Volume 21 Number 2 June 1997 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:
Human-Like Systems Using AI Techniques

Guest Editor:
Se Woo Cheon

The Slovene Society Informatika, Ljubljana, Slovenia

Informatica **21** (1997) Number 2, pp. 157–331

Informatica

An International Journal of Computing and Informatics

Basic info about Informatica and back issues may be FTP'ed from `ftp.arnes.si` in magazines/informatica ID: anonymous PASSWORD: <*your mail address*>
FTP archive may be also accessed with WWW (worldwide web) clients with
URL: `http://www2.ijs.si/~mezi/informatica.html`

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 1997 (Volume 21) is
– DEM 50 (US$ 35) for institutions,
– DEM 25 (US$ 17) for individuals, and
– DEM 10 (US$ 7) for students
plus the mail charge DEM 10 (US$ 7).
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, DALCOM d.o.o., Stegne 27, 1000 Ljubljana, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 61 1773 900, Fax (+386) 61 219 385, or use the bank account number 900–27620–5159/4 Ljubljanska banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

According to the opinion of the Ministry for Informing (number 23/216-92 of March 27, 1992), the scientific journal Informatica is a product of informative matter (point 13 of the tariff number 3), for which the tax of traffic amounts to 5%.

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences (Janez Peklenik)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Cur. Cont. & Comp. & Math. Sear., Engineering Index, INSPEC, Mathematical Reviews, Sociological Abstracts, Uncover, Zentralblatt für Mathematik, Linguistics and Language Behaviour Abstracts, Cybernetica Newsletter

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102. Slovenia taxe Percue.

Guest Editorial: Toward More Human-like Systems Using the Techniques of Artificial Intelligence

In the artificial intelligence (AI) society, many techniques have been researched and developed since the year 1956, starting from general problem-solving techniques. These techniques include AI languages and tools, reasoning, pattern recognition, expert systems, neural networks, and several others.

Concerning the research trend in AI, studies on the "symbolic" AI looked somewhat depressed in the early 1990s due to the re-invitation of neural networks and fuzzy sets. Nowadays, as the computing powers are increasing (i.e., many limitations have been resolved) and the "Information Superhighway" such as the Internet is being the most popular, new research topics are being heavily studied by many researchers. These include intelligent navigation on the Internet, multimedia/distributed expert systems, case-based reasoning and data mining.

Total 11 papers out of the 107 have been selected from the most recent proceedings of "The Joint 1997 Pacific Asian Conference on Expert Systems/Singapore International Conference on Intelligent Systems (PACES/SPICIS '97)" (Chairmen: Dr. Graham Leedham & Dr. Dan Patterson) held on 24–27 February 1997 in Singapore.

The selection process was difficult because there were so many good papers on various applications and techniques of AI. All of the selected papers have been extensively revised and expanded into full papers for the journal. We tried as much as possible to include papers from different countries, providing a coverage of different topics.

The categories of the papers may be classified as follows:

1. intelligent multi-agent structure (2 papers).
2. intelligent information navigation on the Internet (2 papers).
3. diagnosis including case-based reasoning (3 papers).
4. machine learning including data mining (2 papers).
5. reasoning including fuzzy logic (2 papers).

The first two papers address the topic of the intelligent multi-agent structure.

An agent is viewed as an entity functioning continuously and autonomously. An agent can be viewed as consisting of mental components such as knowledge, beliefs, abilities and commitments.

The first paper, "A Safe and Efficient Agent Architecture" (Yuan) presents an agent architecture which is able to enforce the safety of an agent (i.e., generate acceptable actions without harmful actions to human) and work efficiently with incomplete information. The basic idea is to combine the culture support in the domain ontology and the decomposition and causality of partial-order planning.

The main issue in multi-agent systems is how to get a group of agents to carry out coordinated and coherent activities in a dynamic environment. Therefore, the other paper, "Multi-Agent Systems as a Paradigm for Intelligent System Design" (El Fallah Seghrouchni) proposes an efficient approach to build intelligent multi-agent systems. As agents are willing to cooperate, they exchange their respective plans and coordinate them to cancel negative interactions and to take advantage of helpful ones. In the paper, the coordination mechanism for agents relies on distributed planning, and concurrent plans are specified through the Recursive Petri Net formalism.

The two papers in the next category deal with intelligent information navigation on the Internet. The Internet is basically a huge interactive network connecting millions of computers all over the world, which began as a military project to allow computers to talk to or link to each other. This has since transformed the communications world.

The paper "Internet Information Brokering: A Re-Configurable Database Navigation, Data Filter and Export System" (Abidi) describes an Intenet-based database navigation system, called RIIB. The system can dynamically connect and interact in real-time with any remote database via the Internet. The system is based on the notion of database virtual hierarchies, i.e., a database navigation engine allows users to navigate the

database by envisaging it as a user-defined hierarchical structure.

