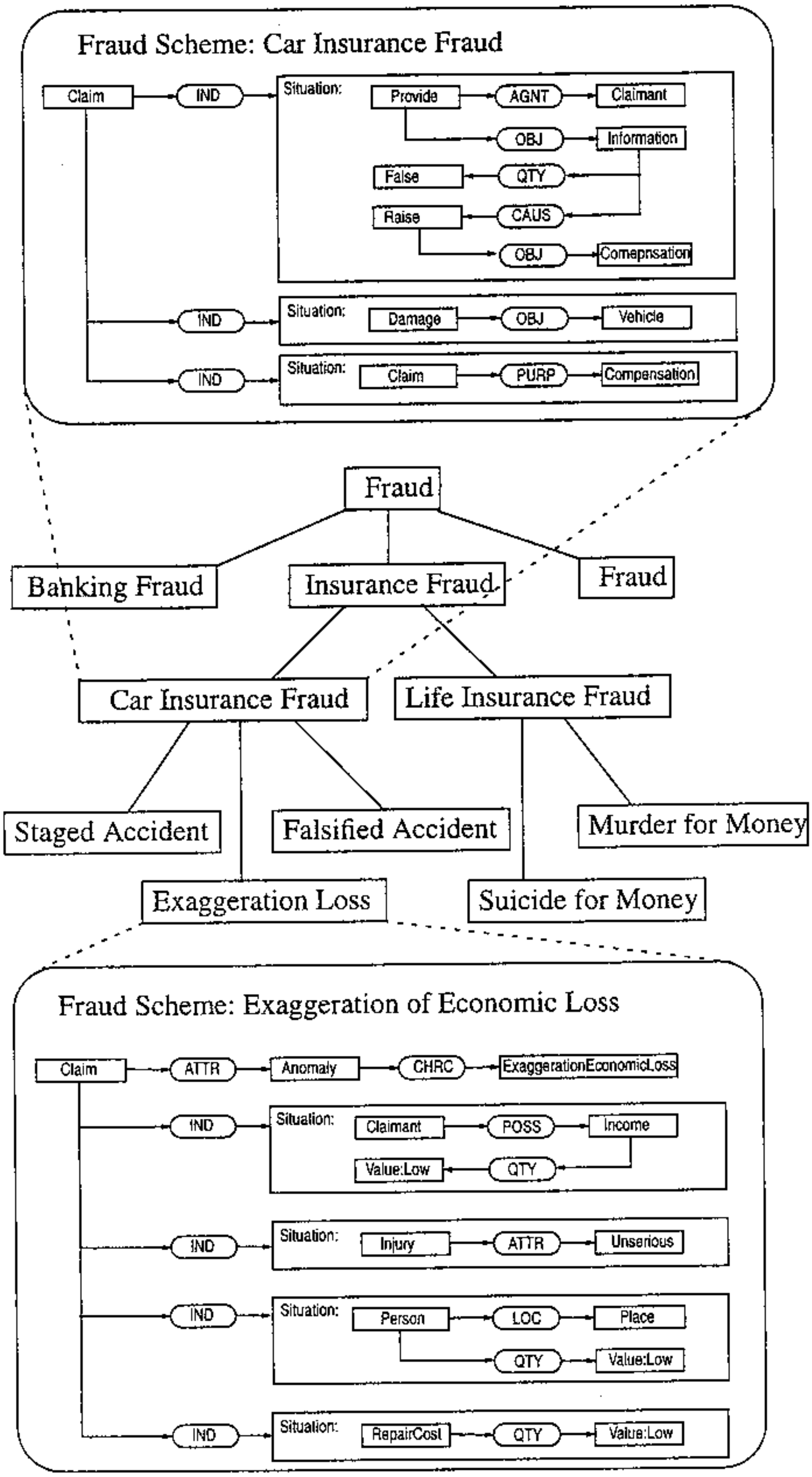


FIGURE 6.2 Fraud Scheme Hierarchy

In the above graphs, we only show the details for two fraud schemes: namely *Car Insurance Fraud* and *Exaggeration of Economic Loss.* The *Car Insurance Fraud* is a supertype of *Exaggeration of Economic Loss* in the fraud scheme hierarchy. Every fraud scheme has a set of indicators. If the explanation can satisfy one of the fraud schemes, this explanation will be accepted. Otherwise, it might be rejected or need to be modified.

The indicators in a fraud scheme are usually abstracted fraud evidence. For example, the scheme for a staged accident has the following indicators:

- The location has few pedestrians,
- The time is after dark,
- Two parties know each other,
- The claimant is in financial trouble, and
- Accident did not cause extra loss.

We tested this method of explanation evaluation through representation of every indicator in the form of conceptual graphs. If the explanation can be projected into any variation of the indicators, it will be accepted. The projection[151][156] algorithm will consider not only the concepts of indicators themselves but also sibling concepts which share the minimal common general type in the concept type hierarchy. The expansion to sibling concepts provides an approximate way to derive explanation measures.

Knowledge based explanation evaluation is, in some degree, analogous to semantic distance based approaches. We evaluate explanation by means of the difference between the explanation graph and the *fraud scheme indicator graph* as shown in the following formula:

$$p' = p_0'(explanation,\ scheme\text{-}indicator)$$

In the same way, we can also calculate credibility p and un-acceptability up (stated before). This method of evaluation for fraud detection is based on the following assumptions:

- the perpetrators (claimant) employ various ways to conceal their frauds

Although there are many different ways to commit fraud, the basic objective is to practise deception for financial gain. To be successful, perpetrators must use various ways to conceal their deception. As a result, the indicators of fraud schemes are relatively simple at an abstract level. (The difficulty is to link the evidence to indicators at the abstract level).

Knowledge based explanation evaluation can be readily extended to hypothesis evaluation by summarizing individual explanation credibilities. This approach is useful for comparing several hypotheses.

$$P = \sum p_i'(explanation_i, scheme\text{-}indicators_i)$$

or in an approximate way as:

$$P = p_i'(\sum explanation_i, \sum scheme\text{-}indicators_i)$$

The knowledge based approaches naturally favour explanations which are consistent with the system's knowledge. This could well disadvantage the novel explanations, and sometimes new discovery may be discounted. This disadvantage affects not only our methodology, but also appears in human expert's problem solving, and is believed be a common limitation of all knowledge based processes[88][128]. A compromise proposal to overcome this limitation involves combining multiple approaches, such as the use of structural and knowledge based approaches for new situations.

6.8 Hypothesis Modification Strategies

It is seldom the case that existing hypotheses can be applied directly to the present situation to either solve a problem, explain an anomaly or plan for a new goal [66]. In order to cope with ill-structured problems, where no existing solution currently exists, the hypothesis space successfully used in the past requires modification in a co-operative manner to make it fit the requirement of a new problem.

6.8.1 Explanatory Relation Modification

In the theoretical treatment of hypothesis conflict resolution, the aim of hypothesis evaluation was to modify explanations and thereby increase their credibility. In our experimental justification, three types of failures were attributed to conflicts:

1. Plausible failures

   Plausible failures correspond to explanations that do not make sense, because they contradict some important aspects of the application domain, as represented in the knowledge base.

2. Failures due to vagueness

   Vagueness arises from explanations that are not detailed enough, not sufficiently convincing, or do not contain sufficient information.

3. Conflicts between concepts

   This is the most common conflict arising from max-join operations. This failure corresponds to explanations that are not consistent and fail to satisfy the explanations' criteria for acceptance.

We have investigated several conflict resolution strategies, which can be used individually or in a (prioritized) combination. The strategies are as follows:

1. Concept substitution

   Concept substitution attempts to fix plausible failures by replacing a concept in an explanation pattern with an equivalent concept at the same level of the concept hierarchy. Alternatively, a contrary notion may be substituted!

2. Concept generalization

   Concept generalization broadens the scope of an explanation, at the cost of some detail information. In effect, an instance is generalised to its class!

3. Concept specialization

   Concept specification attempts to eliminate vagueness through additional information (e.g. specific context) through the creation of information rich hypotheses.

4. Deletion
   That part of a graph, a concept or a relation, which fails acceptance criteria given conditions, is deleted from the explanation. This strategy can be used to simplify explanations.

5. Human computer interaction
   Human computer interaction mechanism is the method of last resort when progress has stalled using the other methods. It provides a flexible dialogue and interface to acquire user knowledge in explaining the available evidence and for revising the hypothesis.

6.8.2 Executable Relation Modifications

The modification of an executable knowledge structure is a complex process even for human experts. The actor paradigm provides an effective solution for some simple modifications. Techniques such as substitution, generalization and specialisation, are commonly used to complement the operations involved in data retrieval processes.

Substitution attempts to fix problems by replacing an actor in the failed executable relations with an actor that could have the same causal consequences. The following substitutions have been investigated:

1. Replace an actor with a common stereotype.
2. Replace an actor with one closely related in the action hierarchy.
3. Replace an actor with one having the same pre-conditions.
4. Replace an actor with one known to have solved similar problems.

Generalizer reworks plausible failures by producing a version of an actor that applies to a broader class of situation, at the cost of some detail in the hypothesis.

1. Generalize old actor to make it compatible with the new actor.

2. Generalize the constraint on the new actor to make it compatible with current actor.
3. Delete the problematic actors.

Specialisation fixes problems of vagueness by expanding details in the executable relation, thereby making it less general, but richer in information content.

1. Expand an actor description by finding an actor matching the initial description derived from the problem situation, or from elsewhere (e.g. in the hypothesis!)
2. Specialise an actor description by finding an actor matching the motivation implicit in the problem case.
3. Add details to the causal connections between two concepts by splicing in another hypothesis that could explain the origin of the link.

At this stage, the process for modifying virtual relations is usually carried out using human computer interaction. Automatic modification is typically limited. Further research is needed to provide an adequate modification process, especially in the process of concept specialization. An effective environment has, however, been realised for virtual relation modification under support of hypothesis based interaction.

6.9 Example of Evaluation

In this section, we explain the hypothesis evaluation process by an example. The evidence detected by the anomaly detection model is described in FIGURE 6.3, as the evidence graph combining all the anomalous attributes detected.

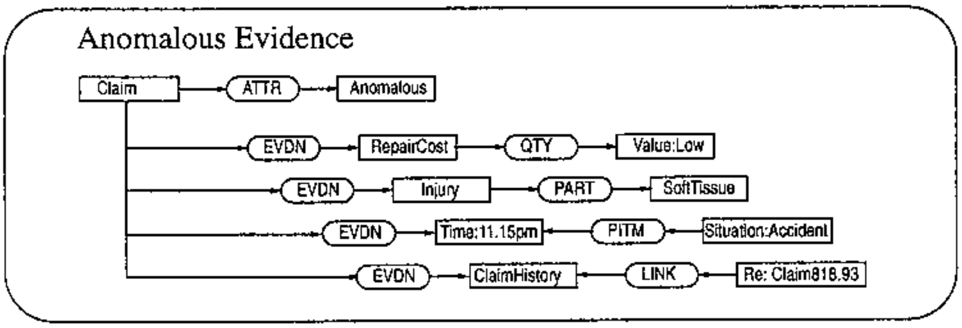


FIGURE 6.3 Anomalous Evidence Graph

This evidential graph can be explained as:

1. Low repair cost,
2. Soft tissue injury only,
3. Accident happened at 11.15 pm, and
4. With previous claim: claim number 818.93

The hypotheses indexed by the evidence are *"Exaggeration of Economic Loss"* and *"Staged Accident"*, which are merged automatically in hypothesis space (shown later FIGURE 6.6) The fraud scheme indicators for these hypotheses are depicted in FIGURE 6.4 and FIGURE 6.5. The next step is to determine which hypothesis has the highest credibility based on following formula:

$$P = \sum p_i'(evidence_i, scheme\text{-}indicator_i)$$

As a result of the evaluation, the hypothesis *"Exaggeration of Economic Loss"* is selected. (For the hypothesis *"Staged Accident"*, the indicator *"repair cost is very high"* is not matched, and the indicator *"two parties involved know each other"* has not been verified). From the activated space we can also determine that there is no match for the evidence *"Repair Cost is low"*

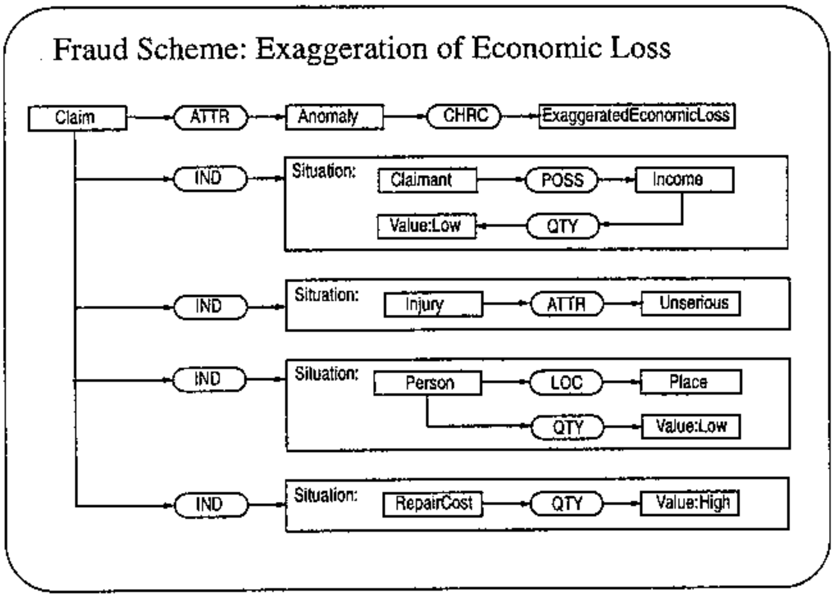


FIGURE 6.4 Fraud Scheme: Exaggeration of Economic Loss

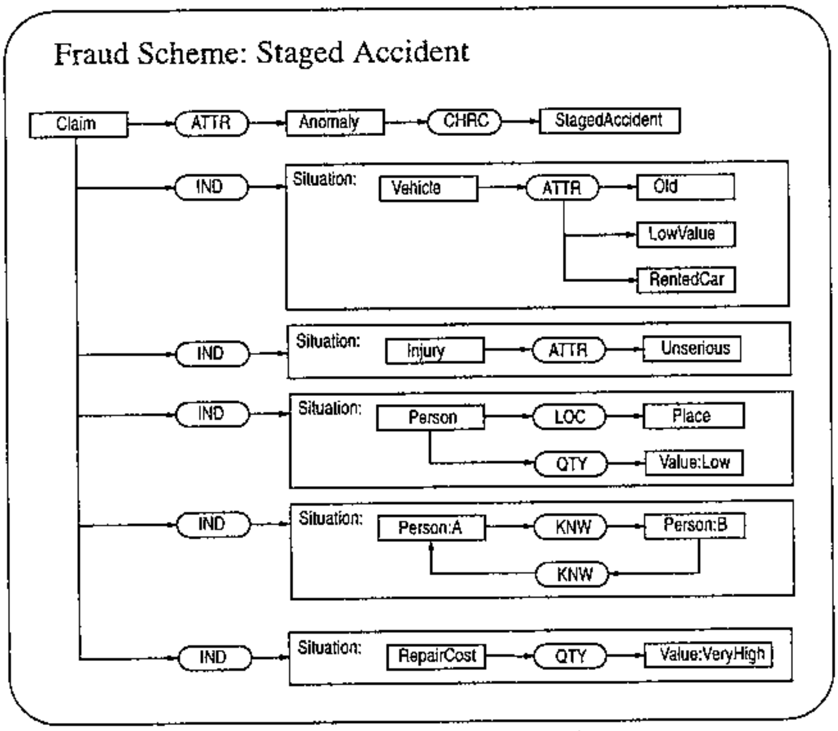


FIGURE 6.5 Fraud Scheme: Staged Accident

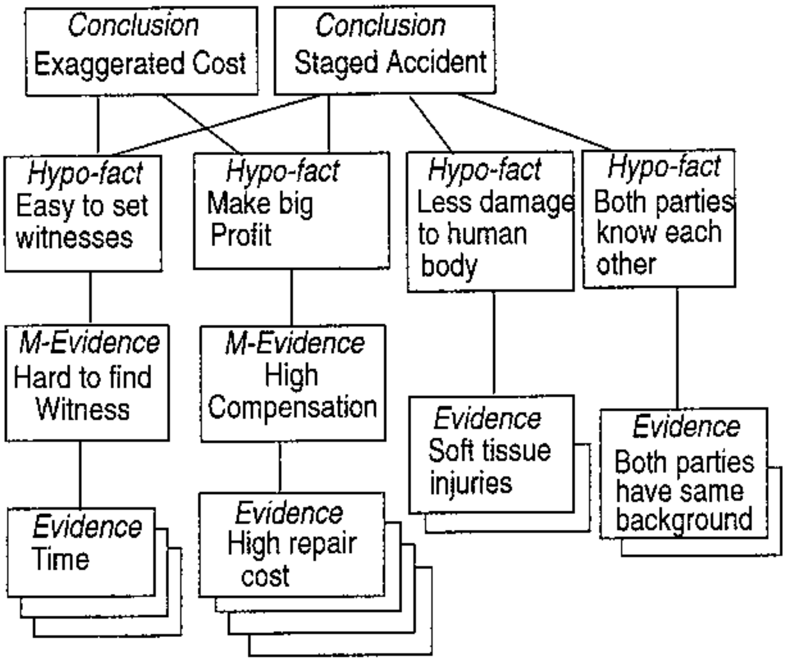


FIGURE 6.6 Activated Hypothesis Space

The resultant graphs for the new evidence *"Repair Cost is low"* is shown in FIGURE 6.7 and the whole explanatory graphs, which are generated by max-join of the explanation paths from evidence node to conclusion node in hypothesis space, is shown in FIGURE 6.8.

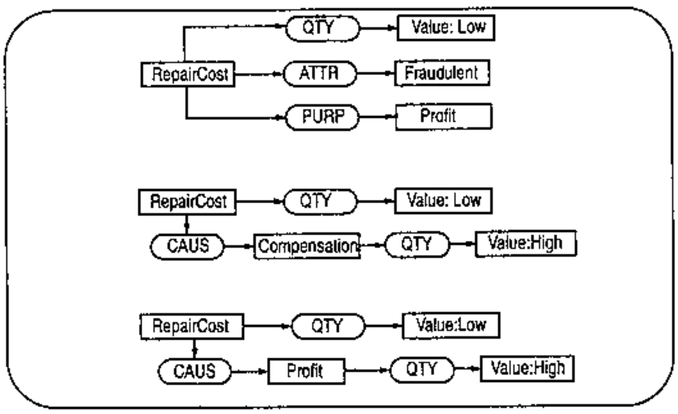


FIGURE 6.7 Resultant Graphs of Abductive Reasoning

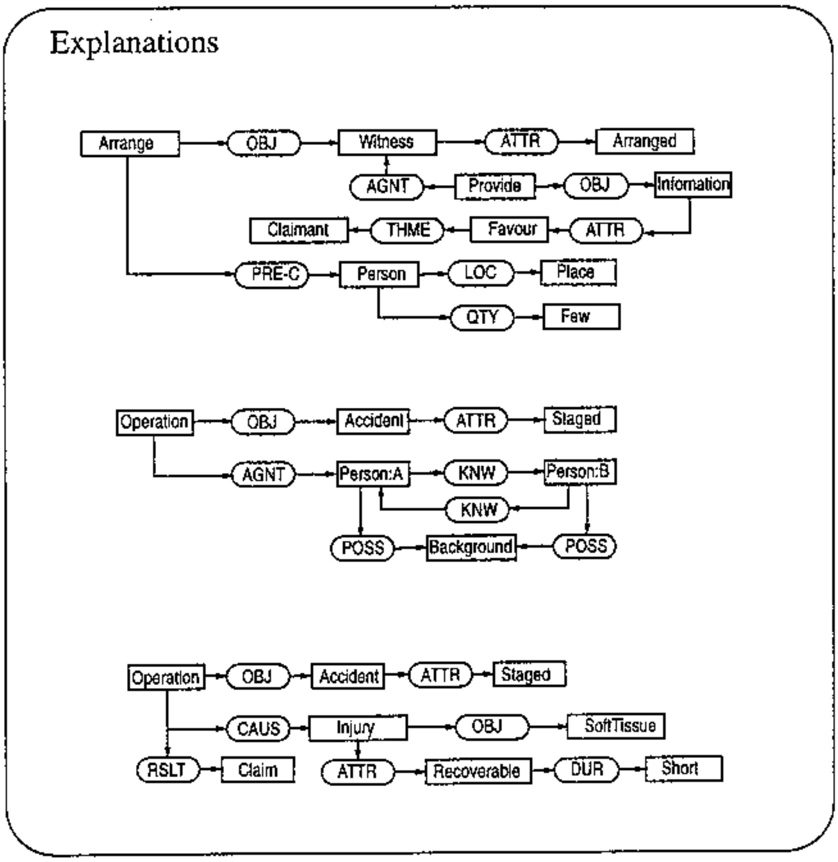


FIGURE 6.8 Explanatory Graphs Involved in Active Hypothesis Space

There are two approaches for evaluating the explanation graphs generated for the new evidence. The first is to evaluate the individual resultant graphs one by one, and then combine the "best" one into other high level explanations by maximal-join operations. Another approach is to generate the combined explanation graphs by applying max-join operation on several resultant graphs, and then evaluate all derived explanation graphs.

Theoretically, the latter approach would appear to be the most compelling. However, our research indicates it also introduced complexity into the evaluation, and the process is more time consuming. FIGURE 6.9 demonstrates the complexity of the resultant graphs and combined graphs using these two approaches. From the graphs, we can determine that a large part of the combined graphs are common to all, which implies that most computation on evaluating combined explanations is redundant. In conclusion, the evaluating resultant graphs offers the fastest solution.

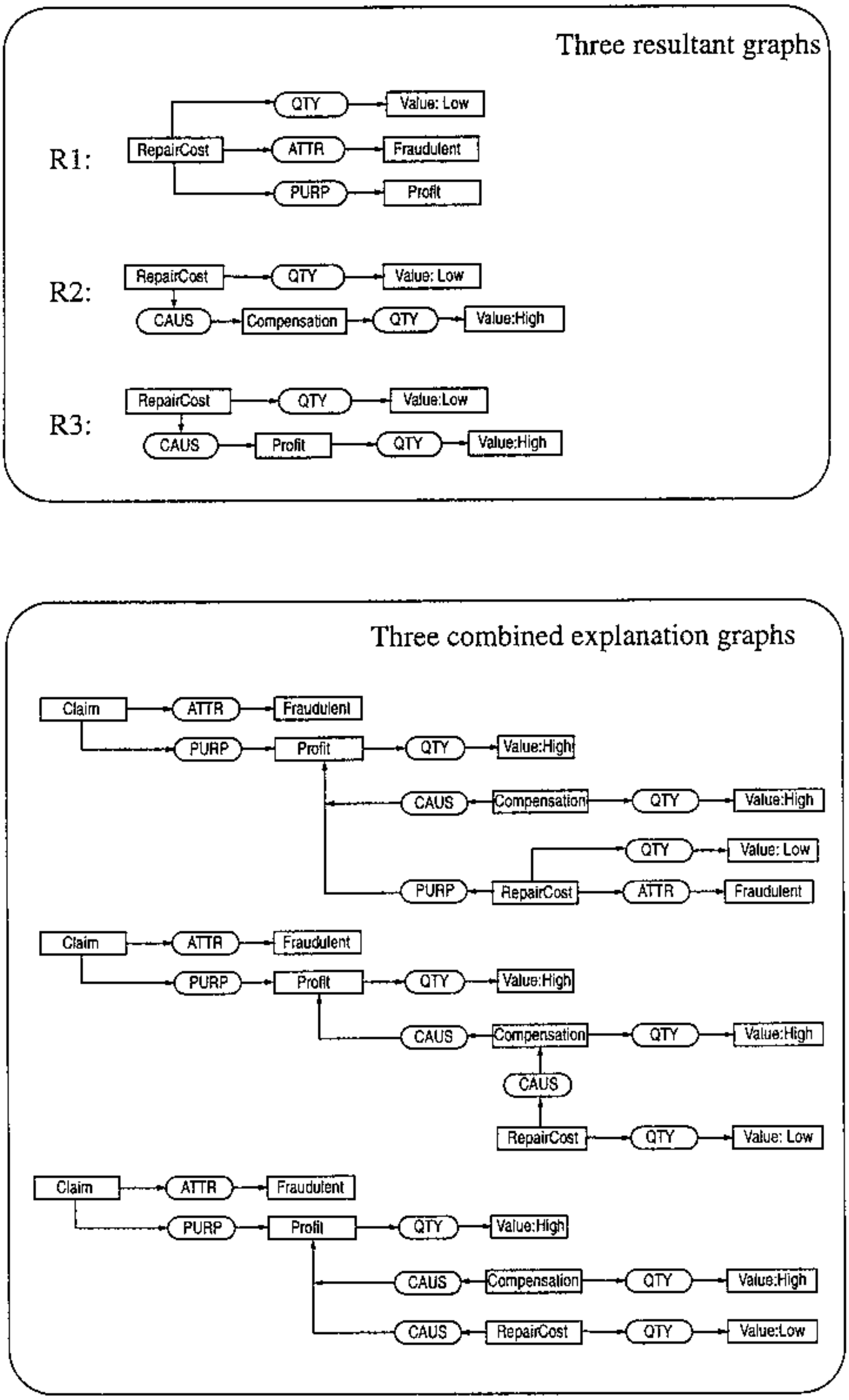


FIGURE 6.9 Complexity Comparison

The three resultant graphs in FIGURE 6.9 are now discussed in more detail:

- There is neither any existing explanatory relation nor any executable relation for *"fraudulent repair cost"* in R1. R1 would not then be considered, at least at this stage.
- *R2* and *R3* are similar in structure and semantics except for the concepts *compensation* and *profit*. Since *compensation* is more specific than *profit* according to the *concept type hierarchy table*, we choose *R2* instead of *R3* based on the strategy "A specific case is better than a general one".
- Because the concept *repair cost* and concept *compensation* have a common root in *the concept hierarchy of insurance fraud domain*[1], the labels for the concept *value* connected by two separated relations *quantity* (QTY) are in conflict. Since *the quantity of Repair Cost is low* is evidence detected, (i.e., verified evidence) the label *(high)* will be replaced by verified label *low* using our concept substitute modification strategy.

After local conflict resolution, the selected resultant explanation is shown in FIGURE 6.10.

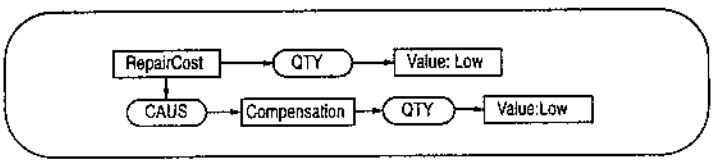


FIGURE 6.10 Selected Explanation Graph

The combined explanation graph using the selected explanation graph (as shown in FIGURE 6.10) will be generated by a max-join operation on all explanation relations from the new evidence to conclusion. The label of concept *value*, which is connected from *profit* by *QTY* is substituted by *low* using the same strategy for local conflict resolution.

1. A concept hierarchy contains common concepts used for insurance fraud detection.

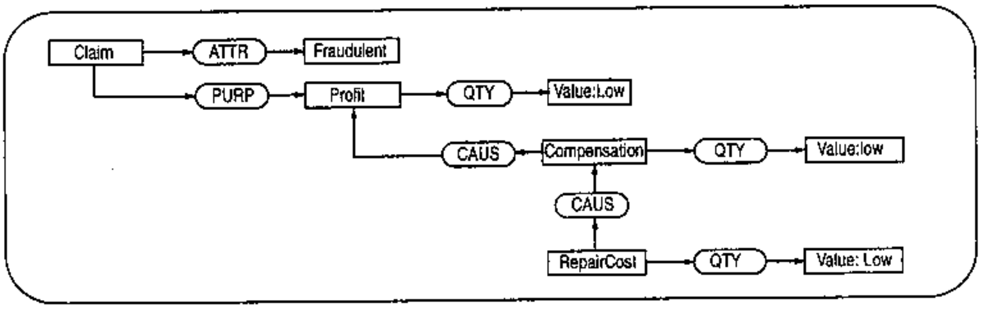


FIGURE 6.11 Combined Explanation Graph

6.10 Summary

Some type of quality measurement for explanation based paradigms are found in many artificial intelligence systems. The explanation credibility and hypothesis credibility measurements developed in this chapter propose an indirect method to measure the "quality" of an explanation for given evidence. All measurement is primarily based on semantic distance. The difference measurement calculates the credibility through the calculation of conceptual graph difference.

There are a number of research groups dedicated to explanation evaluation. Some of them discuss the problem from the viewpoint of psychological criteria. Most of them are interested in generating explanations for expert systems. Our approach is different from both of them in the form of its implementation.

Credibility can be used to resolve local explanation conflicts and to select a suitable explanation from multiple candidates derived from the given evidence. The un-acceptability measure can be used to resolve global conflicts, and to identify explanations from a set of explanations.

The significance of this research is now summarised:

- Our approach is conceptual graph based, and the inherent property of conceptual graphs [95][156] provides unique context-based criteria for evaluation.
- The credibility measurement provides an indirect way to measure the quality of explanations;
- The credibility measurement can be used to resolve the local explanations' conflicts.

The most obvious and frequent criticism raised in use of abductive inference concerns the redundancy produced by *Max-join*. Although we have argued that these are not as serious as would appear at first sight, there clearly does not exist an unique "best" explanation. Also, there are some uncertain features involved in measuring the "goodness" of an hypothesis.

Another conclusion from this chapter is that no explanation assessment technique can be used blindly. The information gained from consideration of competing hypotheses provide the system with an initial assessment, from which an explicit analysis can be made of how sensitive this assessment is to changes in a priori assumptions and to details of the assessment procedure. With the assessment of complex hypotheses no single assessment technique appears to give definitive results. The best that can be hoped for is that a range of different assessment techniques will give comparable results.

This approach is realistic for complex domains such as fraud detection and management intelligence systems where existing hypotheses are not always applicable and existing explanation patterns can rarely explain new evidences.

CHAPTER 7

Case Based Interaction

7.1 Introduction

Many knowledge based systems fail because discrepancies exist between the system's view of the world and reality. Such systems have three primary components: a knowledge base, decision rules, and an inference engine. The techniques for knowledge engineering have not changed much generally, and creating a current knowledge base is a difficult and painstaking process. In particular, the AI research community has had limited success in capitalising on the demands for robust, predictive models using knowledge based systems. As a result, there is an obvious gap between most system models and reality.

Our research[52] into hypothesis generation for management intelligence has demonstrated the value of intelligent user interfaces. We now show that the inclusion of hypothesis in knowledge based system facilitates the cooperation between the human user and the computer, and it also, provides a mechanism for directing the interference strategy for particular user queries.

In this chapter, we discuss, in particular, what roles hypotheses can play between the user and knowledge based systems. Our aim is to bridge the gaps between knowledge based systems and reality by introducing hypothesis as a medium, and to develop the capability to reason about problems at the limits of system knowledge. The requisite roles of hypotheses are as follows:

1. the control capability for inferencing using control strategies implied in the hypothesis structure,
2. improved communication between the user and knowledge based systems in the process of problem solving, and
3. a mechanism to focus the system resources on interesting aspects of the problem solving, and thereby avoid exhaustive search.

FIGURE 7.1 graphically displays the role of hypothesis in problem solving.

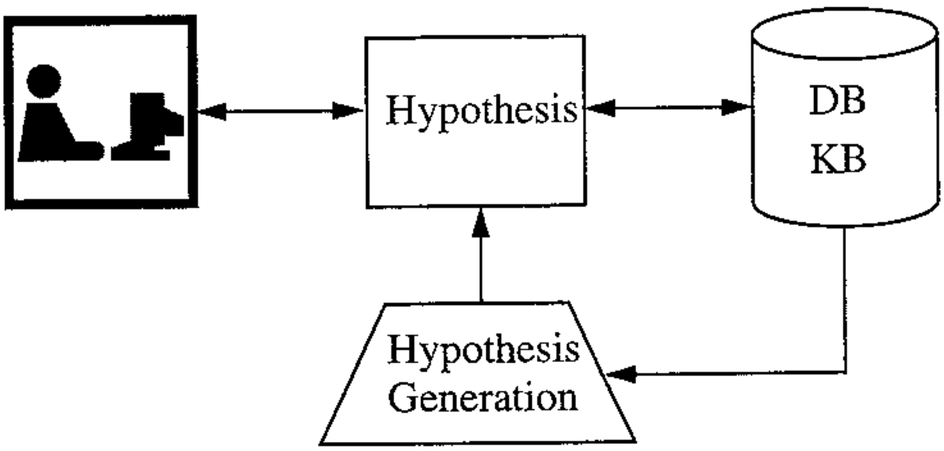


FIGURE 7.1 Hypothesis as an Intelligent Interface

7.2 Case Based Interaction

Cooperative problem solving, which refers to the cooperation between a user and a computer system in our research, is expected to facilitate the competence to solve complex problems by utilisation of the knowledge from both sides of the interface: user and system.

This type of interaction is different from the cooperation involved in *Computer-Supported Cooperative Work* (*CSCW*) [64][9] and *Distributed Artificial Intelligence* (*DAI*)[6][62] [86], or *Distributed Expert Systems (DES*) [168]. *CSCW* has emerged as an identifiable research field focussed on the role of the computer in group work, and has mainly concerned cooperation among users brought together by computer networks, while *DAI* is divided by Bond and Gasser [6] into three areas: *Distributed Problem Solving* which determines how the work of solving a specific problem can be divided among a number of modules, *Parallel AI* which is concerned with developing architectures, languages, etc. for AI in parallel computing environments, and *Multiple-Agent Intelligent Systems* which is concerned with the co-ordination of behavior among a collection of intelligent agents. *DES* mainly addresses the resolution of conflicts caused by applying rules from different rule bases. In our research, the cooperative problem solving system deals mainly with the interactions of a group of intelligent agents (users and systems) to create a team that acts together to solve a complex task, such as fraud detection in an *EDP* environment [49].

The interaction between the user and the computer is the main issue in cooperation, so the communication module plays a key role in information or knowledge transfer between user and computer in order to facilitate complex problem solving. Computers provide external memory for the user, insure consistency, hide irrelevant information and summarize and visualize information. The human works like an extra inference engine and provides knowledge bases to bear for use by the system. The communication module brings the power of both sides together.

Case-based reasoning (CBR) is a problem solving method that involves recalling a previous, similar situation and applying its solution to a current problem. Schank and Leake's recent work has demonstrated [145] the application of case-based reasoning as a new explanation mechanism. The concept of Case-based Interaction (CBI) as a mechanism for intelligent Human Computer Interaction was first expounded by Professor's Edmonds and Garner in 1991, and their ideas were incorporated in a new ESPRIT project proposal[35].

In the current research undertaken at Deakin University, case-based interaction utilizes an hypothesis structure for the representation of relevant planning and strategic knowledge.

The extent to which case-based explanation can be utilized as a module of an *Human-Computer Interaction* (HCI) system is now explored. Our approach extends the notion of goal space modeling[95] utilizing case-based interaction through the mediation of the communication controller.

7.3 The Requirement for Cooperation

A broad range of *Well Structured Problems (WSP)* - embracing forms of diagnosis, category selection, and skeletal planning - are solved by "Expert Systems" with the methods of heuristic classification[24]. But in many situations, the problem is ill structured (*ISP*) such as fraud detection, environment assessment, et. al. It is difficult to anticipate the variety of problems that might arise in achieving a goal in the real world, and there is no universal solution.

The boundary between *WSP* and *ISP* is imprecise. Simon uses the notion of residual concept to define *ISP* as a problem whose structure lacks definition in some respect. A problem is an *ISP* if it is not a *WSP*. According to Simon[149], a well-structured problem has some or all of the following characteristics:

1. There is a definite criterion for testing any proposed solution, and a mechanizable process for applying the criterion.
2. There is at least one problem space in which can be represented the initial problem state, the goal state, and all other states that may be reached, or considered, in the course of attempting a solution of the problem.
3. Attainable state changes can be represented in a problem space, as transitions from given states directly attainable from them.
4. Any knowledge that the problem solver can acquire about the problem can be represented in one or more problem space.
5. If the actual problem involves acting upon the external world, then the definition of the state changes and of the effects upon the state of applying any operator reflect with complete accuracy in one or more problem

spaces the laws that govern the external world.

6. All of these conditions hold in the strong sense that the basic process postulated should require only practicable amounts of computation, and the information postulated is effectively available to the process.

In general, problems presented to problem solvers by the world are best regarded as ill structured. They become *WSPs* only in the process of being categorised as solvable by well known methods.

Many knowledge-based systems are built on the assumption that the user has a *WSP* that the system is supposed to solve. More frequently, users learn incrementally about the nature of their problem, and they want to solve them in cooperation with heuristics provided by a system.

In most classical expert systems, include the rule-based system *MYCIN* [14], frame-based system *GRUNDY* [134], and *SOPHIE* [12] et. al., the user is a passive agent who is asked for inputs, and the system makes all decisions by itself and then returns an answer, together hopefully, with some explanation.

Human-computer interaction (HCI) is believed to be one of the key issues in cooperative problem solving, and the communication module will play a key role in dynamic information transfer between user and computer systems in order to facilitate problem solving. The design of the communication module between human and computer systems should meet the following requirements:

1. Coordinate control when there is a conflict of opinion between agents.
2. Give control to users when they need or desire it, otherwise, provide automatic support.
3. Promote human problem-domain communication; mirror the abstractions of the problem domain, thereby reducing the transformation distance between task descriptions by the domain expert and their representation as computer programs.

One of the most promising approaches for coping with the HCI requirements is to include knowledge-based systems that contain knowledge about specific problem domains, the partners (agents), communication processing, recovery from breakdowns, and help and explanation facilities for increasing comprehensibility.

7.4 Contemporary Human-Computer Interfaces

The Human-Computer Interface (*HCI*) is the mediator of all information between the user and systems. Therefore, it is a crucial element for any intelligent system to succeed in decision support. The infrastructure for evolution of effective human-computer interfaces was laid during the last decade. In this period, the literature on human factors and behavioral science research concentrated on interface design from an empirical perspective [100], and much research in computer science addressed interface design principles and guidelines [22]. Recently, Hartson and Hix wrote an excellent survey article [69] that covered the range of human-computer interface development. In this section, we present the more important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures.

Advances in computer and communication technology continue to result in increasingly complex systems. In many cases, human users are still very much involved in system operations. There are trendy areas of current research designated by the term "adaptive interface". Hoppe [75] proposed the following steps of information processing for adaptive interactive systems:

- assessment of user characteristics (constructing a user profile);
- diagnosis of the user's current needs based on what he/she is actually doing;
- derivation of adequate responses to the user's actual needs taking into account the long-term characteristics.

Two recent attempts in developing human-computer interaction systems are the *ESPRIT II* project 2630 - *FOCUS* (Front-ends for Open and Closed User system) and the *ESPRIT II* project 2474 - *MMI2* (Multi Mode Interface for Man Machine Interaction with knowledge based systems).

> The *MMI2* project aims to build a man/machine interface for different kinds of user, integrating several modes of communication supported by modern workstations: natural language, command language, graphic and gesture. The interface will provide simultaneously modes suitable to support the efficiency of experienced, professional users (command language, menus) and natural communication modes well suited to naive users, such as graphics and natural language [5].

MMI2 is an ambitious architecture for multi-mode interface systems, addressing concerns such as user modeling, communication planning, multi-mode meaning representation, dialogue context managing, etc. But it seems not possible to develop all features to the same degree of completeness within the scope of the *MMI2* project.

In parallel with the *MMI2* project, the *FOCUS* project was co-ordinated by the *LUTCHI* Research Centre, Loughborough. It is designed to assist computer users in a wide and ever-growing range of application packages and libraries and addresses a large variety of computing tasks across the spectrum of subject areas.

> The aim of the *FOCUS* project is to develop tools, techniques and methodologies for the construction of *KBFEs* for use in conjunction with "open" computational systems and "closed" systems.
> It is also developing a harness which provides the framework for the interaction between end users, the front-ends and the user system which is considered as the "Back-End". The combination of the framework and its *KBFE* components will facilitate the provision of an enhanced human-computer interface and will also potentially provide a more profound level of assistance than is generally available at present [34].

User Interface Management System (*UIMS*) attempts to build the construction of consistent interfaces that can be rapidly developed by providing a uniform set of high-quality primitives [113]. The major issue in *UIMS is* how well tools support the development of user interfaces by providing the right functionality to make it easy to develop good user interfaces. Most *UIMSs* are based on a strong separation of interface and application code and it is a good approach to problems for which there is only limited information exchange.

Other works in knowledge-based human-computer interaction include *UIDE*: the User Interface Design Environment, a system to assist in user interface design and implementation [45]; *SAUCI*: Self-Adaptive User-Computer Interface [157]; *Friend21* project: A construction of 21st century human interface [112]; et. al [96] [116].

7.5 Interaction Guided by Hypotheses

Current research undertaken in Deakin University, in case-based interaction utilizes a prototypical knowledge structure, called hypothesis space, for the representation of relevant planning and strategic knowledge for fraud detection within the EDP domain.

In traditional menu-based interaction, the user always drives the interaction process. The menu restricts alternative actions that the user could perform at any one stage of the interaction. For some applications and for some users, these approach with a predefined set of functions would suffice to accomplish nearly all tasks. In cooperative problem solving environments, case-based interaction is guided by the case rather than by the user. The interaction process attempts to explicate the user's goal space to select and instantiate a set of probable cases to solve the ill structured problem that may have no predefined solution.