The second paper, "MANIS: A Multi-Agent System for Network Information Services" (Kang & Shi) presents a multi-agent system, called MANIS, which integrates several methods for text understanding and offers the text understanding services over the Internet. MANIS is organized as a market-like system, in which a broker agent can choose among services based on the sharing information in the service market, and a server agent can reorganize via decomposition and composition. The paper also presents a social agent framework for self-organization.

The next three papers show that the techniques of expert systems can be successfully applied to the diagnosis. In expert systems today, diagnosis is one of the largest application domains.

The first paper, "Cognitive Simulation of Operator's Diagnostic Strategies in Nuclear Power Plants" (Cheon et al.) describes an approach to simulate operator's diagnostic strategies under emergency situation of nuclear power plants. They develop a diagnostic model (SACOM) using a blackboard architecture. Diagnostic agents in the model can simulate four types of operator's diagnostic strategies: i.e., data-driven search, symptomatic search, hypothesis-driven search, and topographic search.

The next paper, "Modelling Human Cognition in Problem Diagnosis: A Hybrid Case-Based Reasoning Methodology" (Law) shows how to implement a hybrid expert system (IHDF) for a business domain, centered around a computer hotline help desk service.

The system is based on both rule-based and case-based reasoning frameworks. Case-based reasoning (CBR) is a method of solving a current problem by analyzing the solutions to previous, similar problems. In the system, the techniques of case-based reasoning include case memory organization networks, case indexing and retrieval schemes, and an interactive and incremental style of reasoning.

The final paper in this section, "Medical Decision Support System for the Management of Hypertension" (Chae et al.) descibes the development of a medical decision support system (MDSS) for the management of essential hypertension.

Logistic regression is used to identify risk factors for essential hypertension, and three knowledge models, i.e., a neural network, case-based reasoning, and a statistical discriminant analysis method, are compared to determine the best method for predicting the severity of hypertension and the prognosis of treatment.

The next two papers describe machine learning including data mining.

The first paper, "Lattice-based Knowledge Discovery in Network Management Data" (Venter et al.) discusses the problem of knowledge discovery in databases and data mining (KDD). They employ a concept lattice to determine dependencies between elements of a computer network. The data mining refers to the process of abstracting various kinds of useful information from data. The paper demonstrates an experiment to investigate the feasibility of a KDD technique called "Lattice based knowledge discovery."

The "Organizational Science Approach to Knowledge Intensive Learning and Adaptation in a Multiagent System" paper (Hatakama & Terano) proposes a knowledge intensive learning model in a multi-agent system. They examine the validity and feasibility of the model to apply it to distributed heterogeneous knowledge systems. The paper describes simulation studies on the dynamic behaviors of decision making and learning in the organization of agents.

The final two papers address the techniques concerning non-monotonic reasoning and fuzzy logic. The paper "A Case Study in Non-monotonic Reasoning: an Alternative Analysis of the Yale Shooting Problem" (Sun) shows a case study on the "Yale Shooting" problem to present non- monotonoc reasoning from a proof-theoretic standpoint.

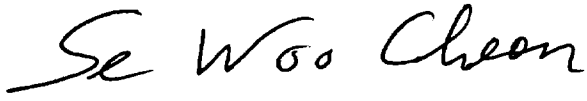
The paper "Weighting Issue in Fuzzy Logic" (Luo et al.) introduces the relative weighted models that are able to cope with weights in fuzzy logic in a sound and efficient manner.

The proposed models are based on T-norms and T-conorms, and satisfy some constraints, thereby overcoming some problems in weighted averaging models.

In conclusion, I hope that the papers in this special issue will provide readers with the glimpse of current research trends in AI worldwide.

I would like to thank Dr. Matjaž Gams who

has co-edited this special issue. I would also like to thank all participants of the PACES/SPICIS '97 Conference, especially Dr. J. K. Lee and Dr. Y. M. Chae.



Se Woo Cheon

Korea Atomic Energy Research Institute
Dukjin 150, Yusong, Taejon 305-353
Korea

GUEST EDITOR

A Safe and Efficient Agent Architecture

Soe-Tsyr Yuan
Information Management Department, Fu-Jen University,
No. 510 Rd. Chung-Chen, Sing-Zoung, Taipei, Taiwan, R.O.C.
E-mail: yuans@tpts1.seed.net.tw
Tel(Fax): +886 2 3693220

Keywords: software agent, safety enforcement, decomposition and causality of partial-order planning, ontology

Edited by: Se Woo Cheon
Received: March 26, 1997 **Revised:** May 14, 1997 **Accepted:** June 6, 1997

When an agent pursues human's goals, the agent should generate an acceptable solution (accordingly protecting humans from harm). We call this necessity safety enforcement. Furthermore, the agent should exercise safety enforcement in a computation tractable manner. In a complex world, the agent often does not have complete information. How can an agent maintain its safety enforcement with incomplete information in a computation tractable manner? This paper presents an agent architecture which can be used to solve the above problem. This framework combines the culture support in the domain ontology and the decomposition and causality of partial-order planning.

1 Introduction

In developing agents, there is a very important but not yet resolved problem - how can an agent generate acceptable actions (while avoiding harmful actions), and do so in computationally tractable manner? We call this solution *safety enforcement* in agents. This problem becomes increasingly pressing as we develop more powerful, complex, and autonomous software agents. For example, an agent has to eliminate travel plans that involve building an airport to take a flight out of the city (or even, stealing money to buy a ticket) because humans are not satisfied with every plan that satisfies their goals, but only a restricted, moral subset of them. In another example, an agent has to eliminate a plan that involves deleting arbitrary files in order to achieve a human goal of reducing the usage of a disk to under 80 percent. That is, an agent must place the protection of humans from harm at a higher priority than obeying human orders.

How to enforce safety in agents adds much more complexity into agent architectures. So far, only Weld [1] proposed a possible solution to this safety problem. Weld assumed a complete information situation and extended the technique of *threat detection* to achieve the agent's safety. It added "don't disturb" constraints in the planning. Before a planner added an action to the plan, the planner needed to check whether the addition of this action would violate any "don't disturb" constraints. As a result, the final plan would satisfy all "don't disturb" constraints to enforce safety. However, this approach sacrifices the efficiency of an agent. Furthermore, when loosing the assumption of complete information, how can this approach adapt to enforce the agent's safety?

In the past, a lot of agent research overemphasized the internal architectures of agents, but paid less attention to the agent environments as well as the structural interactions between agents and their environments. Recently, [2] claimed that it was not necessary to offer a better and newer architecture for overcoming the impasses found in current agent architectures. The most important thing was to focus upon the structures of interaction between agents and their environments. In this paper, we consider such structures of interactions in order to maintain safety enforcement in agents.

We adopt the idea of culture support from [3] and have a culture support domain ontology,

which encodes the safety features in the structure of interaction between an agent and its environment, for enforcing the agent's safety. We use the hierarchical decomposition in partial order causal planning like Young's [4] to achieve the agent's efficiency. We also extend such planning to incorporate conditional effects to handle uncertain situations.

In the next section, we explain how culture support in domain ontology encodes the safety features for an agent and its environment. Section 3 describes the hierarchical decomposition in partial-order causal planning, which will reference the ontology for the generation of safe agent's behaviors. Its extension of conditional effects is stated in section 4. Section 5 explains the machinery of safe agent's behaviors efficiently generated by our agent architecture, and then presents a demonstration by example. Section 6 discusses some related research, and the final section summarizes the work.

2 Culture Support Domain Ontology

Safety enforcement in agents means the behaviors generated from agent programs are morally acceptable. The importance of this issue will grow drastically as agent technologies push forward. However, enforcing safety adds much more complexity into agent architectures. One way out of this dilemma is to find help other than from agent architectures.

In fact, agent's behaviors are not the product of agent architectures only; they are created by both agent architectures and agent environments. [5] presented a framework in which an agent and its environment were modeled as two coupled dynamic systems, and showed that their mutual interaction was in general jointly responsible for the agent's behaviors. That is, safe agent behaviors must be achieved by both the agent architecture and agent environment together. [2] also showed that whatever architecture you employed, you should try to find structure in the environment that fitted the strengths and weaknesses of that architecture. Put another way, when a particular agent architecture ran into trouble, saw if the world contained environment structures that compensated for its weakness. Therefore, it became imperative to discover the features of a given world and the agent's interaction with that world so that we could create a healthy marriage of architecture and environment. From the above remarks made by [5,2], we can use figure 1 to show the *determining relationships* between agent architectures and their environments, their mutual interaction, and agent's behaviors.

The agent architecture we present in this paper is originated from planning. In planning, any ground operator sequence that solves the given problem is considered a solution to the problem. However in many realistic planning problem, not every operator sequence that solves a problem may be an acceptable solution, as the users tend to have strong preferences about the types of solutions they are willing to accept. Handling such restrictions in planning would involve either attempting to change the domain specification (drop operators, or change their preconditions), or implementing complex post-processing filters to remove unwanted solutions. While the second solution is often impractical, the first one can be too restrictive. For example, one way of handling the travel example shown in section 1 is to restrict the domain such that the "airport building" action is not available to the planner. This is too restrictive since there may be other legitimate uses for airport building operations the user may want the planner to consider.

From the above discussion, we know planning the agent architecture has its weakness in handling the agent safety's problem. However, one way out of this dilemma is through discovering the safety features of a given world and the agent's interaction with the world. We call these features *culture support* (this name is according to [3]),and denotes that these features are from the culture of our society). That is, the planner might consult these safety features for the generation of acceptable operations.