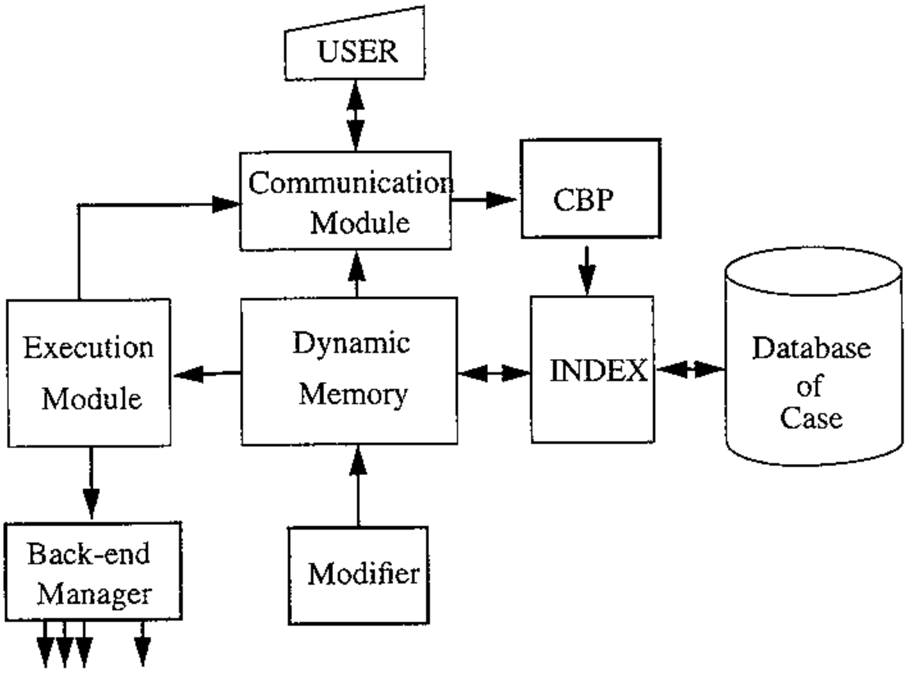


FIGURE 7.2 Structure of Case-Based Interaction

As noted earlier the communication module is the key part in case-based interaction. The role of the communication module is to manage the information flows: the information input to the system from users and the requested output to the users, as well as to managing the system resources and the screen (window) dynamics, in order to utilize effectively the human-computer interface.

It is obvious that some information requests are more important than others. Importance can be expressed in terms of the priorities of the information in question relative to achievement of one or more goals within the goal hierarchy of the domain of interest. A criterion in setting priorities is the timeliness necessary for the information to be useful. If presented too soon, the relevance of the information may not yet be clear; if presented to late, it may no longer be useful.

Within case-based interaction, priorities are determined by information requirements associated with the successful pursuit of users' intentions. More specifically, the case structure includes a hierarchical representation of the goals, plans, and actions that are potentially relevant within the domain of interest. This representation is annotated with information and control requirements that are determined using contemporary knowledge acquisition techniques. In this way, the communication module can obtain a baseline estimation of information requirements but using the active goals and plans as pointers.

This baseline is augmented in several ways. Based on our philosophy of interface design, information requests by the user automatically assume higher priority. In addition, requests submitted via the system are associated with priorities set by the organizer of hypothesis knowledge structures. One of the priority criteria in case-based interaction is the importance of the requirement to verify an hypothesis. For example, the request to verify high weighted evidence has a higher priority than the request to verify low weighted evidence. Another criterion considered here is the cost to retrieve this information. Information with a lower cost should have a higher priority than that with the higher cost.

Once the priorities are assigned, communication control between human and computer becomes a standard scheduling problem and can be solved fairly easily. There are various methods for queue management that can be used to determine the order in which requests should be serviced. Furthermore, if the communication module has a resource allocation problem, information importance priorities can be set. The resources include the user's information-processing capacity, input/output channels and remote databases or knowledge bases, as well as screen (icon) management.

7.6 Knowledge Acquisition in Case-based Interaction

In traditional knowledge based systems, the knowledge acquisition process was completed before the system could be used. Such systems cannot tackle problems without knowledge being available in pre-stored knowledge bases.

With CBI, the knowledge acquisition process is activated when a gap between the existing knowledge base and the real world is detected. Such situations, in which the system could not solve the problem by itself, might basically fall into three groups:

1. Case retrieval failure (Hypothesis retrieval failure in our situation)

2. Case instantiation failure
3. Case modification failure

Recent discoveries in *Goal Interpretation Mechanism* [95] have highlighted goal abstraction as a novel and significant knowledge acquisition mechanism and have resulted in the identification of a number of goal abstractions. Goal interpretation is used to explicate and elicit knowledge from users.

Goal interpretation is an intermediate process, in which the system tries to identify and understand the goal implied in user statements. The objective of goal interpretation has been summarized by Dr. Lukose [95] as follows:

- Identify "user goals" and their associated requirement(s) from the "user aims" (i.e. user input);
- Determine the intentions behind the "user goal", since knowledge of the user intentions can facilitate the replanning process;
- Construct the goal state (i.e. the condition(s)) under which the goal state is fulfilled.

Essentially, there are two types of knowledge relevant to problem solving within CBI; namely, explanatory knowledge and strategic knowledge, which is defined as knowledge concerned with actions, plans, rules, and goals. Following the knowledge elicitation process, knowledge factorization is initiated, in which the user will factorize the knowledge into types or categories. The process of knowledge factorization involves the transformation of expert knowledge into concepts, relations, graphs, and abstractions.

Once a satisfactory knowledge base is obtained, problems within the domain can be solved. During problem solving, the system will also interact with the user to acquire any additional problem specific knowledge that is relevant to the particular problem.

In *CBI*, efforts are concentrated on the acquisition of problem solving strategies from the experts through the procedure developed for hypothesis space modeling, which plays a central role in communication control between the end user and knowledge based systems, as well between various partners of the intelligent human-computer interface system.

Strategic knowledge is the knowledge used by an agent to decide what action to perform next, where actions affect both what are believed by the agent and the state of the external world [3]. We define strategic knowledge as that process knowledge that enables decisions (iteratively) on actions appropriate in the current situation. Given a general structure of strategic knowledge, we must design or adapt an operational representation that can be executed by a knowledge system to achieve the desired behavior. In practical terms, this means designing an architecture for control. The paradigm underlying this acquisition process is an iterative control strategy, where *CBI* decides what to do and decides on the next step at each iteration of the control cycle.

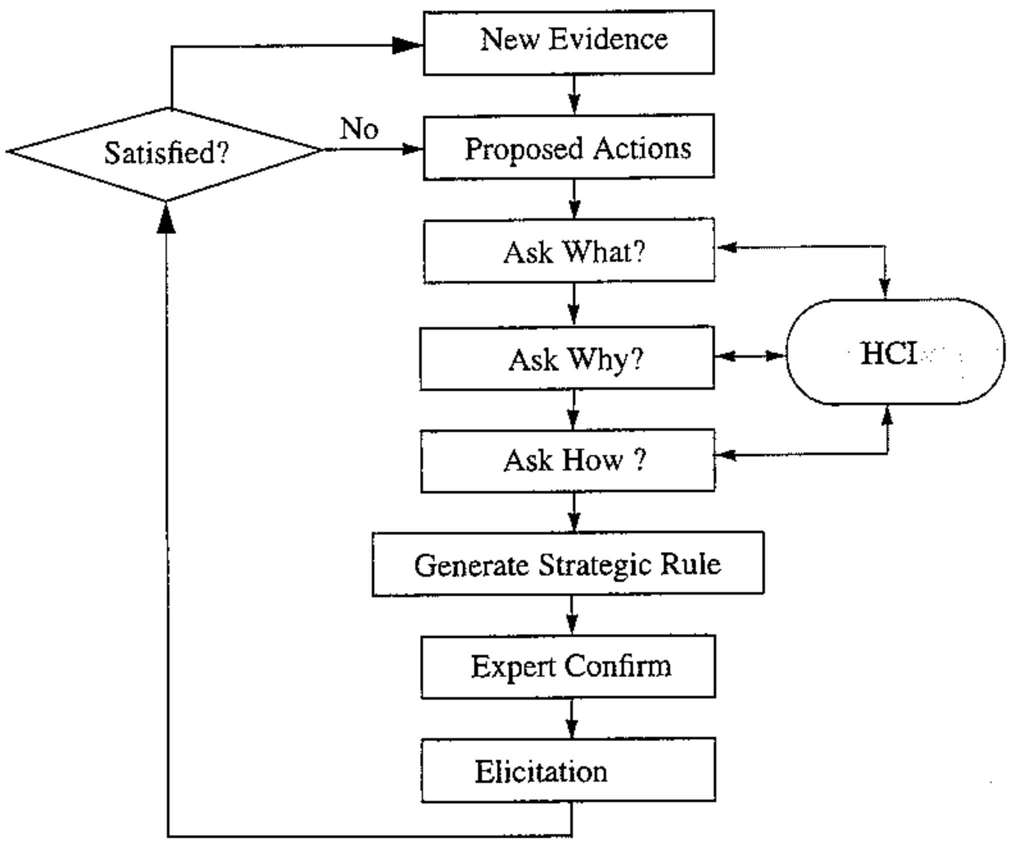


FIGURE 7.3 Paradigm for Strategic Knowledge Acquisition

Figure 7.3 illustrates the strategic knowledge acquisition paradigm. At each iteration, the

CBI proposes a recommended action and a set of alternative actions which are listed, based on their priorities. If the user agrees with the system's decision, *CBI* executes the recommended action. No knowledge acquisition process is required. The system will change the state of the goal space (hypothesis structures), and then enter the next iteration. If the user does not approve of the system's choice, *CBI* will initiate a knowledge acquisition dialogue to resolve the conflict, since the system has insufficient strategic knowledge to continue.

In the process of strategic knowledge acquisition, the system firstly provides a set of actions which might be suitable in the current situation for the user to choose from. The user can simply choose from a menu or enter the name of a new action within the dialogue box. In this process, the user is required to assist through *CBI* in assigning new action to the group it belongs to and in defining the relationship with other actions.

Secondly, the user is required to interpret the reason for his choice. Meanwhile, the system displays the explanation for its requirements. At this stage, the user is provided in the explanation dialogue box with the menu of:

1. No reference;
2. User overwrite *CBI*'s choice;
3. Supplement *CBI*'s choices with reasons; and
4. User explanations.

Once this dialogue has terminated, the *knowledge based management system* will carry on consistency checks. If there is a conflict in introducing the new strategic knowledge into the knowledge base, *CBI* will display information regarding the conflict, together with the systems reasoning. The system runs the dialogue box iteratively until no further conflict is detected. *CBI* will then incorporate the new strategy into an hypothesis or hypothesis space.

Finally, after the acquisition of strategic knowledge, the system will generate a new relation, which is the detailed record covering evidence of the hypothesis, the actions in this hypothesis, and the outcomes (results).

7.7 Roles of Hypothesis in Human Computer Interaction

In a cooperative problem solving environment, case-based interaction is guided by the relationships within the hypothesis rather than by the user. The interaction process attempts to explicate the user's goal space to select and instantiate a set of probable hypotheses to solve the ill-structured problem that may have no predefined solution path.

The function of the hypothesis is to act as an intelligent clearing house for all information between the user and the knowledge based system. The principal role is the control capability for inferencing using strategies implied in the hypothesis structures. Secondly, hypotheses may be used to provide improved communications between the user and knowledge based systems in cooperative problem solving. Finally, hypotheses may be used to focus the system resources on interesting aspects of the problem solving, and thereby avoid exhaustive search.

7.7.1 Inference Control Strategies

The principal role of hypothesis in an intelligent interface is the provision of a control capability for the inference process by using control strategies implied in hypothesis structures. Instead of directly controlling the inference, hypotheses control inference by providing subgoals and modifying the intermediate goals promptly, according to the outcome of the inference. The problem solving process is thus captured at a high level of abstraction, as reflected in the hypotheses.

The intermediate goals (linking sub-conclusions and facts) can be implied by utilizing the relations in the hypothesis structure. In order to verify an hypothesis, it is worthwhile to verify some key facts and sub-conclusions. The key facts are actually used to imply key evidence by means of the probability propagation in the Bayesian model or other cause-effect theories. By providing a suitable intermediate goal, the hypothesis can improve the efficiency of problem solving.

In fact, we can carry this idea one step further: not only can implicit knowledge in hypothesis help control the inference, but it can also be used to direct action intended to accomplish some key intermediate goals. Rather than passively waiting for the outcome of the inference process, the hypothesis can actively anticipate and pursue missing steps. These functions are achieved by the execution of procedural knowledge, (actors) incorporated in the *executable relations* of hypotheses.

With the support of hypothesis, knowledge based systems provide feedback, in which the search outcomes are taken into consideration in determining future control strategies. The results are also used to enhance or reject the current hypothesis. Without hypothesis inference control, the knowledge based system will continue its exhaustive search, unless cut, without considering intermediate states. Using feedback, the system can immediately revise a hypothesis, update its control strategies, and avoid blind alleys.

7.7.2 Communication Control Techniques

The user and knowledge based system must interact with each other to achieve an identified goal. Traditional interaction in knowledge based systems has been command oriented, supported by menus or other means to communicate progress of the inference. This type of support is restricted and quite simple. For example, manipulations may include walking menus, dragged folders, or mouse actions. It can require a high degree of familiarity with the system interface for the user to perform the requisite operations sequence.

Our particular form of modelling human-computer interaction is known as hypothesis based. The communication by means of hypothesis aims to facilitate cooperation, not just provide a set of discrete commands/responses and questions/answers which are prescribed scripts for achieving a given goal. An agent for cooperation between a user and a computer system has been established and provides the competence to solve complex problems by utilizing of the knowledge from both sides of the interface: user and knowledge based system.

Our approach to dialogue control is under the guidance of hypothesis, i.e. controlled by hypothesis logic instead of a fixed menu system or script. After an hypothesis is generated, the dialogue is solely directed to verification of this hypothesis, and there is no need for further communication between the user and the knowledge based system. The strategies for controlling such dialogues are:

1. The *relations* between the nodes in an hypothesis imply the required information, and only the nodes linked by most possible *relations* (high weighting) will be processed;
2. The candidate nodes will be re-organized after receiving outcomes and new information;
3. The hypothesis will be rejected if the key feature of an hypothesis cannot be verified. Hypothesis generation process will be restarted;
4. User categorisation (from novice to expert) can be used to indicate the levels of abstraction required for individual users;
5. The dialogue process is quite flexible.For example, in the execution of an hypothesis, a request is usually sent to the information retriever and user at the same time. If the response from user precedes that from the system, then the retrieval process will be terminated.

In this way, hypothesis based interaction can provide improved communication between user and system. The hypothesis based interface is utilized to identify what information the user needs, what information the knowledge based system needs and how to pass this information in the form of commands.

Under the guidance of an hypothesis, the users are active agents empowered by the systems' knowledge, and the systems have an auxiliary knowledge source in the user. Computers provide external memory for the user, insure consistency, hide irrelevant information and summarize and visualize information [44]. And the human works like an external knowledge base to the systems. Cooperative problem solving enables the strengths of both partners to be exploited to the full.

7.7.3 Focus on Inference Intension

Another promising attribute of an hypothesis based interface is the capability to focus the system resources on interesting aspects of problem solving, and thereby avoid exhaustive search. After a suitable hypothesis is generated and indexed, the problem solving becomes one of hypothesis verification. A complicated problem is divided into several simpler sub-problems, whereby several attributes would require validation to achieve a complete solution. This strategy will dramatically decrease the search space and the cost of problem solving.

Hypotheses focus system's inference intension by providing more detailed sub goals. One of the functions is to verify the causal relation between two nodes. If the system can infer the link between one node to another, or vice versa, the causal relation will be established. In this way, inferencing in our knowledge based system will be better focussed.

Hypothesis can also imply the relevant information sources and knowledge bases. For a complicated problem, it is often less effective to rely on a single information source and a single knowledge base (rule base). Our results indicate that the size of a knowledge base will directly affect the efficiency of search, suggesting that dividing a large knowledge base into several smaller ones will improve the efficiency. On the other hand, distributed AI is interested in utilizing distributed knowledge bases. Hypotheses provide the capability to selectively locate the information and knowledge sources required. This capability is implied in the executable relations among the different attributes.

7.7.4 Using Knowledge to Control Inference

Ideally, only those inferences should be drawn that lead to useful conclusions. But this is not always possible in practice. An obvious choice needs to be made in the design of the inference strategy as characterized by the two extremes; Anomaly driven (Data-driven) and Hypothesis driven (Goal-driven) search.

Our methods are designed to combine these two approaches. The process from attributes to conclusion is data driven, while from conclusion to attributes is goal driven. Only part of the inference paths activated by evidence nodes can participate in the inference. Thus, the program only draws those inferences that are required to match the new structure to its hypotheses.

7.7.5 Memory Management in Hypothesis Generation

Another key issue which affects the efficiency of a system is the management of schemata and their instances. In this section, we will offer a technique for memory organization, for the investigation of those techniques which involve matching data structure to special purpose inference requirements (FIGURE 7.4). The knowledge engineer is expected to learn the characteristics of shared behavior in knowledge engineering activities.

In this section, we will distinguish hypothesis abstraction and hypotheses. The hypothesis generated from hypothesis space is actually an abstracted structure of hypotheses, denoted as schemata, and containing some variables.

In our experiment, hypotheses are organized according to the principle of property inheritance [111]. The relationship between hypothesis and its schema is maintained in an hierarchy table, which supports the inheritance of properties from schemata to hypothesis. The data structure for a schema differs considerably to that of hypothesis, because the schema needs to maintain all the information in its class. This hypothesis data structure thus inherits all the properties of its parent, and there is no need to replicate the generic information about its class in each of the hypotheses.

The hypothesis contains the declarative and executable knowledge which is only of use to itself, and will typically expand before it is executed. In this process, variable instantiation is effected with the support of domain specific knowledge.

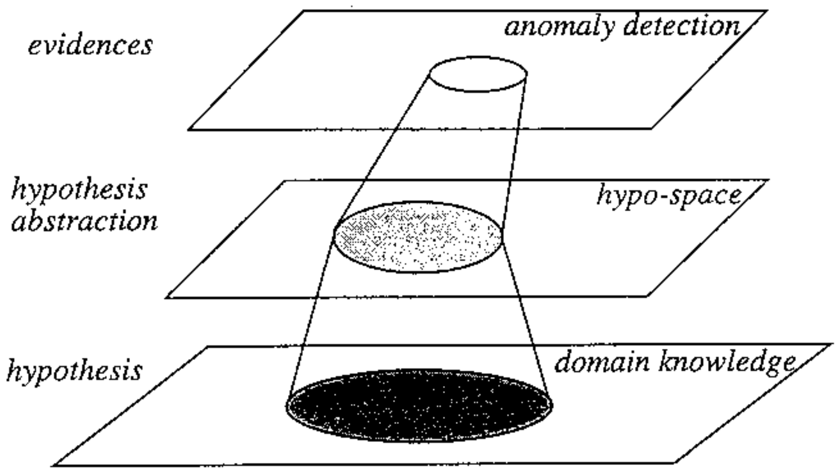


FIGURE 7.4 Stages in Memory Organization

The approach which is under investigation, is to join an individual hypothesis to schemata and to describe them by an associative link (a property-inheritance link). We identify the meaning of this link with two predicates "*instance(hypothesis, schema)*". This is accomplished by having the retrieval system search for "instance" links. In our knowledge representation approach, inheritance serves as a reusability mechanism as well as an hypothesis management mechanism[111].

7.8 Summary

A major conclusion from this study is that in the absence of support structures of the type proposed for guiding interaction between users and the knowledge based system, the powerful capabilities of the knowledge based system for problem solving will not be fully utilized. Further investigation of our approach is planned drawing on contemporary work in the use of conceptual modelling in parallel with knowledge based systems processing.

The key features of the CBI model that distinguish it from previous ones are the following:

- An hypothesis guided inference engine, based on the theory of a generalized set covering model, reduces the combinatorial explosion of abductive inference, which is a central problem in artificial intelligence.

- An integrated anomaly detection model, which is mutually complementary to the hypothesis generation model, can be used to discover not only anomalous data but also anomalous relations between data. This is a great advantage over contemporary anomaly detection techniques.
- Case based interaction provides a sophisticated interface for cooperative problem solving, which refers to utilization of knowledge from users and computer systems. Under the guidance of case based interaction, the users are active agents empowered by the system knowledge, and the systems get their auxiliary information/knowledge from the user.

CHAPTER 8

An Hypothesis Agent for Case Based Interaction

8.1 Introduction

AI research is moving away from "laboratory tasks" such as block stacking towards more realistic problems. Building autonomous agents that interact with real world software environments, such as operating systems or databases is a pragmatically convenient, yet intellectually challenging task for AI researchers. The properties of hypothesis developed in this research in a case based interaction (CBI) context make it possible to formulate a *super object*, *intelligent agent* or *autonomous agent*, which in response to an anomaly in a working environment, generates an explanation and executes actors to achieve the CBI goal and to learn from its experience.