The culture support for a domain can be modeled by *domain ontology*. This domain ontology is based on a conceptualization : the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them and the activities allowed in the domain. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. A domain ontology is an

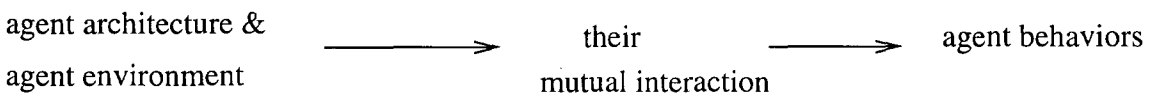


Figure 1: The determining relationships between agent architectures and their environments, their mutual interaction, and agent's behavior.

explicit specification of a conceptualization.

Now that culture support is used to model the safe features of a given world and the agent's interaction with the world, we have to explain what these safety features are. The major role of a safe agent is to act acceptably. As a results, safe acts are the major job. However, how do we enforce safe acts in planning agent architecture? We propose that part of the answer lies in culture, that is, in the *formal properties of a given world's repertoire of artifacts and usual habits of manipulating those artifacts.* Usually, for a given world, the inventory of objects available to the agent and the set of acceptable operations on the objects are determined by the culture of the given world. For example, in a cooking domain, the set of objects includes the materials such as flour, rice, sugar, salt, cake, etc, and the tools like forks, knifes, spoons, stove, pan, etc, and the containers like plates. The acceptable act between a material and a container can be just a 'putting-in' relationship (rather than a 'throwing-in' for example). Next, we will use a cooking domain example to show how the culture support is expressed in a cooking ontology.

2.1 Meta Ontology

In this section, we present the main terms and concepts used to define the cooking world ontology[1]. The cooking ontology is composed of a set of entities, a set of relationships between entities, and a set of activities. Entities can play roles in relationships. An attribute is a special kind of relationship.

- **Entity:** a fundamental thing in the domain being modeled. Examples: a cake is an entity. An entity may participate in relationships with other entities. The attribute 'state' of an entity is a special kind of relationship. For example, the states of an entity 'cake' consist of 'raw', 'dough', and 'cooked'.

- **Relationship:** the way that two or more entities can be associated with each other. Examples: Putting-in is a relationship between a material and a container denoting that the material can be put into the container. A relationship is itself an entity that can participate in further relationships. For example, a cake in 'dough' state is a relationship (denoted as 'dough cake') as well as a cake in 'cooked' state (denoted as 'cooked cake'), and 'dough cake' and 'cooked cake' are further associated in an oven-heating relationship.

- **Roles:** the way in which an entity participates in a relationship. Examples: Container is a role played by an entity in a putting-in relationship. Strictly speaking, the correct way to refer to an entity playing a particular role is to use a phrase like 'the entity playing the container role'. However, for convenience - where ambiguity will not arise - we will often use an abbreviated form such as 'the container'.

- **Activity:** something done over a set of entities. The following pertains to an activity: has an action type, has a list of free variables, has a set of binding constraints on the variables, has preconditions, has effects, is decomposed into more detailed sub-activities, has partial ordering constraints between those sub-activities and this ordering pertains to causal or safety considerations. Examples: 'put-in' act is an activity of type 'putting-in' which is for putting something into another; 'wash' act is an activity of type 'cleaning' which is for cleaning something; 'oven-heat' act is an activity of type 'oven-heating' which is for heating something using an oven, which can be decomposed into warm-up-oven, put-something-into-oven, and bake-something. Warm-up-oven has to precede put-something-into-oven due to safety consideration of avoid-

[1]We follow the conventions used in the [6].

ing something to being dried up too long, but put-something-into-oven preceding bake-something is due to causal consideration.

2.2 Ontology of Cooking World

In this section, we use *hierarchical object-centered representation* with interleaved relationships to represent our cooking ontology. That is, the ontology is organized as a taxonomy with interleaved relationships (portion of the ontology is shown in Fig. 3). The inventory of available objects is formed into an object hierarchy. Objects such as heating-tools would be located under objects such as tools, and objects like ovens and steamers would be located under heating-tools, or objects such as eggs would be located under objects such materials. The usual habits of manipulating those objects are represented by those interleaved relationships. The activities of manipulating objects can be further decomposed into a set of subactivities.

Each object has its own state graph, a nondeterministic finite directed graph whose vertices are called states and whose arcs are called operations. The structure of this graph will depend on what type of object it is. The graph for cakes shown in Fig. 2 has one structure, which might include states corresponding to 'raw', 'dough', and 'cooked'; and the graph for forks will have another structure, which might have the states 'clean' and 'dirty'. The relationships between the states within an object are represented by the links in the state graph of the object. For example, the link between the 'dough' state and the 'cooked' sate within a cake object is a oven-heat operation (rather than a 'boiling' relationship even they both are instances of 'heating'), which denotes performing the operation of 'oven-heat' on 'dough cake' results in a 'cooked cake'. An 'oven-heat' activity on the cake dough in fact can be decomposed into a set of sub-activities, such as warm-up-oven, put-dough-into-oven,bake. The link between the 'dirty' state and the 'clean' state within a fork object is a 'wash' operation, which denotes performing the operation of 'wash' on 'dirty fork' results in a 'clean fork'.

The state structure of a predecessor object can be inherited by its successors . However, the state structure of a more specific object will override the one of its predecessor. Each object has a default state such as the 'cooked' state for cakes and the 'clean' state for forks and plates. When referring an object without particularly specifying its state, it denotes that the object in its default state is being referred.

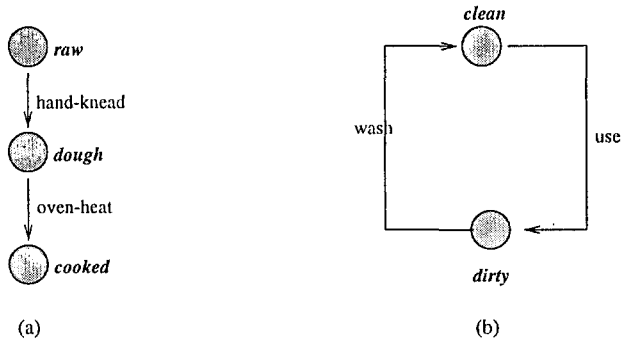


Figure 2: The structures of the state graphs for cakes (a) and for forks (b).

Similarly, the links between objects represent the relationships between the objects. In fact,a link between objects indicates the relationship between objects in their default states. The relationships between predecessor objects can be inherited by its successors. However, the relationship between more specific objects will override the one between their predecessors. For example, there exists a 'putting-in' link between material and container, and such link can be inherited downward between any cake and any plate (i.e. between any 'cooked' cake and any 'clean' plate).

For a single agent of planning architecture, any given user's goals can be thought as either a set of objects with their goal states or a set of goal relationships between objects. For example, a user's goal can be either 'a cooked cake' or 'a cooked cake in a plate'.[2] Any goal state of an object or goal relationship between objects given by users is to be resulted from the practice of some actions. Searching *safe actions* to achieve the goal then might be *efficiently guided by those links relevant to the objects* specified in the goal. (In occasions where the consideration of safe actions is not necessary, the links thereafter do not exist, and the searching of actions to achieve the goal

[2]So far, this paper assumes the agent is given a single goal each time because we consider that there exists an upper layer of goal-ordering reasoning between multiple goals and that such goal-ordering for any specific domain can be done heuristically in order to create efficient plans in that domain.

can then be done by ordinary partial-order planing.) For example, for a goal of 'cooked cake', the relevant link is 'oven-heat' (not 'steamer-heat' or 'boil-heat'), and for a goal of 'cooked cake in a plate', the relevant link is 'putting-in' (not 'throwing-in').

After showing the culture support ontology, the next section will describe the agent architecture we use, which is a decompositional and causal partial-order planner like [4], and it consults this ontology for the generation of acceptable agent behaviors.

3 Hierarchical Decomposition Partial-Order Planner

In this section, we overview the concepts of a hierarchical decomposition partial-order planner , which is a partial-order causal link planner (POCL) that includes action decomposition, and then present our planning algorithm which consults the ontology and imposes specific controls over DPOCL [4] in order to enforce agent safety.

Recent research in AI plan generation was heavily influenced by the development of simple algorithms for POCL planning such as TWEAK [7] and SNLP [8]. Those algorithms based on them (e.g., UCPOP [9]) had been widely accepted as capturing the key insights of earlier planners in a framework that was more amenable to rigorous analysis. However, one aspect of previous work on planning that was not adequately captured in these algorithms is action decomposition - specifying how high-level, abstract actions can be decomposed into more primitive actions.

Plan generation, in fact, involves at least two different kinds of reasoning. First, it involves deciding what actions to use to achieve certain effects (or goals). This process is called causal planning. For example, if you want to achieve the goal of having a cooked cake, you might decide to perform the action of oven-heating on cake-dough. Second, it sometimes involves deciding what actions to perform as a way of performing some higher-level action. This process is called decompositional planning. For example, if you want to perform the action of heating cake-dough, you might first warm up the oven, and then put the cake dough in the oven and bake it. POCL planners like SNLP, perform only the first kind of reasoning; others like Nonlin[16], perform largely the second kind.

Therefore, the process of generating a plan then involves not only establishing causal connections between actions at the same level of abstraction, but also establishing decompositions of the high-level actions in the plan into more primitive ones. Planning with action decomposition is one kind of hierarchical planning. Hierarchical planning has several advantages. First, it can potentially lead to a significant reduction in the amount of search needed [10,11,12]. Second, it can make the task of encoding domain knowledge much easier because the operator writer can reuse operators describing subactions that are common to many actions [13]. Third, hierarchical planning facilitates the interleaving of planning and execution by making it possible to fully expand only some portions of a plan - including those that need to be executed immediately, while deferring the expansion of other portions [14].

The DPOCL algorithm [4] is a partial-order causal link planner (POCL) that includes action decomposition. DPOCL is a sound and complete algorithm for the creation of plans with decompositional as well as causal structures. Furthermore, the constraints on the specification of decomposition schema in DPOCL are less restrictive than previous formal models because the order between the subplans can be partial too.