Since "intelligent agent" is an utterance that has differing interpretations, depending on the situation, the CBI context is defined for my purposes as programs that are capable of responding to anomalous discovery by generating an hypothesis to explain the anomaly, and then creating plans for verifying the hypothesis; programs that can learn about their environment — about other agents and the person using the agent — to provide better explanations and plans; programs that are capable of altering their behaviour in response to unexpected conditions.

The proposition that CBI provides the requisite context for formalising an hypothesis agent is now examined in terms of its practical application.

8.2 Hypothesis as an Autonomous Intelligent Agent

From the viewpoint of a software engineer, hypothesis can be seen as an autonomous intelligent agent which will be called upon to perform a varying range of tasks under a wide range of circumstances. With the development of global computer networks, the number of useful information sources is increasing year by year. The autonomous intelligent agent is desired for problem solving, specifically data mining using information sources distributed over computer networks. For example, it is very attractive to retrieve the results of information analysis from a remote database by sending an agent to a remote site, at which most computations will be done. Instead of downloading a large quantity of data, this method reduces unnecessary traffic in the global networks.

An intelligent agent acting in such complex and unpredictable remote environments must be able to accommodate diverse data structures and resolve conflicts. Classical planning, in which a sequence of actions that the agent intends to execute is produced ahead of time, and reactive planning, in which the agent simply responds to its surroundings at any given moment, instead of following an explicit plan are crucial elements of intelligent behaviour. Pryor and Collins' agent, for example, falls somewhere between the extremes of classical and reactive planning.

The hypothesis agent supports both planning modes. In the hypothesis structure, the actions are planned ahead, but the option remains to accommodate new variables dynamically. After the hypothesis is generated, a small portion of the knowledge base contains customised rules for detailing the variables in the remote environment. For efficiency, it is necessary to reduce the size of knowledge base in hypothesis to its minimum, while for flexibility, it is desirable to increase the coverage of hypothesis space to some degree.

To qualify as an autonomous intelligent agent, it should have the following basic features:

- have a knowledge base and an inference control engine;
- have the ability to revise plans to cope with diverse environments; and
- ability to learn from failure.

The following features would benefit the performance of an intelligent agent:

- An agent should be able to interpolate from known changes to the database (e.g. upgrades) and what to do if database access is temporarily not available.
- An agent should be able to recognize when an achieved outcome is inadequate to realise a goal, and to create a contingency plan for such outcomes.

The hypothesis is actually a portion of the specified hypothesis space. The difference between hypothesis space and hypothesis lies in scale and in specification. In this way, hypothesis inherits the conclusions derived from the evidence.

A particular hypothesis is a customised agent derived from the defined hypothesis space. The principal difference between hypothesis space and hypothesis thus lies in the specific conclusions (results) derived from the evidence presented. Conversely, the ability to revise a plan is limited once the hypothesis is generated. However, the hypothesis (agent) encapsulated several options depending on the type of failure encountered in executing this agent. The relations between the nodes in the knowledge base determines which strategy is chosen. Failure on execution of an actor will cause a change in pre-condition or post-condition, which will then activate other actor(s) satisfying the pre-condition. Details of the actor system are discussed in [54][55].

If all actors involved in an hypothesis failed, an agent will send a failure message back to the CBI system. The system will then send other agents based on the information feedback. The learning process provided by CBI is supported by hypothesis space evolution and fusion, which were discussed in chapter 4.

8.3 Object-Oriented Design for the Hypothesis Agent

An object-oriented design for the hypothesis agent is proposed as a natural extension of the principles adapted for hypothesis generation.

Objects typically contain data, procedures and an activation mechanism, typically a message passing mechanism. An hypothesis can be seen as an extension this object construct and contains both a knowledge base and an inference engine. Properties such as information hiding, knowledge abstraction, dynamic binding and inheritance are also exhibited by hypothesis agents, as noted below:

8.3.1 Information Hiding

Information hiding is important for ensuring efficiency in meta-level reasoning such as the hypothesis generation process. It is also important for ensuring reliability and flexibility of knowledge based systems by reducing dependencies between software components. The concepts in hypothesis are contained within private variables, which are visible only to the domain of the hypothesis.

8.3.2 Knowledge Abstraction

Knowledge abstraction could be considered as a way of using information hiding. Data abstraction mechanism provides a certain degree of protection, since no direct access to the internal state of the hypothesis structure is provided. Abstractions are generally acknowledged to be desirable for knowledge based systems. A schema supported by conceptual instantiation technique provides an effective form of implementation. A schema will be transformed to an hypothesis by providing specialized concepts and values.

The purpose of using such an abstracted knowledge structure is to hide the details and maintain processing at a meta-level in the first stage of hypothesis generation. Later, at the hypothesis instantiation stage, the system has the freedom to choose relevant parameters and control structure to solve the different problem. For instance, in the schema *No-genuine theft* the system has to select a different control structure to deal with the diverse situations:

Vehicle is not recovered or *Vehicle is recovered.*

8.3.3 Inheritance

Inheritance enables programmers to create classes and their instances, in our case the schemas and hypotheses. From its classes, the instance inherits variables and methods that are appropriate to more specialised objects. In addition, the instances may override or provide additional functionality for methods inherited from a class.

Under the principle of inheritance, the knowledge which is shared by a group of hypotheses is stored in a common schema, while the instance only contains the knowledge which is specialized to itself. In this way, systems not only reduce the size of information to be stored, but the system knowledge structure is readily maintained.

8.4 Goal Driven Mechanism

The goal of an hypothesis agent is to verify itself using external data to increase certainty and to resolve inconsistencies. Research on goal-driven problem solving mechanism has been motivated mostly by computational considerations [95]. The problem of combinatorial explosion of inferences, especially abductive inference, is well known; in real world problem solving, resource and time constraints prevent consideration of all but a few of the possible inferential paths. Consequently, human experts or computer systems must quickly focus their attention and resources on pursuing those inferential paths that are most promising.

It follows that in any realistic situation, there are several different types of problem solving strategies that might be chosen, several kinds of knowledge that might be acquired, and several kinds of reformulation or reorganization of existing knowledge that might be required. Again, due to resource and time constraints, it is only practicable to perform a few of these options. Because the utility of an inference or a piece of knowledge can best be evaluated relative to a particular task or goal, goal based reasoning must guide reasoning[130]. CBI adds value to this process.

The attractions of computational models employing goal driven mechanisms are also well grounded in psychological research[87], which has addressed the cognitive basis of human problem solving and provided strong evidence for the use of strategic and goal-driven processes in many kinds of human reasoning.

The hypothesis provides a conceptual framework for a goal driven strategy and aims at integrating a diverse range of inferential problem solving strategies into a unified goal driven problem solving mechanism.

Although the generation/index of hypothesis is driven by anomalous evidence, the problem solving process is goal (hypothesis) driven.

In summary, the execution of an hypothesis agent entails:

1. Verification of itself
2. Monitoring its own performance during task execution
3. Analysis of the learning effectiveness
4. Instantiate the hypothesis during the inference process
5. Detection of relevant evidence in modification of the problem-solving strategy; and
6. Activates the strategy as a goal directed search through the (constrained) hypothesis space, but guided by the CBI principles.

8.5 Features of Hypotheses

In the methodology of case-based reasoning or object-oriented programming, the problem solving modules (cases and objects) are discrete, while hypothesis space is a continuum. The hypothesis space involves a lattice of hypotheses, partially overlapped, but including nested graphs as discrete entities. The activity of hypothesis space is dynamic and will expand according to the evidence discovered and the inference results. This coincided with the process used by experts in problem solving. Sometimes a case is remembered by existing evidences, and another case may arise as information accumulates. The original case and the new one would usually have similarities.

8.5.1 Combining Explanation with Action

In contemporary AI research, the issues of explanation and action are usually considered separately. Various agent models have generally exhibited one or the other, but not both, of these capabilities. In particular, they are apparently developing in parallel, but independently.

A competent agent should be capable of explaining anomalous evidence where possible, yet be able to react to the anomaly quickly in seeking to verify its goal, even in situations in which the current hypothesis may not be substantiated. Furthermore, the capabilities of explanation and action should be mutually supported. The action can bring additional information to support explanation while the explanation can be used to select a suitable action.

The challenge of co-existence of both explanation and execution is mainly an issue of knowledge representation. This problem has been solved by our hypothesis structure. The activation of explanation and action is managed by the higher attribute nodes in hypothesis space, and it is naturally controlled by the deduction process. This is also the property inherited from the hypothesis space.

Hypothesis is an intelligent agent, which brings explanation and action together. Hypothesis execution in a complex and unpredictable environment must be able to both explain conclusions as they change. When working in a remote environment, there is a mirror agent in the host system. This agent mirrors the changes according to the feedback of information. The user is provided with explanations provided by the mirror agent and relays information on the state of the hypothesis updates.

8.6 The Role of Virtual Relations in Decision Support

A new family of relationships between *evidence, facts, sub-conclusions and conclusions* in hypothesis space[51] has been defined as a key element of the reasoning process for hypothesis generation. This generic class of relationship is termed "Virtual Relation" and can be classified primarily as explanatory relations and executable relations, depending on the type of knowledge to be represented. They can also be used to explain the causality between two nodes and to verify the respective facts. On the other hand, virtual relations can also be classified as domain (context) dependent and domain independent, based on the original knowledge from which the virtual relations are constituted. Domain independent virtual relations usually deal with the task at a more abstract level, while domain dependency requires greater specificity.

A principal role of virtual relations [53] is to provide an inference control strategy for intelligent decision support. The topological structures constructed by weighted relations in hypothesis space represents a virtual reasoning process from facts to possible conclusions. The virtual relations demonstrate their novel capabilities most powerfully when there are some missing facts or access is available to some new facts. In this situation, the performance of many current decision support models lack discrimination or strategic knowledge.

When there is new evidence, an hypothesis generation stage is usually activated in our process[51]. In this process, the existing virtual relations will compete to explain the new evidence under the criterion of suitability. If successful, a new virtual relation will be used to connect the new fact into the hypothesis space. If it fails, the case-based interaction process[49] would be employed to establish new relations by eliciting knowledge from the user for discrimination between choices.

If there is a missing fact, (i.e., the fact is not verified), the virtual relation may guide the system to an on-line information source and use the information acquired to evaluate competing hypotheses. This process relies on a novel virtual relation, that is the *executable relation*, and provides the methods to retrieve and analyse information. By utilizing such executable relations, the system expects to verify every fact involved in an activated hypothesis. A "good" decision will be made if the hypothesis passes the evaluation, and all the facts are verified.

Of course, we cannot always anticipate every situation. Where the missing fact cannot be verified by executable relations, it may be possible to estimate the plausibility of missing facts based on other verified facts. Then, a 'weak' decision is expected to result, based on the plausibility of the missing "fact".

The importance of virtual relations lies in the expanded use of hypotheses to provide feedback of information from users or other information sources, thereby enhancing inference control strategy. Hypothesis is additionally used to draw both the system and user's attention to a more limited goal space for decision support, and therefore, avoids exhaustive search. Finally, hypotheses may be used to focus the system resources on interesting aspects of problem solving[52].

Decision support requires a sophisticated interface to facilitate dialogue between the system and user. Case Based Interaction principles support a sophisticated interface for cooperative problem solving utilising co-operating agents for knowledge acquisition from users and computer systems[49]. Under the guidance of case based interaction, the users are active agents empowering system knowledge. Strategies to conduct the dialogue include:

- the virtual relations in an hypothesis imply the required information, and only the fact linked by most "weightiest" relation will be pursued for verification;
- the priority of missing facts will be re-organized after receiving outcomes and new information;
- an hypothesis will be rejected and the hypothesis generation process will be restarted if the key fact(s) of the hypothesis cannot be verified; and
- a users' model (from novice to expert) can be used to indicate the level of abstraction required for support of users.

The power of virtual relations has been explored in our research into hypothesis generation modelling[51]. A major conclusion here is that virtual relations offer a logical approach to decision support; in particular, when the knowledge is incomplete, or a new problem is encountered. Further investigation is planned into virtual relation induction drawing on contemporary work in automated knowledge acquisition. Abductive inference schemes also appear to have a promising role in explaining new evidence.

8.7 Summary

The previous discussion shows how hypotheses can provide considerable power in problem solving and its contribution to object-oriented design. In problem solving, hypotheses can be used to focus inference and to constrain the search. They can also be used to guide the information seeking process and to make strategic choices.

Furthermore, hypotheses can be used as a theoretical device to build computational models of strategic and active inference processes. Such models have practical ramifications for the design of instructional material. Ram and Leake [130] show that the inclusion of learning goals facilitate different kinds of reasoning in different kinds of learning. Their results suggest that learning is largely a goal-directed process.

Every program encodes knowledge about the application or the domain of application. But the conventions for encoding that knowledge depend solely on programmer's ingenuity [153].

Hypothesis based (goal-driven) mechanisms have many advantages over traditional procedure-oriented languages. Information hiding and knowledge abstraction increases the reliability and helps to separate procedural and declarative constructs as complementary knowledge. Inheritance coupled with dynamic binding permits consistency in hypothesis space to be easily maintained. This has the attendant advantage of reducing the size of the knowledge base and increasing system productivity, since all shared knowledge is maintained at a higher level of abstraction

Unfortunately, our current hypothesis structure has limitations. The most obvious one is the increased complexity in implementation. At concept level, hypothesis is instantiated from an abstraction, separated from hypothesis space, then encapsulated into an entity. From the implementation viewpoint, hypothesis is interconnected with other hypotheses instead of the common use of a type hierarchy as in traditional object oriented programming.

Next shortcoming is the increasing difficulty in knowledge acquisition. In hypothesis generation, not only the declarative knowledge but also executable knowledge is required. The executable knowledge is more difficult to acquire than declarative knowledge, although the actor system provides modules for construction. At this stage, it seems that automated acquisition of executable knowledge is dependent on the effectiveness and efficiency of actor management.

In comparison with contemporary expert systems, hypothesis based problem solving is time consuming by its very nature in comparison with CBI, as it aims to solve a new problem when there is no existing solution available. It is necessary to perform some knowledge acquisition, explanation evaluation, and conflict resolution process.

Subject to these limitations, however, our approach is believed to be an effective paradigm for problem solving, especially in complex environments. The potential benefits of our hypothesis structure will no doubt, attract other researchers in pursuit of an alternate metaphor.

CHAPTER 9

Case Study: Fraud Detection

9.1 Introduction

Insurance companies are exposed to fraudulent claims of increasing diversity and scope[162]. Such claims are also characterized by collusion and ingenuity. Contemporary interest in improved methods for fraud detection[49][50][63][91][97] has focussed on the role of knowledge rich strategies, particularly the use of knowledge based systems, to complement the use of system controls and auditability mechanisms derived from risk based audit requirements.

The insurance industry spends millions of dollars every year on information systems. Despite these enormous expenditures, many companies remain data rich and information poor. Much of the data necessary for fraud control has not been translated into actionable intelligence that can be accessed by the right individuals at the right time for fraud control purposes.

Our risk analysis of the requirements for fraud detection in *Electronic Data Processing* (EDP) environments, for example, suggests that traditional approaches to fraud management would be ineffectual and that a new problem solving paradigm is urgently required. Expert systems can be designed to automatically identify fraud from appropriate indicators and to detect suspicious claim patterns. They are particularly effective in connection with "fraud rings", which routinely reuse phoney identifications, phone number; etc.

The research results from psychology indicate that the skill to apply knowledge flexibly is one of the major differences between novices and experts[87] and, based on our research, generating a suitable hypothesis is constructive in solving a difficult problem. Experts are seen to use hypotheses in several different ways for complex problem solving:

- use hypothesis as a knowledge frame. Problem solving is then reduced to the process of knowledge instantiation using knowledge either from previous cases or from new observations.
- use hypothesis as an inference strategy, which will invoke the desired inference processes with condition branches, utilising an appropriate inference engine.
- use hypothesis to guide the reasoning process and to select different reasoning paths.

In this case study, we utilise an hypothesis generation paradigm as a psychologically plausible cognitive model [132] of human expertise within the domain of fraud detection in EDP environments. The prototype developed in the case study has been adopted by a major insurance software company as a basis for field tests, and a commercial fraud detection system is in under development.

9.2 Hypothesis Generation for Fraud Detection

Our suggested paradigm for cooperative problem solving in EDP fraud detection[49] was Case-Based Interaction (CBI) drawing on earlier work by Professors' Garner and Edmonds [35]. The knowledge available for EDP fraud detection, although complicated and extensive, was categorized into: evidential reasoning and relation of the evidence to past cases; knowledge about fraud investigation techniques; e.g. in EDP domains auditing and control techniques provide a basis for conjecture/deduction; and learning abilities derived from the successful use of specific investigative techniques provides a basis for discourse with the fraud analyst!

In this case study, the scope of this paradigm has been substantially extended to general insurance fraud (e.g. false claims) through the development of Case Based Hypothesis Generation (CBHG). This research was prompted by insights derived from Schank and Leake's[145] work on creativity and learning in a case-based explainer. In broad terms, CBHG relies on both case-based explanation and the four interaction modes supported by CBI:

1. communication by users of Hypotheses to explain fraud in question i.e. analysis of anecdotal commentary on fraud circumstances;
2. clarification (transaction mapping) of initial and intermediate conditions for system verification!
3. elicitation of strategic knowledge for disambiguation of alternative fraud paths proposed by the system; and
4. analysis (abduction) of explanations given by previous investigators of similar fraud cases, based on the anecdotal commentary provided and the explanation index for past cases.

The significance of our CBHG paradigm lies in the extension of Case Based Explainer [145] to a deeper level, from explanation pattern level to hypothesis level, and from minor modification of old knowledge to a totally new view.

The benefits of providing a client server architecture to integrate access to multiple data sources with the hypothesis generation paradigm will be evident from the management intelligence system architecture shown in FIGURE 9.1. Ongoing research into diverse fraud scenarios are expected to define the limitations of our current paradigm, but the evidence todate fully vindicates the value of the five components of the integrated structure; namely, anomaly detection model (ADM), hypothesis generation modelling, interaction modes selection(CBI), data mining and intelligence synthesis.