The process of decomposition is one of creating a subplan from a valid decomposition schema. Each step named in the decomposition schema is added to the plan, either by choosing an existing step of the same action type as the step named in the schema or by instantiating a new step from the library of action operators. Ordering and binding constraints for each step are added and causal links[3] are created between steps where specified by the decomposition.

The major concept of action decomposition is that each precondition of the parent action in a decomposition schema should be needed by

[3]During plan generation, the planner needs to keep track of the decompositional and causal decisions that it makes. That is, a *causal link* is used to record the fact that the purpose of some step is to establish the preconditions of some other step (or the goal), and a *decomposition link* is used to record the fact that the purpose of some step is to be part of a more primitive realization of some other step.

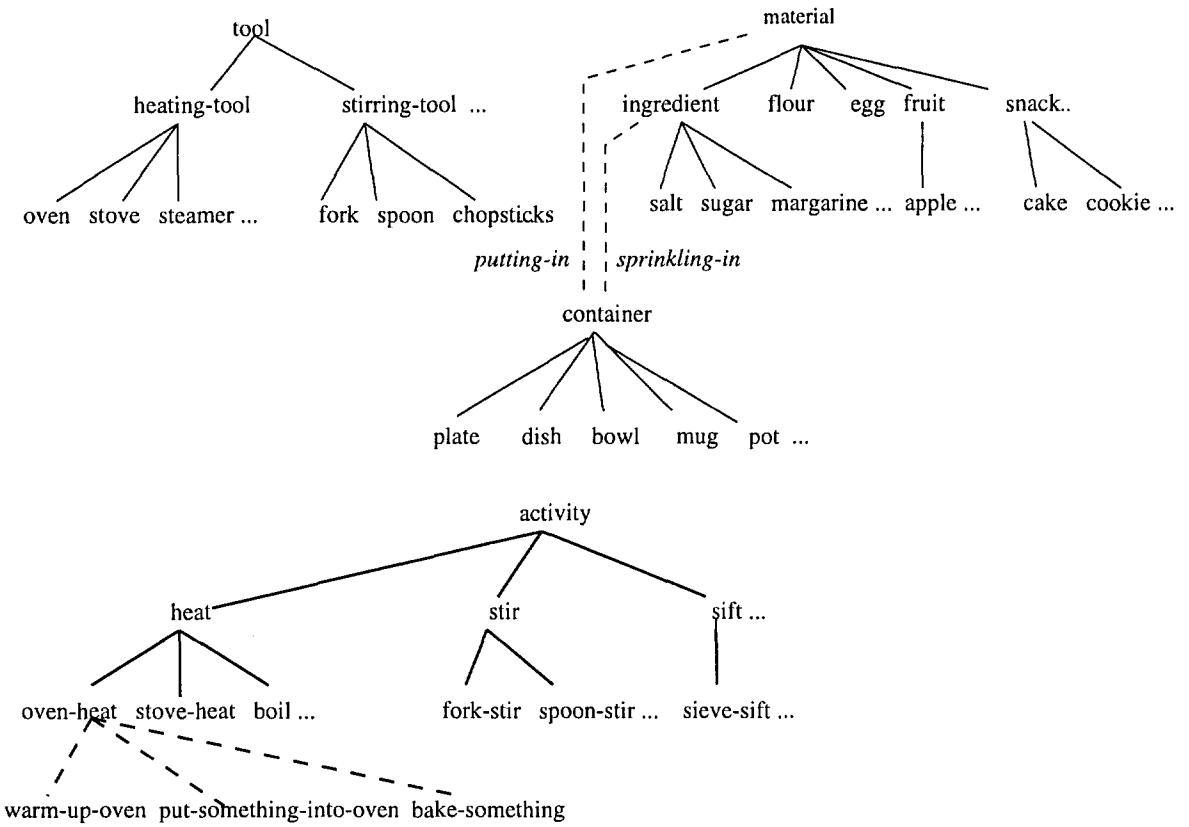


Figure 3: The graphical representation of some portion of our cooking ontology.

some step in the subplan, which itself contributes, through a chain of causal links, to the establishment of one of the parent's effects. The decomposition in DPOCL is without *the completeness of decomposition specification* - every effect of the parent action must be asserted by some step in the decomposition specification and that, there is no possible subsequent action in the decomposition specification that clobbers that effect. DPOCL allows open effects for an action to be established later under the larger context of the plan in order to generate a more efficient plan. However,*we disallow such open effects in an action decomposition* because there might be some danger act in a larger plan asserting that local situation of open effects. That is, we still need the completeness of decomposition specification, and such restriction of course will not influence the soundness and completeness of planning.

For the hierarchical decomposition partial-order planners, a plan is complete if and only if

- For every precondition p of each step s, there is an established causal link.
- All threats are resolved.
- All composite steps are expanded.