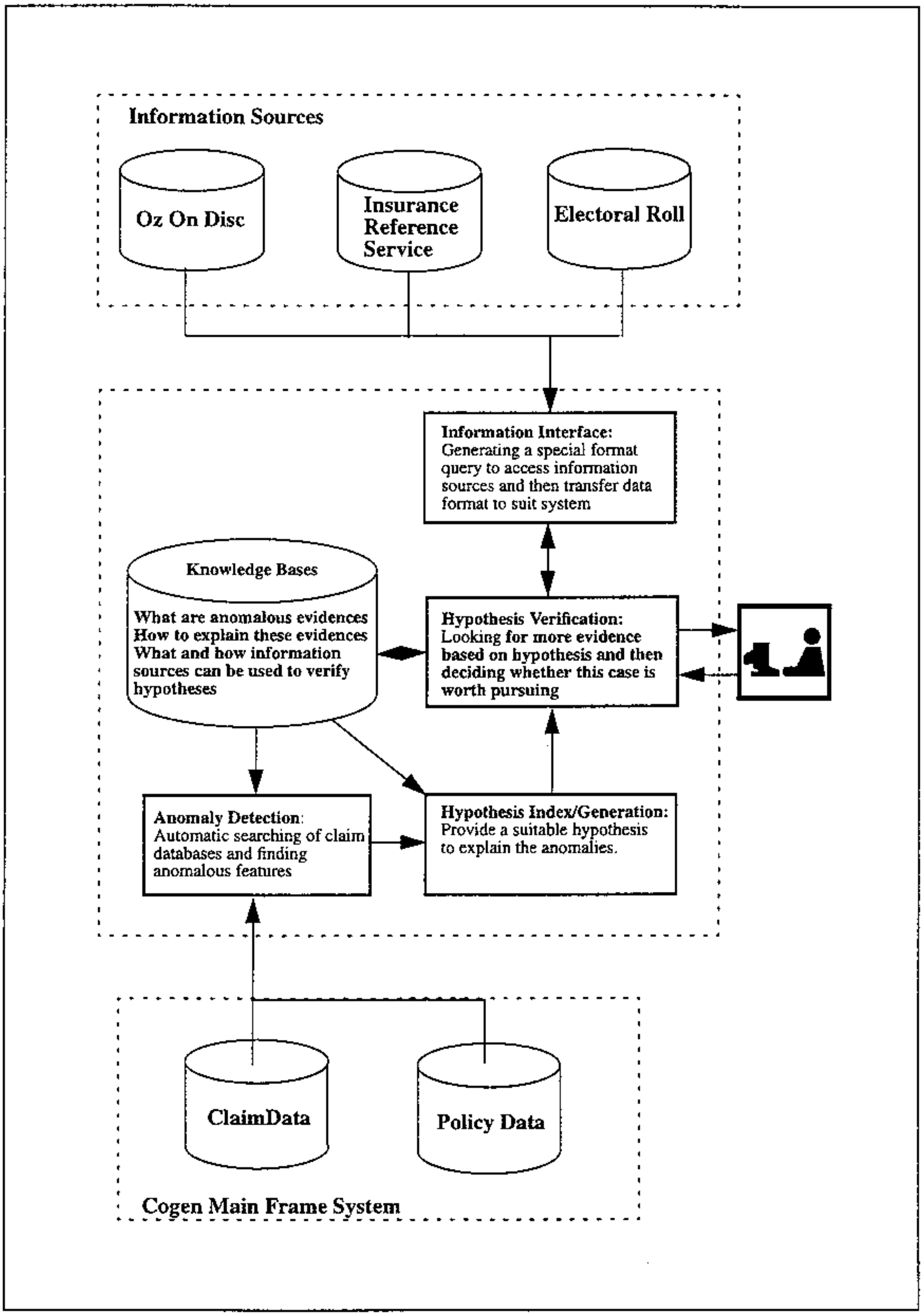


FIGURE 9.1 System Structure for Fraud Detection and Discovery

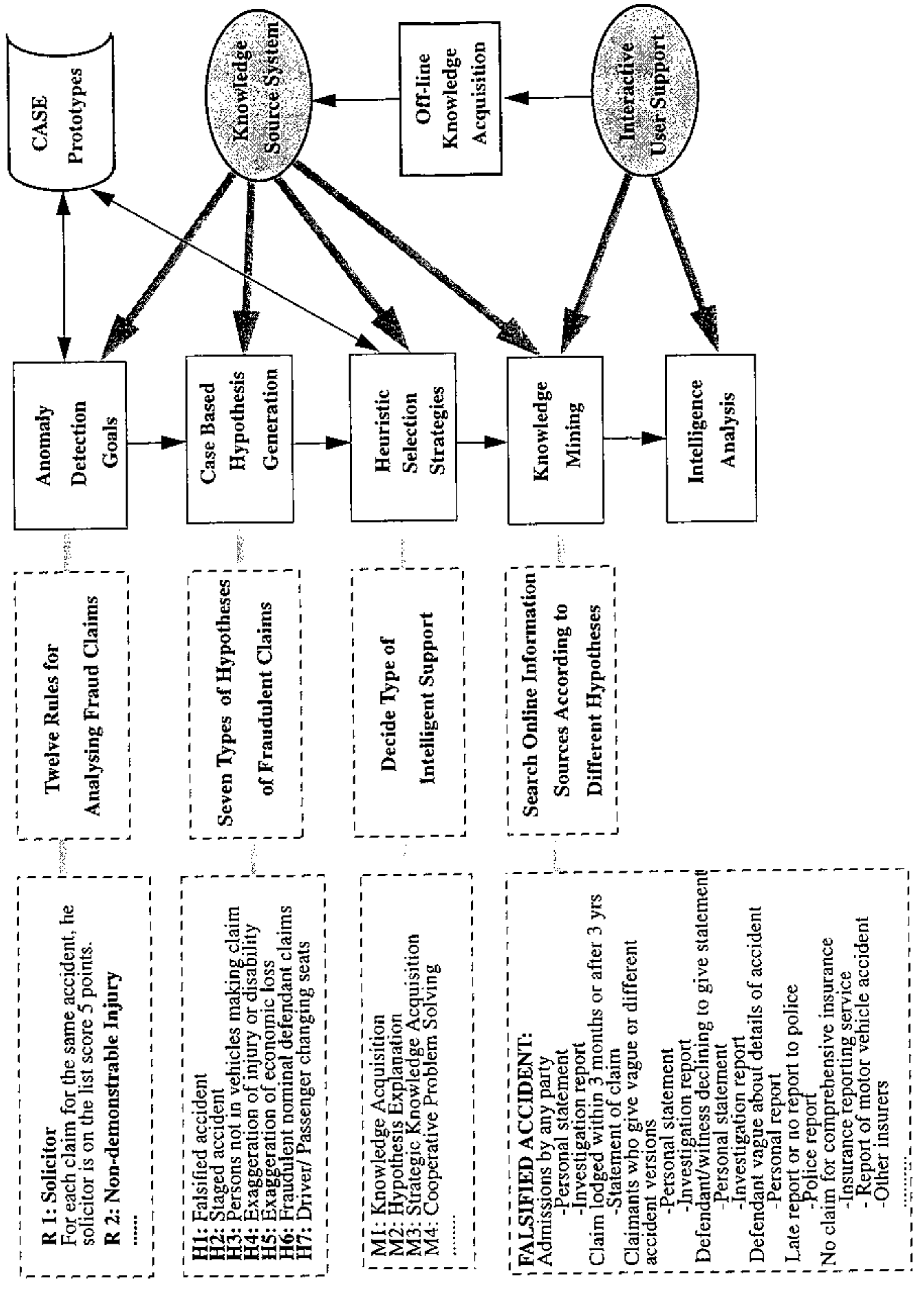


FIGURE 9.2 Structure of hypothesis Based Insurance Fraud Detection

9.3 System Control Strategies

As noted in section 9.2 and illustrated in FIGURE 9.1, there is a need to provide a client server architecture to integrate access to multiple data sources with the hypothesis generation paradigm. The successful application of this paradigm in practice depends no less, however, on the availability of an adaptable control structure.
The following control requirements are provided by the system.

9.3.1 Agenda-control

The diversity of tasks in fraud detection, especially in ad hoc situations, requires a flexible control structure. Our approach offers the requisite flexibility through the synthesis of different levels of control structure. Agenda control is concentrated at a high level. Its task is selection of the appropriate module. Details of the process and hypothesis verification are secondary issues.

The *Management Intelligence System* is generally controlled by a recursive algorithm whose Pseudo code is as FIGURE 9.3:

```
BEGIN
    IF fraud-evidence-found = false
    THEN Anomaly detection model
    ELSE Index-hypothesis (evidences)
        IF hypothesis exists
        THEN retrieve-hypothesis(index)
        ELSE Hypothesis-generate-model (evidence)
            Hypothesis-manager( new-hypothesis, evidence)
            Update-hypothesis-space (hypothesis)
        END
        Execute(hypothesis)
    END
  END
```

FIGURE 9.3 Algorithm for Insurance Fraud Detection

9.3.2 Hypothesis-control

The control structure involved in building new hypotheses (cases) is derived from the strategic knowledge acquisition processes [95]. Control complexity is usually due to the number of alternative actions available to achieve intermediate goals/alternative sub-plans, or is due to the different evidence requirements to activate a case.

Hypothesis-control structure is represented by strategic rules specialized by the domain expert and can encompass various states of the problem domain.

Hypothesis control is concentrated on hypothesis verification, on the selective activation of actors, is responsive to agenda control and in charge of process management. The hypothesis control structure is derived from hypothesis space. Data (evidence)-driven instantiation processes occur when the hypothesis is executing.

9.3.3 Process-control

In an hypothesis, *actors* are used to perform the requisite actions (activities). Actors are entities that combine the properties of procedures and data (or knowledge). The communication method in the *Actor System* is *message passing.* The actor can be activated by receiving a message and can activate other actors by sending messages.

The message is made up of two components. The first is the name of the actor that is going to receive the message, while the second component is the message itself.

The diversity of tasks in fraud detection requires flexible control structures. Our approach offers the requisite flexibility through different levels of control.

When an actor is activated, the process will be guided by the control structure inherited from its superclass. With the current implementation of our actor system, the relationship between a class and superclass is maintained in the Inheritance Hierarchy Table [151][156]. This relationship is maintained to enable an actor to inherit micro-control structure from its superclass.

9.3.4 Dialogue Control

Our particular form of modelling in HCI is known as "hypothesis based interaction" which aims to facilitate a system model of co-operating agents, not just provide a set of discrete commands/responses and questions/answers constrained by a preliminary goal. The approach to control dialogue is under the guidance of the respective hypothesis, i.e. controlled by the hypothesis logic (implied in relations) instead of being derived from a fixed menu system.

The functions of dialogue control are:

1. Selecting the information sources: internal or external;
2. Providing the control privileges when needed by the user;
3. Identifying the user's profile and providing suitable assistance;
4. Querying the end-user when crucial information is missing; and

5. Natural language processing.

Process control, which is an integral component of all actors, is in charge of detailed steps of the process, including database dial-up, data retrieval and invokation of analysis algorithms. When an actor is activated, the process will be guided by the control structure inherited from its parent. With the current implementation of our actor system, the relationship between an actor and its parent is maintained in the *Inheritance Hierarchy Table* [95][156].

9.3.5 Inference Strategy in Hypothesis Execution

Components of the inference strategy work together to produce a solution. FIGURE 9.4 shows the following components:

- Control: Executing agenda statements.
- Backward Chainer: Resolving evidences.
- Forward Chainer: Monitoring data/evidence, and invoking rules and hypotheses.
- Execution: Executes rules and hypotheses.

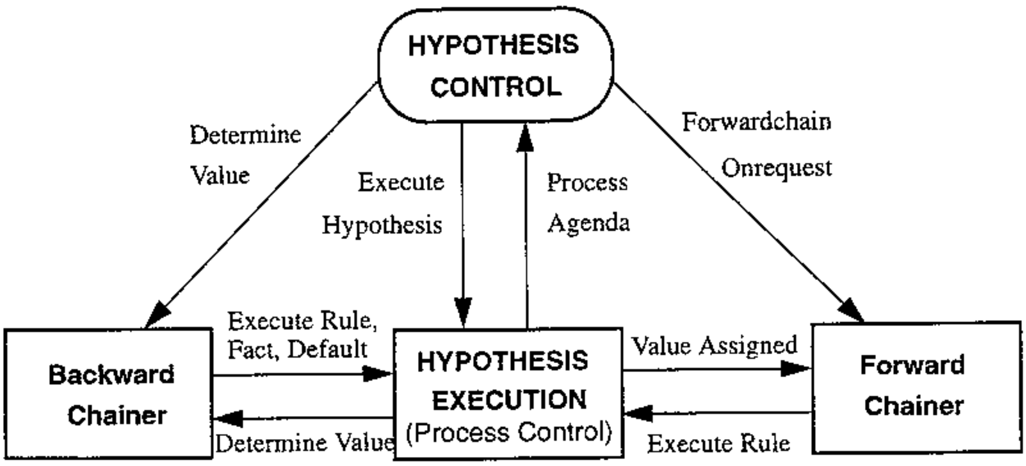


FIGURE 9.4 Inference Engine Model

Interaction between the components may be summarised as follows:

- *Hypothesis Control* executes agenda statements. The *Control* calls the backward chainer to resolve the evidence which is needed to execute an agenda statement. The *Control* calls *Execution* to process the actors contained in the hypothesis, and calls the *Forward Chainer* to fire appropriate rules.
- *Backward Chainer* processes parameters and evidences. It calls *Execution* to execute actors, rules, and facts for the resolution of missing evidence.
- *Execution* calls the *Forward Chainer* whenever it assigns a boolean value to evidence. *Execution* calls *Backward Chainer* when it needs to determine the value of evidence.
- *Forward Chainer* monitors changes to data and maintains a list of rules that can be executed. When a rule is ready to execute, the Forward Chainer calls Execute.

The inference strategy for rule processing determines the degree of interaction during a consultation. For example, *Forward Chainer* may not participate in the processing.

Agenda statements also affect component interaction during processing. For example, you can control the actions that the inference engine takes when resolution is prolonged.

9.4 Anomaly Detection

The *Anomaly Detection Model* (ADM) [55], which was developed following extensive discussion with fraud detection experts, provides a unique way to detect anomalies automatically in insurance claims.

The anomaly detection model (ADM), integrated with the hypothesis based problem solver, is used to scan and detect anomalous claims in the claim database. Two types of anomaly are addressed in this case study; outliers from normal distribution models or unlikely events, however defined, and anomalous relationships, which are 'strange' relations within the ordinary data. In the domain of insurance fraud detection, anomalous attributes could be:

1. *accidents happening at midnight,*
2. *claims with very big losses,*
3. *vehicle theft just before registration expired,*
4. *vehicle theft in first year of insurance,*
5. *vehicle stolen after the insured purchases another car,*
6. *rented vehicle involved in an accident.*

Anomalous relations may be between data in the same database, or with data from different databases. Here are some examples:

1. *vehicle involved in accident has previous claims,*
2. *two parties involved in an accident were known to each other,*
3. *vehicle, with third party property damage plus theft and fire, involved in single vehicle accident and the driver has decamped,*
4. *the stolen vehicle had a 'for sale' advertisement recently,*
5. *the claimant had financial difficulty recently.*

To deal with these two types of anomalies, the detection stage is separated into two steps, each of them utilising different scanning strategies. The first step is a classic rule-based inference for anomalous data detection, and the second step is a novel hypothesis-guided process for anomalous relationship detection. Another reason for this separation is that the determination of such relationships is very costly. For instance, to verify *vehicle involved in accident has previous claim,* a dial-up to the database in *Insurance Report Service* through a modem is necessary. The cost will be 25 cent telephone line connect and about 5 dollars[1] for each search. Thus, a relationship search can be used to reduce the number of suspicious

1. The cost is dependent on the agreement with IRS.

claims. The aim of the Anomaly Detection process is to narrow the scope of the claim data, and then, to investigate the summary information drawing on the applicable hypothesis.

Anomaly (data) detection entails recognizing any data which appears to deviate considerably from normal reference samples. There are various methods to deal with it[50][91][97]. In order to take advantage of statistical methods[3] to provide a flexible detection process, our rule-based strategies which are incorporated with the statistical techniques, are utilized to process the data structures most commonly used in the insurance domain.

The outcome of anomalous data detection is a listing of the claims with a score, which is linked to the suspect attributes through the defined threshold, together with any evidence detected (usually it is not strong enough to prove the hypothesis). At the next stage, the scope of the claims which should be investigated are narrowed.

Hypothesis guided detection process will look for new relations between the data by means of the heuristics implied in the hypothesis. Heuristics are activated by anomalous evidence. Some examples are shown below:

1. Historical pattern of events (fraud pattern);
2. The linkage with other possible evidence;
3. Failure of field studies to verify purported evidence;
4. Credibility of sources of conflicting facts supporting the evidence;
5. The experimental methods for examining evidence.

Using the heuristic knowledge represented by an hypothesis and the information available from on-line sources, the anomalous relation detection process is able to identify unusual relationships.

9.5 Hypothesis Generation Process

In the process of hypothesis based problem solving, the system indexes the knowledge base of existing hypotheses for problem solving. Through the instantiation, modification, and verification of the selected hypothesis, a suitable explanation would be provided. Hypothesis generation would be necessary, however, when, in problem solving, there are no existing hypotheses which can be utilised to explain new evidence.

The current hypothesis space involves the following fraud schemes:

1. *Dumping vehicle after devaluation (loss of value),*
2. *Dumping vehicle after failing to sell,*
3. *Phantom accident,*
4. *Change drivers,*
5. *Multiple claims, and*
6. *Organised crime ring.*

In the process of case based hypothesis generation (CBHG), conceptual graph [151] based abductive reasoning [117] is playing a central role. Although abduction lacks rigour in theoretical terms, our CBHG has provided new insights, and plausible hypotheses have been discovered. The following process of hypothesis generation (refer 1 - 4) begins with a set of anomalies.

1. Using evidence to index explanation patterns;
2. Using maximal join operation to generate plausible explanations;
3. Instantiating the explanation patterns;
4. Hypothesis synthesis and conflict resolution.

In the first step, all anomalies are used to index a set of explanations, which provide the basis for generating a new hypothesis. The explanation index is based on similarity measurement between the concepts in current evidence and the hypothesis index. Both evidence and the hypothesis index are represented as conceptual graphs. The similarity measurement is the sum of the semantic distances of corresponding concepts [47]. The threshold of the similarity is adjusted dynamically to balance precision and extent.

The conceptual graph operation *Maximal-join* [156] has been used as an operation for both abductive reasoning and to provide possible hypotheses for a set of evidences. One of the major problems caused by the max-join operation is that of generating too many (redundant) hypotheses. In order to overcome this problem, all the explanations are arranged into a tree structure from generalized to specialized. Every explanation in the tree structure can have two states: active and inactive. The state is decided by the current evidence. In this way, only those explanations activated by the current evidence will be involved in max-join abductive operations, and resultant graphs (candidate hypothesis) are fewer. Additionally, constraints stemming from canonicity and conformity requirements increase the likely credibility of the hypothesis.

Thirdly, because the hypothesis is an abstract structure, it needs to be related to the application domain. This instantiation process is supported by the domain knowledge bases to fill the slots or to specify the relevant concepts in the explanation. The operation is based on the *Concept Type Hierarchy*[151], and may need to refer to type definitions or to schemata for concept specification or substitution.

Finally, a resolution process has been employed relying on an *Explanation Believability* measure to detect conflicts. A simple rule-based strategy is used to classify the type of conflict; principally plausibility and vagueness failures. To resolve these conflicts, the system provides resolution strategies based on concept substitution, generalization, specification, and user interaction.

9.6 Interaction Mode Selection Strategies

To provide intelligent support for hypothesis based problem solving, an intelligent interface is needed to perform communications between user and computer. The interface provides four types of basic interaction mode, and can be changed dynamically according to the communication requirements during hypothesis execution.

9.6.1 Composition of the four interaction modes

Through the CBI research, we have shown that four fundamental interaction modes [49] are important for cooperative problem solving in management intelligence systems. The modes are declarative knowledge acquisition, hypothesis explanation, strategic knowledge acquisition, and cooperative dialogue.

Every hypothesis can be attached to a different interaction script, which relates to the specific needs and circumstances. The interaction mode appropriate to each hypothesis reflects the requirement for a dynamic mix of the four basic interaction modes, and entails a dynamic control structure, which will change according to progress in the hypothesis verification process. This kind of dynamic control structure provides a unique human computer interface for our hypothesis-based problem solving technique.

9.6.2 The strategy for selection of interaction modes

There are several facts that affect the interaction scripts, such as the information needed to verify hypotheses, actors [95] involved in hypothesis, and evidence to hand. In our hypothesis paradigm, the interaction modes scripted for an hypothesis reflect the above factors.

In the process of hypothesis execution, a queue of candidate actors is created and a queue of corresponding candidate interaction modes will be created simultaneously. The queue of interaction modes may, of course, change to reflect changes to the queue of actors.

9.6.3 Advantages of flexible interaction modes

In the traditional menu-based approach, the user selects the interaction mode. The menu limits the actions that the user could perform at any one stage. On the other hand, case-based interaction is guided by the case logic rather than by the user. The interaction process attempts to explicate the user's domain knowledge to select and instantiate a plausible hypothesis to achieve the user goal.

9.7 Data Mining Process

We have currently reached the stage where many widely accessible information resources are available. Full text access to business information remains one of the fastest developing areas for on-line information searching. Because of this growth in accessible information, the network community has begun to show a great deal of interest in the location, retrieval, and analysis of network information.

In hypothesis generation, we make full use of on-line information sources to fill the knowledge gaps. Data bases which have been used or could be used for detecting anomalous personal relationships are shown in TABLE 9.1.

Although these databases provide routine commercial information, they can be very useful for verifying information, or to acquire missing knowledge. Because of the variability of the data structure in these databases, however, it is necessary to convert the query data into a common data structure; conceptual graphs in our case. This knowledge mining process thus involves: data format investigation, data base query generation, information retrieval, and translation into conceptual graphs.

TABLE 9.1 Typical Information Sources

DATABASE	CONTENTS
Aus. Security Commission	Directors, major shareholders and executives
Electoral Roll (CD-ROM)	All registered voters
Electronic White Pages (WWW)	Name, address, phone no contained in white page.
Insurance Report Service	Most insurance claims reported
Land Titles	Property ownership, price
OZ on Disk (CD-ROM)	Name, address, phone no.
News Classified (WWW)	Classified advertisements in nine newspapers

The knowledge mining process involves:

- Data format capture
- Data base query generation
- Relevant information retrieval
- Conceptual graphs composition

After the information retrieval process, the hypothesis in working memory will adapt to the new information retrieved.

9.8 Intelligence Synthesis

Intelligence synthesis is the analytical process that transforms the disorganized, confused, and sometimes contradictory stream of business related information into relevant, accurate, and usable knowledge. The synthesis process also provides conflict resolution, which is based on three approaches. The first is a statistical counter, the second is a priority table of actions, and the third involves the human expert. Intelligence is the product resulting from the collection, evaluation, integration and interpretation of all available information and concerns one or more aspects of explanations derived from the evidence.

The proposed intelligent system thus requires a model for memory (hypothesis) management and a process monitor for knowledge processing, including access to an actor model. The blackboard structure has been adapted for our requirements.

Implementation of the intelligence function requires a working "memory", which will be able to satisfy a range of functions to ensure the intelligence analyst meets the established design (synthesis) criteria. A blackboard structure is the chosen model for a central working memory for intelligence synthesis.

The original blackboard concept is credited to A. Newell [110], being advanced as a knowledge system architecture having the flexibility to enable general reasoning to be performed. Control of the blackboard is by a problem solving strategy implemented by a scheduler.

In the hypothesis generation process, a blackboard scheduler is responsible for selecting the best explanations to resolve conflicting evidence. This supports a dynamic, dialogue driven system, where, not only is there the problem of best explanation, but also, the task of keeping track of any changes made during the synthesis process. The challenge at this stage is the intellectual sifting of diverse, often conflicting strands of fraud-related information to find meaningful patterns.

9.9 Example: Intelligence Analysis of Fraudulent Insurance Claims

The conceptual structure shown in FIGURE 9.1 has been used to study fraudulent insurance claims experienced by a major Australian insurance company. The knowledge acquisition phase conducted with the *Special Claims Unit* and *Motor Vehicle Claims* department enabled the existing fraud detection process to be defined, and known frauds (successful prosecutions!) were used to extract key fraud attributes for testing the fraud detection system. Novel cases for evidential search have been constructed from our CBI model, and typical rules under examination for the detection of fraudulent claims have been incorporated into the system.

In our prototypical model the rules for initial evidence detection (anomaly data) are exemplified as:

Rule 1: IF both parties have same background THEN add 5 points to fraud score.
Rule 2: IF theft with TPPDO insurance THEN add 35 points to fraud score.
Rule 3: IF vehicle is a hire car THEN add 10 points to fraud score.
Rule 4: IF vehicle is over 10 years old THEN add 10 points to fraud score.
Rule 5: IF vehicle is 5 to 10 years old THEN add 5 points to fraud score.
Rule 6: IF vehicle is unidentified THEN add 40 points to fraud score.
Rule 7: IF it is a single vehicle accident THEN add 40 points to fraud score.
Rule 8:......
Rule 101: IF fraud score < 39 THEN suspect = clear.
Rule 102: IF fraud score > 40 THEN suspect = dubious.
Rule 103: IF fraud score > 60 THEN suspect = obvious.

This type of rule is consistent with current practice in Australian insurance companies. Other types of rule are more knowledge intensive, and are based on forward chaining inference:

Rule 1: IF the policy is TPPDOPT (third Party Property Damage Only Plus Theft)
AND the claim is theft claims
THEN Motivation to cover loss is obvious
Rule 2: IF accident time is difficult to find witness
OR accident location is difficult to find witness
THEN description of accident is dubious
Rule3: IF vehicle was stolen in the front of insured home
AND the garage was not occupied
THEN the theft was dubious

Rule 4: IF theft vehicle left at the scene of accident
AND the driver was not identified
THEN the theft was dubious
Rule100: IF the motivation to cover the loss is obvious
AND description of accident is dubious
AND the theft was dubious
THEN the claim is very likely fraudulent

The test results show that the second type of rule narrows the scope for fraudulent claims. Experts on claim investigation agree that it is specifically suited to patterned investigation and could be further developed towards claim processing automation. The first type rule will provide more freedom of decision making, especially by experienced investigators, and also provides more opportunities to discover new fraud patterns.

9.9.1 Results from Anomaly Detection Model

The purpose of the anomaly detection model (ADM) is to limit the quantity of claims, and to send suspicious claims to the hypothesis generation model. Assuming the evidence of a particular (suspicious) claim is:

- *Time when it would be difficult to find witness,*
- *The car was allegedly stolen in the front of insured house*
- *No report to police.*

When these evidential graph nodes were input into the hypothesis space, the hypothesis '*dumping vehicle after devaluation*'[1] is activated by forward propagation.

But the reasoning process does not stop there, and the system continues propagation from top to bottom. In insurance business practice, some information is not available in the database, and human investigators will be sent out to check the evidence. The results from investigation will then be input to the system at a later stage.

Back-propagation produces additional evidence:

1. Scenario: The insured received a bill from insurance company before the current policy expired. The insured noticed that the vehicle (insured) value has been reduced; i.e. devaluation! Thus, the insured dumped the vehicle and claimed for the original value.

- *Single vehicle accident*
- *No forced entry,* and
- *Steering lock is intact.*

At the next round, the system uses six pieces of evidence (instead of three) to forward propagate from bottom to top again. If the resultant hypothesis from this round is the same as the last one, this hypothesis is the final conclusion. Otherwise, the system will continue processing until no further evidential node can be activated.

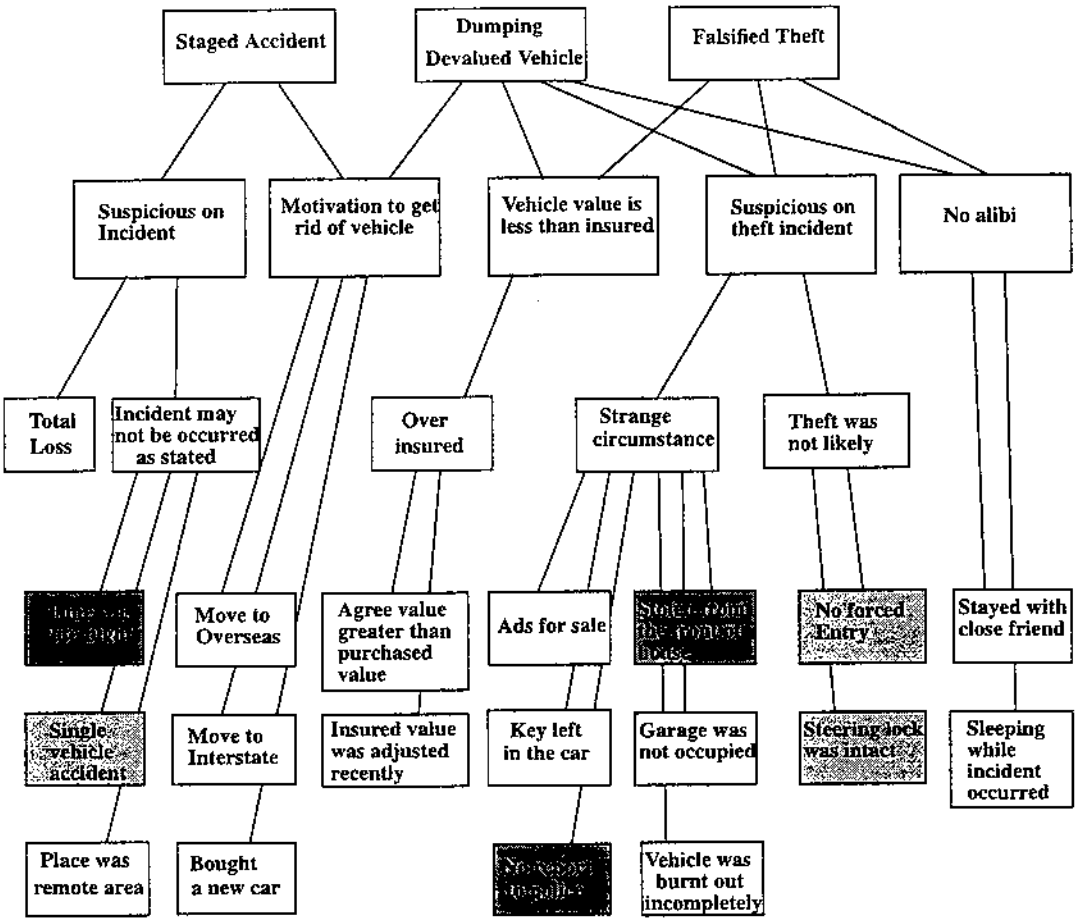


FIGURE 9.5 Action in Hypothesis Space

9.9.2 Evidence Discovered

After several rounds of forward-backward propagation (it is two rounds in this example), the evidence found by the ADM model is:

- *Time when it would be difficult to find witness*

- *The car was stolen in the front of insured house*
- *Garage was not occupied*
- *No report to police*
- *Single vehicle accident*
- *Steering lock is intact*
- *No forced entry*
- *Sleeping while incident occurred*

After several simple manipulations such as deleting irrelevant evidence/conclusions, the active hypothesis space (as shown in FIGURE 9.6) is the basis for generating a new hypothesis.

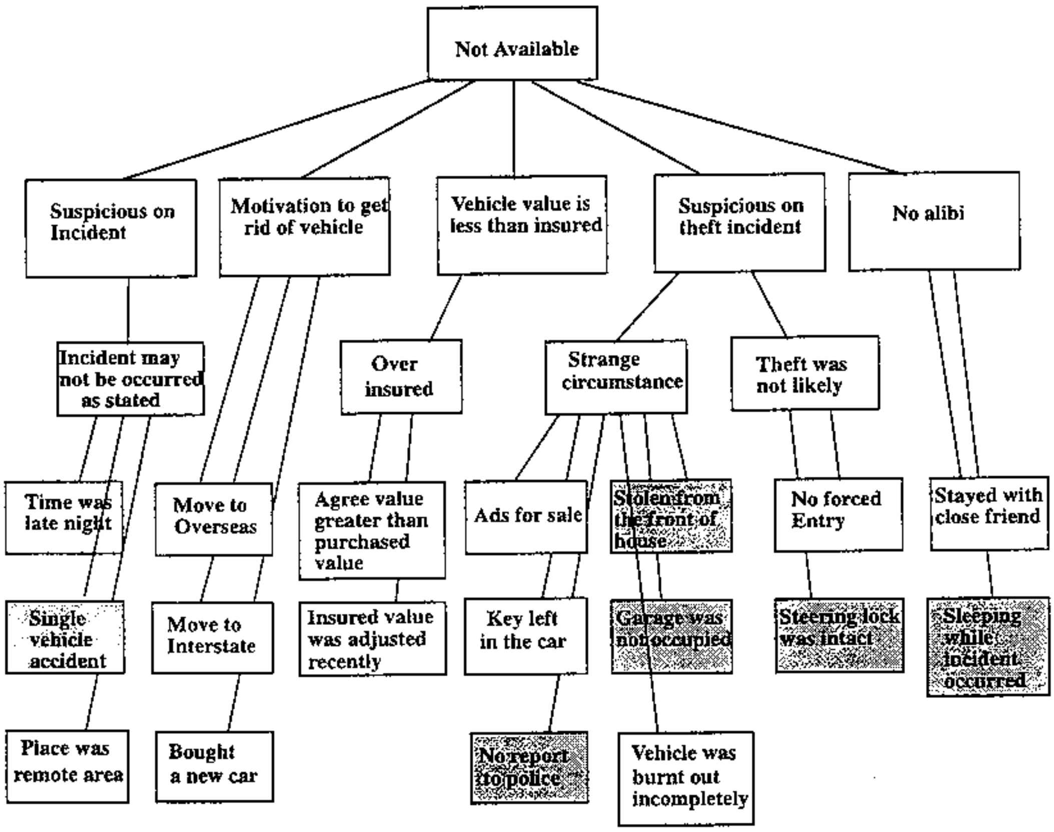


FIGURE 9.6 Part of Hypothesis Space Used to generate Hypothesis

9.9.3 Hypothesis Generated

The key operation in hypothesis generation is abductive reasoning. The objects of abduction are the evidential graphs and existing explanations.

The draft hypothesis reveals key evidence (or requirements). If and only if such kinds of evidence are verified, is the hypothesis justified. For instance, an hypothesis "*Falsified Theft*" must meet requirements: *The Policy Is Third Party Property Damage Plus Theft and Fire* or *The Policy Is Comprehensive*. Hypothesis *Abandon Vehicle After Accident* must meet the evidence *Single vehicle accident*. If one of the key pieces of evidence is not satisfied, hypothesis synthesis will be executed.

The shadow part of hypothesis space shown in FIGURE 9.7 is constructed by the user as a supplementation of knowledge acquisition based on the information from the investigation report. This kind of knowledge acquisition is supported by case-based interaction.

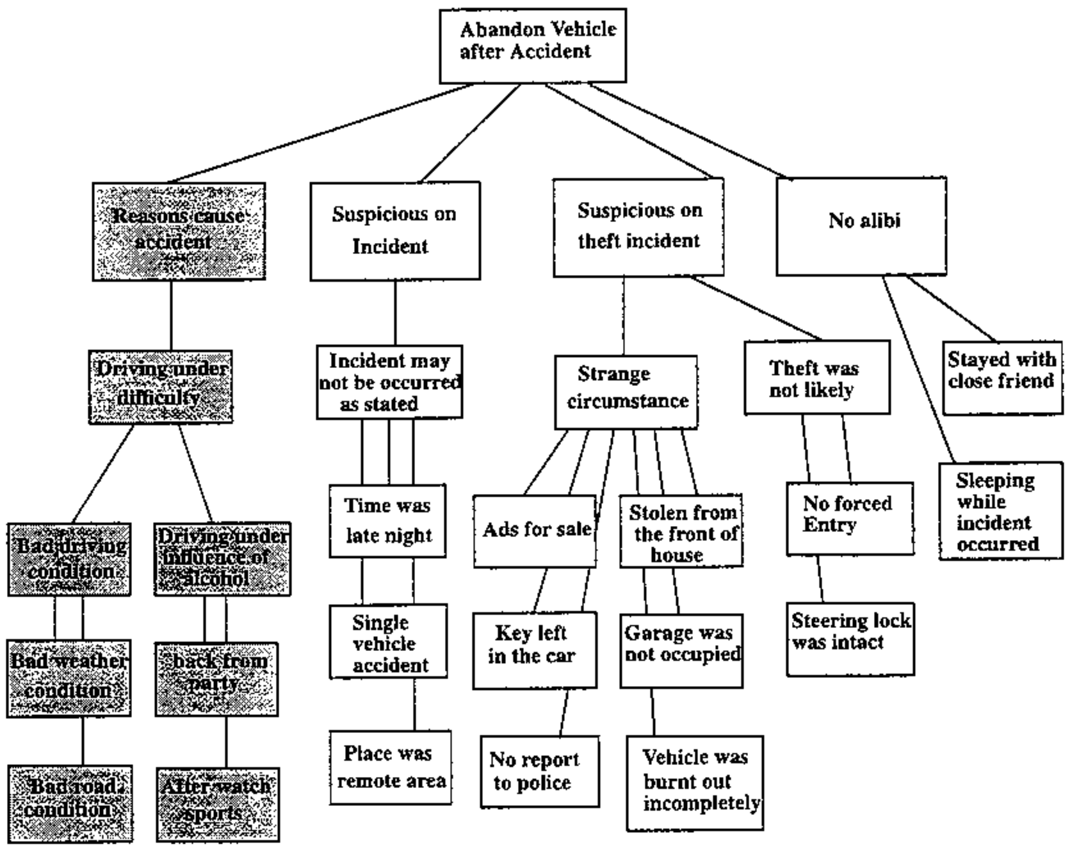


FIGURE 9.7 Interim Hypothesis Generated

9.9.4 Strategies Used in Intelligence Synthesis

The principal function of hypothesis synthesis is to explain the unexplained facts. From the evidence obtained, more than one explanation might be triggered. A resolution strategy is incorporated in an explanation mediator, which will rule out any explanation that violates observed or deduced facts.

To detect contradictory explanations, three methods were employed:

- Knowledge abstraction: If two explanations have the same abstraction [173], it is believed that they are contradictory, or at least one of them is redundant.
- Explanation pattern: If two explanations have the same explanation pattern [159], it is believed that they are contradictory, or at least one of them is redundant.
- Hypothesis Revision Heuristics: To make a candidate hypothesis more acceptable, a set of heuristics will be chosen to examine the effect of hypothesis revision. Some exemplars employed by CBHG are Concept reversion, Concept generalization, Concept specification, Concept substitution, and Relation substitution.

CBHG provides six contradiction resolution strategies, which may be used individually or in prioritized combination. The strategies are as follows:

- Match most: triggers the explanation which matches most evidence.
- Left least: triggers the explanation which has the least unmatched evidence.
- Recent used: triggers the explanation most recently used.
- Recent not used: triggers the explanation least recently used.
- Antecedent ordered: triggers the explanation which has the highest antecedent priority

- Consequent ordered: triggers the explanation which has the highest consequent priority.

To understand how these methods work in combination, consider the ranked combination: (1) Match most, (2) Left least, and (3) Recent used.

Strategy "Match most" will be applied to the contradictory explanations. If more than one explanation remains, the "Left least" strategy will be invoked. If there are unresolved contradictions at this point, strategy "Recent used" would be expected to produce the final explanation. Hypothesis generation guided by this revision process makes it possible to explore evidential conclusions more thoroughly.

9.9.5 Final Hypothesis Generated

The particular hypothesis generated by this system, based on the anomalous data and relations deduced, may be stated as follows:

Hypothesis: Abandon Vehicle after Accident.

Feature: The vehicle was involved in a single vehicle accident caused by careless driving. Since the policy related to this vehicle was '*Third party property damage plus theft and fire*' the loss was not covered by insurance. Since there is no witness to the accident, the insured alleged the vehicle was stolen.

Evidence: The vehicle was allegedly stolen from the front of his home. The vehicle was then involved in a single vehicle accident. At the time of the accident, it was difficult to find a witness. The policy is "*Third party property damage plus theft and fire*".

Further Instruction:

- ask for an explanation of 'why the car was parked in the front of the house while the garage was not occupied'.
- verify the activities of any related person at the time of accident.

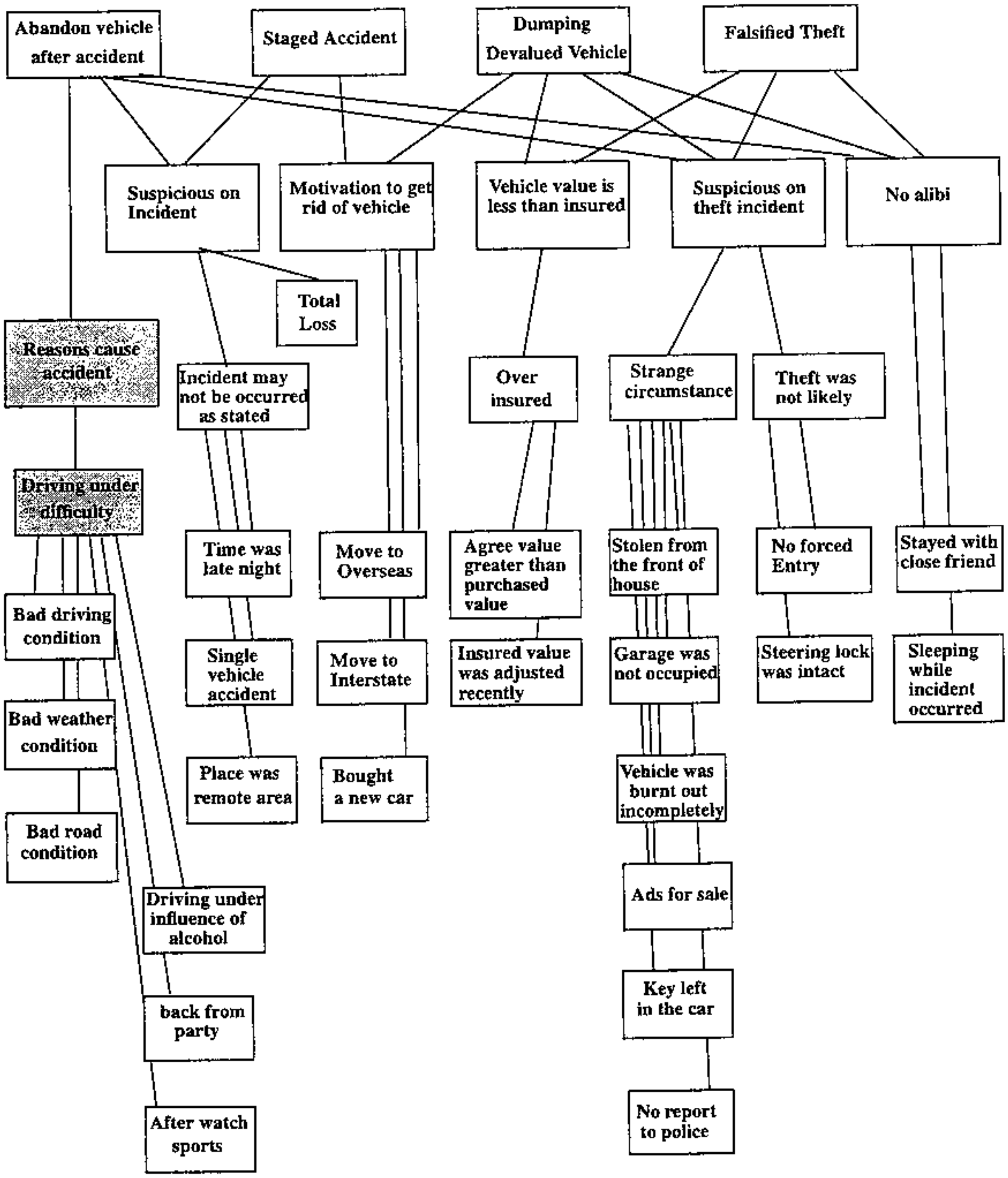


FIGURE 9.8 Final Hypothesis Incorporated into Hypothesis Space

The generated hypothesis may still reveal some inconsistencies, but it is reasonable. It can be passed to a solicitor for further consideration, and be used as a rule in anomalous data scanning to avoid future loss.