Standard POCL planners iterate through a loop in which they first check for a completed plan, then perform plan refinement by adding causal links for open conditions, and finally resolve threats to existing causal links created by recent plan modifications. DPOCL differs from this approach by providing an additional option for the plan refinement phase: the planner may either do causal planning or may do decompositional planning. Either of these options is followed by a threat resolution phase. The algorithm terminates when there are no open conditions and when all the abstract steps have been decomposed into primitive actions.

In its general form, the decision about whether to perform causal or decompositional planning at each iteration is left open: the forms of plan refinement can be fully interleaved. Traditional hierarchical planning first did complete causal planning at a single layer of the hierarchy, and then did decomposition. Control rules that enforce this ordering could be added to DPOCL. However, DPOCL claimed that in general it is advantageous

to allow for interleaving of causal and decompositional planning, especially in situations in which planning and execution must be interleaved.

Next we present our planning algorithm (shown below) which consults the ontology and imposes a special control over the original DPOCL planning for asserting agent safety. This addition of control rules will not influence the soundness and completeness of our planning algorithm. For the details of DPOCL, please refer to that paper directly.

The Algorithm

1. **Termination** If the resulting plan is complete, then return the result.
2. **Plan refinement** Nondeterministically do one of the following:
   - **Causal Planning**
     - **Goal Selection** Nondeterministically select a goal - s_{need} with precondition p and no causal link.
     - **Operator Selection** Nondeterministically obtain a step s_{add} that adds an effect e which can be unified with p. The selection of s_{add} is guided by the links relevant to the objects specified in p if there exist such links in the ontology. (This way of action selection guarantees the generation of acceptable solutions from the very beginning when safety needs to be considered.)
   - **Decompositional Planning**
     - **Action Selection** Nondeterministically select an elaborated action decomposition hierarchy, and select the first unexpanded composite step s_{parent} within this hierarchy in depth-first manner. (This way of expansion guarantees safe execution ordering between steps specified in decomposition schemas in situations in which planning and execution must be interleaved.)
     - **Decompositional Selection** Nondeterministically select decomposition schema D that matches the type of s_{parent} from the ontology. Replace the composite step s_{parent} with the substeps in the decomposition schema.
3. **Threat Resolution**
4. **Recursive Invocation of The Algorithm**

4 Uncertain Agent Environment

When an agent has incomplete information about the initial state of the world , how does the agent come up with an acceptable plan? Unlike UCPOP, DPOCL did not consider the inclusion of action conditional effects. As a result, the expressiveness of DPOCL is its major drawback. Conditional effects mean the effects of actions can be conditional. Conditional effects are useful when we have incomplete information about the initial state of the world, so that we still can try to come up with a plan that would work in any situation that is consistent with the information available. Therefore, in this section, we show how to extend such expressiveness in DPOCL without sacrificing any functionality and complexity.

The tasks of incorporating conditional effects in DPOCL are two-fold: the causal planning part and the decomposition planning part. First, in the causal planning part, it had been a widespread belief that planning with conditional effects is harder than planning with regular STRIPS actions. However, [15] already showed if the planning operators were extended to allow conditional effects, this did not affect the complexity of the planning . His proof was rather straight forward - each conditional operator could be replaced with a number of ordinary STRIPS style operators to obtain an equivalent planning system.

Second, in the decomposition planning part, [12] addressed the problem that there existed one main obstacle in extending conditional effects in decompositional actions of partial-order planners, and thus advocated such extension for future research. Here is the main obstacle mentioned in that paper: each action's preconditions are known when it is added to plan during a reduction, and the reduction specifies how those preconditions are handled. However, conditional effects allows

runtime new preconditions to be introduced whenever such conditional effects are required or confronted. Therefore, in this section, we show how to incorporate conditional effects in decompositional and causal partial-order planning in order to broader its application.

We have known that action decomposition in DPOCL is based on the concept that each precondition of the parent action in a decomposition schema should be needed by some step in the subplan, which itself contributes, through a chain of causal links, to the establishment of one of the parent's effects. In fact, each conditional effect consists of two parts - the precondition part and the effect part. We term these two parts as Cprecondition and Ceffect. Cprecondition regards the uncertain states of initial world. Ceffect regards the effects resulted from the action when Cprecondition is held.

As follows, we show how to extend a reduction planner to handle conditional effects:
Suppose given a parent action:
Precondition: $P_1, P_2, \cdots, P_n$
Effect: $E_1, E_2, \cdots, E_m$, and $Cprecondition + Ceffect$
We can express the parent action as follows:
$P_1, P_2, \cdots, P_n, (Cprecondition) \Rightarrow E_1, E_2, ..., E_m, (Ceffect)$

Suppose this action can be decomposed into a set of subplans - $S_1, S_2, \cdots, S_k$. Since each precondition of the parent action at least should be needed by some step[4] of a subplan and the reduction should specify how those precondition are handled, we can defer Cprecondition into a Cprecondition of some relevant step in a relevant subplan. That is, $P_1, P_2, \cdots, P_n$, will be consumed accordingly by subplans $S_1, S_2, \cdots, S_k$, and Cprecondition will become Cprecondition of some subplan S_i, $1 \leq i \leq k$. This process can be recursive down into the level of primitive actions.