9.10 Fraud Detection as Knowledge Discovery: A Conclusion

In this chapter, a case study was presented for the new hypothesis generation paradigm and its application to insurance fraud detection. While the results of this study are very promising, there are some issues which are still outstanding. From the authors' standpoint, CBHG offers a novel methodology for complex problem solving and for Knowledge Discovery in Databases.

Knowledge discovery has undergone explosive growth in recent years [40][42][43]. The rapid growth of many business, government, and scientific databases has far outpaced our ability to explain this data, thereby creating a demand for new tools and techniques for automated and intelligent database analysis[41]. Knowledge discovery through database searches is the overall process of detecting and preparing data, selecting projections, selecting data mining methods, extracting patterns as potential 'knowledge' and consolidating knowledge.

My research has directly addressed the key issues of knowledge discovery, focussing on the search for missing data, verification of complex relationships, pattern understanding, and user computer interaction. The new CBHG paradigm promises a new era for knowledge discovery.

In this study, the anomaly detection model is a data mining tool for fraud detection, comprising a particular data mining algorithm and rules that, under the acceptable computational efficiency limitations, produces a particular enumeration of evidence. Hypothesis generation is mainly concerned with how to extract (identify) *New discoveries* and involves the selection and evaluation of possible patterns, especially unfamiliar patterns. Conflict resolution provides a way to consolidate discovered knowledge (new fraud scheme) and to integrate it with the existing knowledge base (hypothesis space).

The capability of this fraud detection system depends upon the accumulation of actual cases suitably classified by fraud type. With our model (current hypothesis space), most of the common anomalies in the motor insurance domain can, we believe, be detected and explained effectively.

The most obvious limitation of CBHG is its slow response to unfamiliar situations, and a sophisticated mechanism is required to build up hypotheses automatically. These are topics for further research.

Finally the omission of key facts in a case has led us to examine a novel hypothesis generation paradigm, with particular emphasis on the use of conceptual graph based abduction. The impact of this extension will bring some degree of creativity to our current anomaly detection model.

A commercial fraud detection system, currently under development, avoids these limitations by providing a static knowledge base which is fully implemented. The hypothesis generation process is carried out only in the master system, which works on data mining, claim analysis, knowledge acquisition and hypothesis generation.

CHAPTER 10

Conclusion and Future Direction

Research reported in this thesis, into hypothesis in human problem solving and the design of an hypothesis generation model for management intelligence systems, encompasses a wide range of contemporary studies on knowledge engineering.

In this research, the author initially reviewed developments in contemporary information systems for management. Then, the need was explained for a management intelligence system to utilize and integrate various intelligence derived through the information superhighway.

The omission of key facts in case based explanations provides the motivation to examine a new paradigm for hypothesis generation. Novel results have been reported, with particular emphasis on the use of conceptual graph based abduction and conflict resolution This paradigm will bridge the gap between reality and the system's knowledge base.

The hypothesis guided problem solving process elaborated in this thesis is believed to represent a plausible cognitive model for complex problem solving. The resulting paradigm required the introduction of several key concepts that are applicable to the construction of any hypothesis generation model:

- anomaly detection,
- abductive inference control in hypothesis generation,
- conflict resolution and hypothesis evaluation, and
- case-based interaction.

The generalised hypothesis paradigm provides an attractive reasoning capability for complementing other problem solving paradigms.

10.1 Contributions

In this thesis, we have described how an hypothesis generation model solves some of the theoretical and technical problems of building a management intelligence system. The theoretical issues tackled include hypothesis representation, evolution of a dynamic hypothesis space, and the knowledge requirements for controlling both abductive reasoning and the conflict resolution strategy. The technical problems addressed included anomaly detection, online information analysis and knowledge acquisition. The proposed solutions are based on the hypothesis generation framework.

In summary, the results from this research include several original contributions in interactive knowledge engineering.

- Hypotheses support cooperation between human and computer systems. Cooperation between human experts and computer systems is important when solving complex problems, especially when the system doesn't have an existing base case. Using hypothesis as the driver, cooperation is achieved by dialogue or negotiation.
- Abductive inference plays a central role in generating explanations. Although it is a weak mechanism for generating explanation and hypothesis, and although it cannot guarantee a practical solution, the inspiration provided by abductive inference has been shown to be very useful.

- Conflict resolution provides a mechanism to resolve the inconsistency between explanations generated and existing explanations.
  When an hypothesis is generated, the hypothesis generation model usually gives different explanations, The final hypothesis should be obtained by synthesis of the different explanations. This approach is a novel metaphor for the synthesis of solutions using a variety of source mappings contained in hypothesis space.
- Hypothesis space is a dynamical knowledge structure. It has the ability to organise itself, and maintain its complexity at a certain level according to the requirements of the application.

The experimental framework of this research has been validated in a complex commercial situation, namely fraud detection in the insurance industry.

10.2 Future work

Plans for future work have two aspects: application of the hypothesis generation methodology to commercial practice, and extension and improvement of the current hypothesis generation model.

For fraud applications, an on-line fraud detection system will be built based on hypothesis based problem solving methodology. At the same time, the template knowledge base for fraud detection will be extended to meet any new application requirements. Other features, such as on-line information retrieval and analysis, may be supported. The fraud detection system is capable of generalisation to intelligent decision support systems based on the global information resource.

An improved hypothesis generation model is currently being investigated for application in the domain of general insurance. Design studies of this model in collaboration with experts in the insurance industry are in progress.

At the theoretical level, further improvements should be anticipated:

- Although the anomaly detection model is quite sophisticated, its construction, especially the knowledge acquisition, is still quite time-consuming. An automated or semi-automated process is the goal, with operational records and knowledge acquired from their actual actions. Domain independent abstractions/rules are expected to be included in future anomaly detection models.
- Complexity control based on the semantic meaning of hypothesis space is required at a number of levels. At the same time, the hypothesis space should be capable of dynamic change for efficient operations.
- The effectiveness and efficiency of the hypothesis generation model is crucial. A more effective form of strategic knowledge for controlling abductive inference and conflict resolution is needed. The option to use other algorithms as a basic abduction operator is still open to investigation.
- In case based interaction, it is desired to change the case/hypothesis incrementally. In order to achieve this, it is necessary to maintain an hypothesis competition priority list in background. Dynamic connections within the hypothesis space should provide more flexibility.

The methodology developed in this research, for an hypothesis based problem solving approach, has wide application. We also believe that hypothesis generation offers intelligent decision support with the functionality and performance required. With further development, as outlined above, the hypothesis generation model extends the notion of machine intelligence.

Appendix 1. Network Resources Tools

- *WAIS* is the prototype that allows users to navigate information networks to locate resources. This tool is already in use around the world and provides access to greater diversity of services[78].
- *WHOIS* is another prominent example, used by Network Information Centers (NICs) and other organizations to maintain databases of registered users, and domains.
- *X.500*, a distributed directory service standard jointly developed by CCITT and ISO, describes a hierarchical name space, with provisions for caching, authentication, and replication. Users access this information through Directory User Agents. The most widespread use for X.500 currently is as a user directory and it can also store other types of information.
- *ARCHIE SERVICE* maintains a list of approximately 1,100 UNIX anonymous FTP archives world-wide, and builds a database of retrievable files by performing recursive directory listings at each site once per month. These sites contain about 150 gigabytes of information, in more ran 2.6 million files.
- *PROSPEO* file system allows users to organize files according to their personal preference while Archie allows users to search for files. In this sense, Prospeo is an "enabling technology" for building information infrastructure, and allows users to create their own views of the information in a distributed file system. Several global file systems, including the Andrew File System (AFS), and the Alex file system allow users to form local views of files by creating symbolic links from their own directories.

- ***KNOWBOT*** (Knowledge Robot), the notion introduced by the corporation of National Research Initiative, can launch searches for information in a network, possible replicating itself onto other nodes.
- ***WWW*** (World Wide Web) system allows user to organize and access information without concern for the distribution of the information.
- ***GOPHER*** system provides a simple menu-driven user interface that allows users to browse and locate information from a number of different sources throughout the world. Gopher provides a relatively uniform interface to this data, so that users need not understand many of the details of interacting with each of the systems being accessed.

Appendix 2. On-line Information Sources

- **Adtrack**

  a golden resource for the advertiser, ad agency, and media, publishing, and marketing professionals, Adtrack has gathered advertisements of a quarter page or larger from 150 major consumer and business publications.

- **D & B - Dun's Financial Records**

  In putting together its credit reports, Dun & Bradstreet collects a huge amount of information on American Companies. Through Dun's Financial Record, people can find up to three years of balance sheets and income statements, fourteen pre-calculated ratios, comparisons of the company's performance with the industry as a whole, company history, and operations summaries.

- **Harvard Business Review Database**

  Just a week after the Harvard Business Review is released in print, it appears on-line to join the more than 2,000 records already on file. The database for this noted bimonthly management periodical contains the full text of articles from 1976 to the present.

- **World Patents Index**

  This bible of international patents information comes from Derwent Publication Ltd. in London. Derwent suggests that only patents can give business professionals "Advance warning of technological innovation and development."

- **Dow Jones News/Retrieval**

  DJN/R divides its services into two major categories: Business & Investor Services (company/industry data and news; quotes and market averages; and on-line brokerage from Fidelity Brokerage Inc.) and General Services (World news, sports, weather; shopping, travel, MCI E-mail; and education/entertainment).

- **Text-Search Services**

  Text-Search Services includes over a thousand full text sources searchable in either a menu-driven or command version. These sources at current stage includes the following: *Dow Jones news, Wall Street Journal, Barron's Washington Post, Los Angeles Times,* and *Business Week.* Text search offers most of the advantages of full features online services, including field and proximity searching, sorting, limiting by various parameters, and custom output [158].

- **Telenet**

  Telenet claims to be the world's largest public data network. It is possible to get many commercial services thought *Telenet,* such as *The Sources* and *Dow Jones News/Retrieval.*

- **Dialog**

  Dialog is an information retrieval service with access to numerous large databases. It is reached by direct login over *Public Data Networks* such as *DIANET, TYMNET,* or *Telenet.*

- **Telebase**

  Telebase provides information retrieval services and specializes in sophisticated user interfaces to diverse databases.

Appendix 3. Information Source for Fraud Detection

- ASCOT

  Australian Security Commission database

- Electoral Roll

  Australian citizen registration data

- Electronic White Pages

  Name, address, telephone number (current information)

- Insurance Report Service

  Special claims

- Newsclassified

  For sale advertisements in nine newspapers of News Limited

- Land Title

  Property ownership, price

- OZ on Disk

  Name, address, phone Number.

- Cogen

  Insurance Policy and Claim database

- TPVR

  Weekly Trading Post Records

- Business Who's Who Australia

  Details about companies listed in Stock exchange market

References

[1] Awad, E. M. (1988) *Management Information Systems: Concepts, Structure, and Application*, Benjamin Cummings, Menlo Par, California, USA, 1988.

[2] Barker, S. F. (1969) *Induction and Hypothesis*, Cornell University Press, New York, USA, 1969.

[3] Barnett, V. and Lewis, T. (1978) *Outliers in Statistical Data*, John Wiley & Sons, New York, USA, 1978.

[4] Benbasat, I. and Wand, I. (1984) *A Structural Approach to Designing Human-Computer Dialogues*, International Journal of Man-Machine Studies, Volume 21, 1984.

[5] Binot, J. L., Peres, R., Sheehy, N. and Rouault, J. (1990) *Architecture of A Multi-model Dialogue Interface for Knowledge-based Systems*, Proceedings of the Annual ESPRIT Conference, Brussel, November, 1990.

[6] Bond, A. and Gasser, L. (1988) *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, California, USA, 1988.

[7] Boush, G. B. (1991) *A Program for Sharing Corporate Intelligence*, The Journal of Business Strategy, January/February 1991.

[8] Bower, G. H.(1975) *Cognitive Psychology: an Interpretation*, In W. K. Estes,(Ed). Handbook of Learning and Cognitive Processes, Erlbaum, Hillside, New Jersey, USA, 1975.

[9] Bowers, J. M. and Benford, S. D. (Ed) (1991) *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, North-Holland, 1991.

[10] Brachman, R. J. (1979) *On the Epistemological status of Semantic Networks*, In N. V. Findler (Ed), Associated Networks: Representation and Use Knowledge by Computers, Academic Press, New York, NY, USA, 1979.

[11] Brown, C., Pattabhiraman, T., Boyer, M., Masssam, D. and Dahl, V. (1986) *Tailoring Conceptual Graphs for Use in Natural Language Translation*, Proceedings of the IBM Workshop on Conceptual Graphs, Tornwood, USA, 1986.

[12] Brown, J. S., Burton, R. R. and De Kleer, J. (1982) *Pedagogical, Natural Language, and Knowledge Engineering Techniques in SOPHIE I, II, and III*, In: D. Steeman and J. S. Brown (Eds.), Intelligent Tutoring Systems, Academic Press, London, UK, 1982.

[13] Bruner, J. S., Goodnow, J. J. and Austin, G. A. (1962) *A Study of Thinking*, R. E. Krieger Pub. Co. Huntington, New York, USA, 1962.

[14] Buchanan, B. G. and Shortliffe, E. H. (1984) *Rule-based Expert Systems*, Addison-Wesley, Reading, Mass., USA, 1984.

[15] Buckland, J. A. and O'Brien, R. P. (1989) *Strategic Information System*, In Critical Issues in Information Processing Management and Technology, Vol 1, QED information Science, Inc., Wellesley, Mass., USA, 1989.

[16] Buckner, G. D. and Shah, V. (1993) *Future Vision: Impacts of artificial Intelligence on Organizational Success*, Kybernetes, Volume 22, No. 8, 1993, pp.40-50.

[17] Burton, R. R. and Brown, J. S. (1979) *Toward a Natural Language Capability for Computer-Assisted Instruction*, In: O'Neil, H. ed. Procedure for Instructional System Development, Academic Press, New York, NY, USA, 1979.

[18] Chan, M. C., Garner, B. J. and Tsui, E. (1988) *Recursive Modal Unification for Reasoning with Knowledge Using a Graph Representation*, Knowledge-based Systems, Volume 1, No. 2, 1988, pp. 94-104.

[19] Charniak, E and McDermott, D. (1985) *An Introduction to Artificial Intelligence*, Addison-Wesley Publishing Company, Reading, Mass., USA, 1985.

[20] Charniak, E and Shimony, S. (1990) *Probabilistic Semantics for Cost Based Abduction*, Proceedings of the AAAI Conference, 1990.

[21] Chen, J. and Srihari, S. (1993) *Hypothesis Generation in Conflict-based Diagnosis,* International Journal of Pattern Recognition and Artificial Intelligence, Volume 7, No. 3, 1993, pp. 475-492.

[22] Cheriton, D. R. (1976) *Man-machine Interface Design for Timesharing Systems*, Proceedings of the ACM Annual Conference. ACM, New York, USA,1976, pp.362-380.

[23] Clancey, W. J. (1985) *Book Review of Sowa*, Artificial Intelligence, Volume 27, 1985, pp. 113-124.

[24] Clancey, W. J. (1985) *Heuristic Classification*, Artificial Intelligence, Volume 27, 1985, pp. 289-350.

[25] Clancey, W. J. (1983) *The Epistemology of a Rule-based Expert System: A Framework for Explanation*, Artificial Intelligence, 20 (3), 1983, pp. 215-251.

[26] Cox, P. T., Pietrzykowski, T. (1986) *Causes for Events: Their Computation and Applications*, Proceedings of 8th International Conference on Automated Deduction, CADE'86, 1986. pp. 608-621.

[27] Cullingford, R. (1981) *SAM*, In R. Schank and C. Reisbeck (Ed) Inside Computer Understanding, Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, 1981.

[28] Debenham, J. (1993) *Objects for Knowledge Modelling*, In J. Liebowitz (Ed) Moving Towards Expert System Globally in the 21st Century, Proceedings of The Second World Congress on Expert Systems, Lisbon, Portugal, 1993.

[29] Demolombe, R. and del Cerro, L. F.(1991) *An Inference Rule for Hypothesis Generation*, Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, 1991, pp. 152-157.

[30] Dempster, A. P. (1968) *A Generalization of Bayesian Inference*, Journal of the Royal Statistical Society, Series. B, Vol.30, 1968, pp. 204-247.

[31] Doyle, A. C. (1953) *The Complete Sherlock Holmes*, Doubleday & Company, Inc. New York, USA, 1953.

[32] Dunbar, K. and Klahr, D. (1989) *Developmental Differences in Scientific Discovery Processes*, In D. Klahr and K. Kotovsky (Ed) Complex Information Processing (21st Carnegie-Mellon Symposium on Cognition), Lawrence Erlbaum Association, Hillsdale, New Jersey, USA, pp. 109-143.

[33] Dung, P. M. (1991) *Negotions As Hypotheses: An Abductive Foundation for Logic Programming*, Proceedings of 8th International Conference on Logic Program, 1991, pp. 3-17.

[34] Edmonds, E. A., Hague, S. J., McDaid, E.and Reid, I. (1990) *The FOCUS Project: A Separable Architecture for KBFEs*, Proceedings of ESPRIT II Conference: Information Processing Systems, November, 1990.

[35] Edmonds, E. A., Garner, B. J., and Murray, B. S. (1991) *Proposal for Research into Case-based Interaction for User Interfaces*, University of Technology, Loughborough, UK, 1991.

[36] Ellman, T.(1989) *Explanation-Based Learning: A Survey of Programs and Perspectives*, ACM Computing Survey, Volume 21, No. 2 1989, pp. 163-222.

[37] Ellsworth, J. (1994) *The Internet Business Book*, Wiley, New York, USA, 1994.

[38] Eshghi, K.(1988) *Abductive Planning with Event Calculus*, Proceedings 5th International Conference and Symposium on Logic Programming, Washington, Seattle, USA, pp. 562-579.

[39] Fann, K. T. (1970) *Peirce's Theory of Abduction*, Martinus Nijhoff, The Hague, Holland, 1970.

[40] Fayyard, U (Ed.) (1995) Proceedings of the International Conference on Knowledge Discovery and Data Mining, Mentral, Canada, August, 1995.

[41] Fayyad, U., Piatetsky-Shapiro, G.and Smyth (1995) *From Data Mining to Knowledge Discovery: An Overview*, In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P and Uthurusamy, R. (Ed.) *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Boston, USA, 1995.

[42] Fayyad, U., Piatetsky-Shapiro, and Matheus, C. (1992) *Knowledge Discovery in Database*, AI Magazine, Fall, 1992.

[43] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P and Uthurusamy, R. (Ed.)(1995) *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Boston, USA, 1995.

[44] Fischer, G., Lemke, A. C., Mastaglio, T., and Morch, A. I. (1991) *The Role of Critiquing in Cooperative Problem Solving*, ACM Transaction on Information Systems, Volume 9, No.3, 1991, pp. 123-151.

[45] Foley, J. Gibbs, C. Kim, W. C. and Kovacevic, S. (1988) *A Knowledge-Based User Interface Management System*, Proceedings of ACM CHI'88 Conference on Human Factor In Computing Systems, 1988, pp.67-72.

[46] Forrester,. C. (1993) *Decision Support under Unstructured Environment*, Ph. D. Thesis, School of Computing and Mathematics, Deakin University, Victoria, Australia, 1993.

[47] Foo, N., Garner, B., Rao, A. and Tsui, E. (1992) *Semantic Distance in Conceptual Graphs*, in T. Nagle, J. Nagle and L. Gerholz Edit: Conceptual Structures, Current Research and Practice, Ellis Horwood, New York, USA, 1992.

[48] Fuld, L. (1991) *A Recipe for Business Intelligence*, The Journal of Business Strategy, January/February, 1991.

[49] Garner, B. and Chen, F. (1992) *Case-Based Interaction for Fraud Detection in EDP*, International Conference on Intelligent Information Systems and Processing, Beijing, China, 1992.

[50] Garner, B. and Chen, F. (1994) *Anomaly Detection Modelling*, Proceedings of The International Conference on Industrial and Engineering Application of AI and Expert Systems, Austin, Texas, USA, 1994.

[51] Garner, B. and Chen, F. (1994) *Hypothesis Generation Paradigm for Fraud Detection*, Proceedings of IEEE Region 10's Ninth Annual International Conference on: "Frontiers and Computer Technology", Singapore, 1994.

[52] Garner, B. and Chen, F. (1994) *The Rules of Hypotheses in Human Computer Interaction*, Proceedings of OZCHI Annual Conference on Human Computer Interaction, Melbourne, Australia, 1994.

[53] Garner, B. and Chen, F. (1995) *Role of Virtual Relation in Decision Support*, Proceedings of the Second International Symposium on Knowledge Acquisition, Representation and Processing, Auburn, USA, September, 1995, pp. 27-30.

[54] Garner, B. and Lukose, D. (1989) *Active Agent Paradigm for Knowledge Based Systems*. In S. Ramani, R. Chandrasekaran and K. S. R. Anjaneyulu (Ed) Lecture Notes in Artificial Intelligence 444, Springer-Verlag, New York, 1989.

[55] Garner, B. and Lukose, D. (1991) *Actor Graphs: A Novel Executable Conceptual Structure*, Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Objects and Artificial Intelligence, Sydney, Australia, 1991.

[56] Garner, B. and Lukose, D.(1992) *Knowledge Fusion*, Proceedings of the Seventh Annual Workshop on Conceptual Graphs, Las Cruces, July, 1992.

[57] Garner, B., Lukose, D. and Tsui, E. (1987) *Parsing Natural Language through Pattern Correlation and Modification*, Proceedings of the 7th International Workshop on Expert Systems and Applications, Avignon, France, 1987, pp.1285-1299.

[58] Garner, B. and Tsui, E. (1986) *Control Scripts for the Reorganization of Office Knowledge Bases*, Proceedings of the First Australian AI Congress, November, Melbourne, Australia, 1986.

[59] Gevarter, W. E. (1984) *Artificial Intelligence, Expert Systems, Computer Vision, and Natural Language Processing*, Noyes Publications, Park Ridge, New Jersey, UAS, 1984.

[60] Gholson, B.(1980) *The Cognitive-Developmental Basis of Human Learning*, Academic Press, New York, USA, 1980.

[61] Ghoshal, S. and Kim S. K. (1986) *Building Effective Intelligence Systems for Competitive Advantage*, Sloan Management Review, Fall, 1986, pp. 49-58.

[62] Gmytrasiewicz, P. J., Durfee, E. H. and Wehe, D. K. (1991) *A Decision-Theoretic Approach to Coordinating Multiagent Interaction*, Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, 1991, pp. 82-68.

[63] Goldschmidt, P (1994) *ALCOD: An IDSS for Stock Market Surveilance, Technical report*, Depatment of Accounting and Financial, The University of Western Australia, 1994.

[64] Greif, I. (1988) *Computer-Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann, San Mateo, California, U. S. A. 1988.

[65] Gruber, T. R. (1989) *Automated Knowledge Acquisition for Strategic Knowledge*, Machine Learning 4, Kluwer Academic Publishers, Boston, USA 1989, pp. 293-336

[66] Hammond, K. J. (1989) *Adaption of Cases*, Proceedings of a Workshop on Case-Based Reasoning, Pensacola Beach, Florida, USA, 1989.

[67] Hartley, R. H. (1985) *Representation of Procedural Knowledge for Expert System*, Proceedings of the IEEE Conference on Artificial Intelligence Applications, December 11-13, Miami Beach, Florida, USA, 1985.

[68] Hartley, R. and Coombs, M. (1991) *Reasoning with Graph Operations*, In J. Sowa (Ed) Principles of Semantic Networks, Morgan Kaufmann Publisher, San Mateo, California, USA, 1991.

[69] Hartson, H. R. and Hix, D. (1989) *Human-Computer Interface Development: Concepts and Systems for its Management*, ACM Computing Surveys, Vol21, No. 1, 1989, pp. 5-92.

[70] Harvard Business Review (1987) *Do You Think There is a Competitiveness Problem?* May/June, 1987, pp. 1-27

[71] Hewitt, C. (1969) *PLANNER: A Language for Proving Theorems in Robots*, Proceedings of the International Joint Conference on Artificial Intelligence, Washington D. C., USA, 1969.

[72] Hewitt, C., Bishop, P. and Steiger, R. (1973) *A Universal Modular ACTOR Formalism for Artificial Intelligence*, Proceedings of the International Joint Conference on Artificial Intelligence, Stanford, USA, 1973.

[73] Hine, J. and Montgomery, A. (1993) *The Development of Research and Education Networks in Southeast Asia*, Proceedings of the Ninth International Congress on Cybernetics and Systems, New Delhi, India, 1993.

[74] Hobbs, J. R., Stickel, M. E., Appelt D. E. and Martin, P. (1993) *Interpretation as Abduction*, Artificial Intelligence, Volume. 63, 1993, pp. 69-142.

[75] Hoppe, H. U. (1988) *Task-oriented Parsing - A Diagnostic Method to be Used by Adaptive Systems*, Proceedings of the ACM HCI,88 Conference on Human Factors in Computing Systems, 1988, pp. 241- 247.

[76] Inoue, K. (1988) *Problem Solving with Hypothetical Reasoning*, Proceeding of the International Conference on Fifth Generation Computer Systems, 1988, pp.1275-1281

[77] Ji, G. (1993) *Representing Inference Control by Hypothesis-Based Association*, IEEE transactions on Knowledge and Data Engineering, Volume 5, No. 2, 1993, pp. 363-367.

[78] Kahle, B. and Medlar, A. (1991) *An Information System for Corporate Users: Wide Area Information Services*, Online, September, 1991, pp. 56-60.

[79] Kass, A. and Leake, D. (1988) *Case-based Reasoning Applied to Constructing Explanations*, In Kolodner, J. (Ed.), Proceedings of the Case-based Reasoning Workshop, Morgan Kaufmann, Palo Alto, USA, 1988.

[80] Kochen, M. (1989) *Management Intelligence Systems*, in M. Yovits (Ed) Advances in Computers, Volume 28, Academic Press, Boston, USA, 1989, pp. 227-279.

[81] Kochen, M. and Min, M. (1987) *Intelligence for Strategic Planning*, Proceedings of the 7th International Workshop on Expert Systems and Their Applications, 1987.

[82] Konolige, K. (1990) *A General Theory of Abduction*, Proceedings of the AAAI Spring Symposium on Automated Abduction, Stanford, CA, USA, 1990.

[83] Konolige, K. (1992) *Research Note: Abduction versus closure in Causal Theories*, Artificial Intelligence, Volume 53, 1992, pp. 255-272.

[84] Keppler, K. (1993) *Falcon Fights Credit Fraud*, AI Expert, March, 1993, p56.

[85] Kneale, W. (1949) *Probability and Induction*, Oxford University Press, UK, 1949.

[86] Kraus, S. and Wilkenfeld, J. (1991) *Negotiation Over Time in a Multi-Agent Environment*: Preliminary Report, IJCAI-91, 1991, pp. 56-61.

[87] Krems, J. and Bachmeier, M. (1993) *Hypothesis Generation And Strategy-Selection Dependent on the Level of Expertise*, German Journal of Psychology Volume 17, No. 1, 1993, pp. 32-34.

[88] Leake, D. (1991) *Goal-based Explanation Evaluation*, Cognitive Science, Volume 15, No. 4, 1991.

[89] Lehnert, W. (1977) *A Conceptual Theory of Question Answering*, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 1977.

[90] Levesque, H. J. (1989) *A Knowledge-level Account of Abduction*, Proceedings of the 11th International Joint Conference on Artificial Intelligence, Detroit, MI, USA, pp.1061-1067

[91] Leonard, K. (1993) *Detecting Credit Card Fraud Using Expert Systems*, Computers and Industrial Engineering, Volume25, September, 1993, pp. 103-106.

[92] Levinson, R.A. (1985) *A Self Organizing Retrieval Systems for Graphs*, Ph. D. Thesis, Artificial Intelligence Laboratory, The University of Texas, Austin, Texas, UAS, 1985.

[93] Lin, F. and Lin, M. (1993) *Analysis of Financial Data Using Neural Nets*, AI Expert, February, 1993, pp. 37-41.

[94] Lui, D. (1990) *Strategic Knowledge for the Identification and Construction of Goal States*, M. Sc. Thesis, Department of Computing and Mathematics, Deakin University, Geelong, Victoria, Australia, 1990.

[95] Lukose, D. (1992) *Goal Interpretation as a Mechanism for Knowledge Acquisition*, Ph.D. Thesis, Department of Computing and Mathematics, Deakin University, Australis, 1992.

[96] Manheimer J. M., Burnett, R. C. and Wallers, J. A. (1989) *A case study of user interface management system development and application*, Proceedings of ACM CHI'89 conference on human factor in computing systems, 1989, pp.127-132.

[97] Major, J. and Riedinger, D. (1992) *EFD: A Hybrid Knowledge/Statistical Based System for the Detection of Fraud*, International Journal of Intelligent Systems, Volume7, 1992, pp. 687-703.

[98] McCalla, G. and Cercone, N. (1983) *Guest Editors' Introduction: Approaches to Knowledge Representation*, IEEE Computer, Volume 16, No. 10, 1983.

[99] McCreath, E. and Sharma, A. (1995) *Extraction of Meta-Knowledge to Restrict the Hypothesis Space for ILP System,* Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence, Canberra, Australia, 1995, pp. 75-82.

[100] Miller, L. A. and Thomas, J. C. (1977) *Behavioral issues in the use of interactive system*, International Journal of Man-Machine Studies, Volume 9, 1977, pp. 509-536.

[101] Mineau, G. (1990) *Induction on Conceptual Graphs: Finding Common Generalizations and Compatible Projections*, Forth Annual Workshop on Conceptual Graphs, IJCAI-89, Detroit, Michigan, USA, 1990.

[102] Minsky, M. (1975) *A Framework for Representing knowledge*, In Patric Henry Winston (Ed.) The Psychology of Computer Vision, McGraw-Hill, New York, NY, USA, 1975.

[103] Mockler, R. J. (1989) *Knowledge-based Systems for Management Decisions*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[104] Morgan, C. (1971) *Hypothesis Generation by Machine*, Artificial Intelligence Volume 2, 1971, pp. 179-187.

[105] Myaeng, S. H. and Lopez-Lopez, A. (1991) *A Flexible Algorithm for Matching Conceptual Graphs*, Proceedings of the Sixth Annual Workshop on Conceptual Graphs, July, 1991, pp.135-151.

[106] Nagle, T. E. (1986) *Conceptual Graphs in an Active Agent Paradigm*, Proceedings of the IBM Workshop on Conceptual Graphs, Thornwood, USA, 1986.

[107] Nagle, T. E., Esch, J. W. and Minerau, G. (1990) *A Notation for Conceptual Structure Graph Matchers*, Proceedings of the Fifth Annual Workshop on Conceptual Structures, AAAI-90, Boston, Mass., USA, 1990, pp.121-129.

[108] Namkiu, C. and Anzai, Y (1995) *Integrating Hypothesis-Based Exploratory Strategy and Reinforcement Learning*, Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence, Canberra, Australia, 1995, pp. 483-490

[109] Newell, A. (1982) *Knowledge Level*, Artificial Intelligence, Volume 18, 1982, pp. 87-127.

[110] Newell, A. (1962) *Some Problems of the Basic Organization in Problem-Solving Programs*, Proceedings of the Second Conference on Self-organizing Systems, Spartan Books, Washington DC, USA, pp. 393-423.

[111] Nierstrasz, O. (1989) *A Survey of Object-oriented concepts*, In Kim, W. and Lochovsky, F. (Eds) Object-Oriented Concepts, Databases and Applications, ACM Press and Addison-Wesley, New York, NY, USA, 1989, pp. 3-21.

[112] Nonogaki, H. and Ueda, H. (1991) *Friend21 project: A construction of 21st century human interface*, Proceedings of ACM CHI'91 conference on human factor in computing systems, 1991, pp. 407-414.

[113] Olsen, J., Whitten, W. and Gruenenfelder, T. (1984) *A General User Interface for Creating and Displaying Tree Structures, Hierarchies, Decision Trees, and Nested Menus*, in Y. Vassiliou (Ed.) Human Factors and Interactive Computer Systems, Albex, Norwood, New Jersey, USA, 1984.

[114] Owens, C. (1989) *Plan Transformations as Abstract indices*, Proceedings of a workshop on Case-based Reasoning, Pensacola beach, Florida,USA, 1989.

[115] Pagnucoo, M. (1994) *A Simple Approach to Abductive Inference using Conceptual Graphs*, Proceedings of 1st Australian Conceptual Structures Workshop, Armidale, New South Wales, Australia, 1994.

[116] Patel, J. and Butcher, M. (1990) *A Case Study in Intelligent Front End Design*, Research Report 41, School of Computing and Information Technology, Griffith University, Australia, 1990.

[117] Paul, G. (1993) *Approaches to Abductive Reasoning: an Overview*, Artificial Intelligence Review 7, 1993, pp.109-152.

[118] Peirce, C. S. (1974) *Collected Papers of Charles Sanders Peirce* (Edited by C. Hartshorne and P. Weiss) Harvard University Press, Cambridge, Mass. USA, 1974.

[119] Peng, Y. and Reggia, J. A. (1986) *Plausibility of Diagnostic Hypotheses: The Nature of Simplicity*, Proceedings of AAAI86 National Conference on Artificial Intelligence, AAAI Press, Menlo Park, USA, 1986, pp. 140-149.

[120] Peng, Y. and Reggia, J. A. (1987) *A Probabilistic Causal Model for Diagnostic Problem Solving*, IEEE Transactions on Systems, Man and Cybernetics, Volume SMC-17, 1987.

[121] Peterson, J. L. (1981) *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, Englewood Cliffs, New Jersey. USA, 1981.

[122] Philip, G. (1990) *Smalltalk-80: a Practical Instruction*, Pitman, London, UK, 1990.

[123] Popper, K. R. (1959) *The Logic of Scientific Discovery,* Basic Books, New York, USA, 1959.

[124] Poole, D. (1988) *A Logical Framework for Default Reasoning*, Artificial Intelligence, Vol. 36, 1988, pp. 27-47.

[125] Poole, D. (1989) *Explanation and Prediction: an Architecture for Default and Abductive Reasoning*, Computational Intelligence Volume 5, 1989, pp. 97-110.

[126] Pople, H. E. (1973) *On the Mechanization of Abductive Logic*, Proceedings of the Third International Joint Conference on Artificial Intelligence, 1973.

[127] Porter, B. W., Bareiss, R. and Holte, B. C. (1990) *Concept Learning ana Heuristic Classification in Weak-Theory Domains*, Artificial Intelligence, Volume 45, 1990, pp. 229-263.

[128] Pryor, L. and Collins, G. (1992) *Reference Features as Guides to Reasoning about Opportunities*, Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society, 1992, pp. 230-235.

[129] Ram, A. and Leake, D. (1991) *Evaluation of Explanatory Hypotheses*, Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, Chicago, Illinois, USA, 1991.

[130] Ram, A. and Leake, D. (1994) *Learning, Goals, and Learning Goals*, in A. Ram and D. Leake (ed.) Goal-Driven Learning, MIT Press, Cambridge, Mass., USA, 1994.

[131] Ram, A. and Hunter, L. (1992) *The Use of Explicit Goals for Knowledge to Guide Inference and Learning*, Applied Intelligence, Volume 2, No. 1, 1992, pp. 47-73.

[132] Reggia, J., Nau, D. and Wang, P. (1983) *DiagnostiC Expert Systems Based on a Set Covering Model*, International Journal of Man-Machine Studies, Volume 19, 1983, pp. 437-460.

[133] Remenyi, D. (1991) *Introducing Strategic Information Systems Planning*, NCC Blackwell Limited, Oxford, UK, 1991.

[134] Rich, E. (1979) *User Modeling via Stereotypes*, Cognitive Science, Volume 3, 1979, pp. 355-366.

[135] Rich, E. (1982) *Artificial Intelligence*, McGraw-Hill, New York, USA, 1982.

[136] Richard, J. (1993) *Navigating the Internet*, Sams, Indianapolis, Ind. USA, 1993.

[137] Riesbeck, C. and Schank, R. (1989) *Inside Case-based Reasoning,* Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, 1989.

[138] Rissland, E.L (1984) *Interdients of Intelligent User Interface*, International Journal of Man-Machine Studies, Volume 21, No. 3,1984.

[139] Sammon, W. L. (1984) *Competitor Intelligence: An Organizational Framework for Business*, In W. L.Sammon, M. A. Kurland and R. Spitalnic (Eds) Business Competitor Intelligence, A Ronald Press Publication, USA, 1984.

[140] Sammon, W. L. (1984) *Competitor Intelligence: An Analytical Approach*, In W. L.Sammon, M. A. Kurland and R. Spitalnic (Eds) Business Competitor Intelligence, A Ronald Press Publication, USA, 1984.

[141] Schank, R. (1972) *Conceptual Dependency: A Theory of Natural Language Understanding*, Cognitive Psychology, Volume3, 1972, pp. 551-631.

[142] Schank, R. (1980) *Language and Memory*, Cognitive Science, Volume 4, No. 3, 1980, pp. 243-284.

[143] Schank, R. (1986) *Explanation Patterns: Understanding Mechanically and Creatively*, Lawerance Erlbaum Associates, Hillsdale, New Jersey, USA, 1986.

[144] Schank, R. and Abelson, R. (1977) *Scripts, Plans, Goals and Understanding*, Lawerance Erlbaum Association, New Jersey, USA, 1977.

[145] Schank, R., and Leake, D. B. (1989) *Creativity and Learning in a Case-based Explainer*, Artificial Intelligence, Volume 40, 1989, pp. 353-385.

[146] Scott, A., Clancey, W., Davis, R. and Shortliffe, E. (1984) *Methods for Generating Explanations*, In B. Buchanan and E. Shortliffe (Editors) Rule-based Expert Systems, Addison-Wesley, Reading, Mass., USA, 1984.

[147] Shafer, G. (1976) *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, New Jersey, USA, 1976.

[148] Simon, H. A. (1977) *Models of Discovery*, D. Reidel Publishing Company, Dordrechi, Holland, 1977.

[149] Simon, H. A. (1973) *The Structure of Ill structured Problems*, Artificial Intelligence, Volume 4, 1973, pp. 181-201.

[150] Simon, H. A. and Lea, G. (1974) *Problem Solving and Rule Induction: A Unified View*, In L. W. Gregg (Ed.) Knowledge and Cognition. Lawrence Erlbaum Associates, Hillsdale, New Jersey, USA, 1974.

[151] Sowa, J. (1984) *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, Mass., USA, 1984.

[152] Sowa, J. and Way, E. (1986) *Implementing a Semantic Interpreter Using Conceptual Graphs*, IBM Journal of Research and Development, Volume 30, No. 1, January, 1986, pp. 57-69.

[153] Sowa, J. (1995) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, PWS Publishing Company, Boston, Mass., USA, 1995.

[154] Swartout, W. and Moore, J. (1993) *Explanation in Second Generation Expert Systems*, In J. Davis, J. Krivine and R. Simmons (Editors) Second Generation Expert Systems, Springer-Verlag, New York, NY, USA, 1993.

[155] Thierauf, R. J. (1991) *Executive Information Systems: A Guide for Senior Management and MIS Professionals*, Quorum Books, New York, USA, 1991.

[156] Tsui, E. (1988) *Canonical graph model*, Ph.D. Thesis, School of Computing and Mathematics, Deakin University, Victoria, Australia, 1988.

[157] Tyler, S. W. (1988) *SAUCI: A Knowledge-Based Interface Architecture*, Proceedings of ACM CHI'88 conference on human factor in computing systems, Associations for Computing Machiney, New York, USA, 1988, pp. 233-240.

[158] Wagers, R. (1992) *DowQuest and Dow Jones Text-search: Which Works Best and When?* Online, November, 1992, pp. 35-41.

[159] Winograd, T. (1973) *A Procedural Model of Language Understanding, Computer Models of Thought and Language*, In R. Schank and K. Colby (Ed) W. H. Freeman, New York, USA, 1973.

[160] Winograd, T. (1975) *Frames and the Procedural Declarative Controversy*, In Bobrow, D. and Collins, A. (Eds) Representation and Understanding, Academic Press, NewYork, NY, USA, 1975.

[161] Wiseman, C. (1988) *Strategic Information Systems*, Irwin, Homewood, Illinois, USA, 1988.

[162] Wold, G. and Shriver, R. (1989) *Computer Crime Techniques Prevention*, Bankers Publishing Company, Rolling Meadows, Illinois, USA, 1989

[163] Woods, W. A. (1978) *Semantics and Quantification in Natural Language Answering*, In M. Yovits (Ed) Advances in Computers, Volume 17, Academic Press, New York, NY, USA, 1878.

[164] Woods, W. A. (1983) *What's Important about Knowledge Representation*, In G. McCalla and N. Cercone (Eds.) IEEE Computer Special Issues on Knowledge Representation, IEEE, New York, NY, USA, 1983.

[165] Wuwongse, V. and Niyomthai, S. (1991) *Conceptual graphs as a Framework for Case-based Reasoning*, Proceedings of the Sixth Annual Workshop on Conceptual Graphs, July, 1991, pp.119-133.

[166] Zadeh, L. A. (1978) *Fuzzy Sets as a Basis for a Theory of Possibility*, Fuzzy Sets Systems, Volume 1, 1978, pp. 3-28.

[167] Zhang, M. and Zhang, C. (1993) *Conflict Resolution in Distributed Artificial Intelligence*, Technical Report 93-65, Department of Mathematics, Statistics and Computing Science, The University of New England, New South Wales, Australia, 1993.

[168] Zhang, C. (1991) *Cooperation under Uncertainty in Distributed Expert Systems*, TR-91-38, Department of Mathematics, Statistics and Computing Science, University of New England, New South Wales, Australia, 1991.