5 Safe and Efficient Machinery

In the above sections, we have shown how the safety features are encoded in the culture support domain ontology as well as how action-decomposition is performed within the POCL framework. What's left is the rationale to justify why our agent architecture is to be safe, efficient, dynamic, and flexible. As follows are the justifications:

1. **Efficiency:** Our agent architecture is a hierarchical planner, and such planner can potentially lead to a significant reduction in the amount of search needed.

2. **Safe:** [15] has shown that the solution languages defined by all partial-order causal planners is a regular language. The solution plans produced by the partial-order causal planners can be described by a regular language (such as $\{a_1|a_2|a_3\}^*$), while acceptable plans are like higher order solution languages (such as $\{a_1^n a_2^n a_3^n\}$). Using partial order causal planners in domains with strong structural constraints is thus akin to using regular expressions to generate strings that are valid sentences in a context free language - although it can be done, it will be very inefficient. Therefore, we use a domain ontology to encode the safe features described in section 2, which will guide the agent toward the safe route from the very beginning (that is, from the highest level of actions) until the end.

3. **Dynamic:** We have incorporated conditional effects in decompositional and causal partial-order planning of our agent architecture. Therefore, the agents are able to handle uncertain information about the initial state of the world. Agents can come up with a plan that would work in any situation that is consistent with the information available.

4. **Flexibility:** Intuitional object-centered representation of goals and actions [5] increases the flexibility of the agent architecture.

An Example

Now we use a very simple example to demonstrate the generation of acceptable agent behaviors from our agent architecture.

Imagine a scenario of a cooking process of making a cake and a cookie. Suppose the agent decides first to make a cake and then a cookie, and

[4]Without loss of generalization, here we assume one step in some subplan consumes this Cprecondition.

[5]The preconditions and effects must be expressed in terms of the properties and relationships of objects.

it percepts all the materials such as flour, sugar, egg, and yeast and the heating tools like ovens are already available. Therefore, the agent first goes through a planning process of making a cake. The agent may interleave its planning and execution of actions.

Goals: 'cooked cake', 'cooked cookie'.

Initial information of the environment: flour, sugar, egg, yeast, oven.[6]

Working memory of our agent: containing all the most current information about the environment. (We presume there exists a perception device of the agent, which is able to record all its most current precepts in the working memory.) As a result, the current content of working memory is just this initial information of the environment shown above.

Next, we use a simplified diagram as in figure 4 to demonstrate the flow of reasoning processes inside the agent.

As follows are the sequence of actions the agent executes for achieving the goals of a cooked cake and a cookie:

1. Get flour, get sugar, get egg, get yeast, stir them into a cake dough.
2. Warm up oven, put dough into oven (when opening the door of the oven, the agent precepts a cookie batter inside the oven, and thus records such information in the working memory.)
3. Bake cake and cookie at the same time.

6 Discussion

[14] was also a SNLP-like partial-order planner with task-decomposition. Its task-decomposition was done via bottom-up plan parsing. It used an incremental parsing algorithm to recognize which partial primitive plans match the decomposition schema. It used hierarchical planning (as a result, it also lead to great search reduction). However, its bottom-up essence prohibited the interleaved planning and execution, in which the decomposition of some higher-level action could be deferred. Furthermore, the most important problem is that this planner is still not safe due to the essence of partial-order causal planning discussed in section 5.

Hierarchical task network (HTN) [16,17] was top-down hierarchical task decomposition. It provided the users a way of exercising control over their solutions. In particular, by allowing non-primitive tasks, and controlling their reduction through user-specified task-reduction schemas. However, we think the HTN style of planning is much more rigid and less flexible, but this might be improved in further research.

We have completed the implementation of our agent architecture, and plan to do some other experiments on new domains in addition to the cooking domain. We believe we will gather more evidences of the worth of our architecture for agent safety. The future research is to extend our agent architecture for a multi-agent environment.

7 Conclusion

Agent safety is becoming a very important issue in agent technology especially when more and more powerful agents are created in reality. However, rare research mentions anything about it. We think the issue of agent safety should be put in the top of our head before designing and developing any agent.

This paper presents an agent architecture which is safe and efficient and is able to work under uncertain agent environment. The basic idea is to combine a domain ontology, encoding the safety features of the world and the interaction between the agent and the world, and a partial-order causal link planner with decomposition to efficiently generate safe agent's behaviors. Furthermore, allowing conditional effects enables an agent to work in a uncertain environment.

References

[1] Weld, Daniel and Etzioni, Oren. (1994) The first law of robotics. *The proceedings of AAAI-94.*

[2] Agre, P.E. (1995) Computational research on interaction and agency. *Artificial Intelligence.*

[3] Agre, P.E. and Horswill, Ian. (1992) Cultural support for improvisation. *The proceedings of AAAI-92.*

[6] Those object names, in fact, represent some specific object instances.

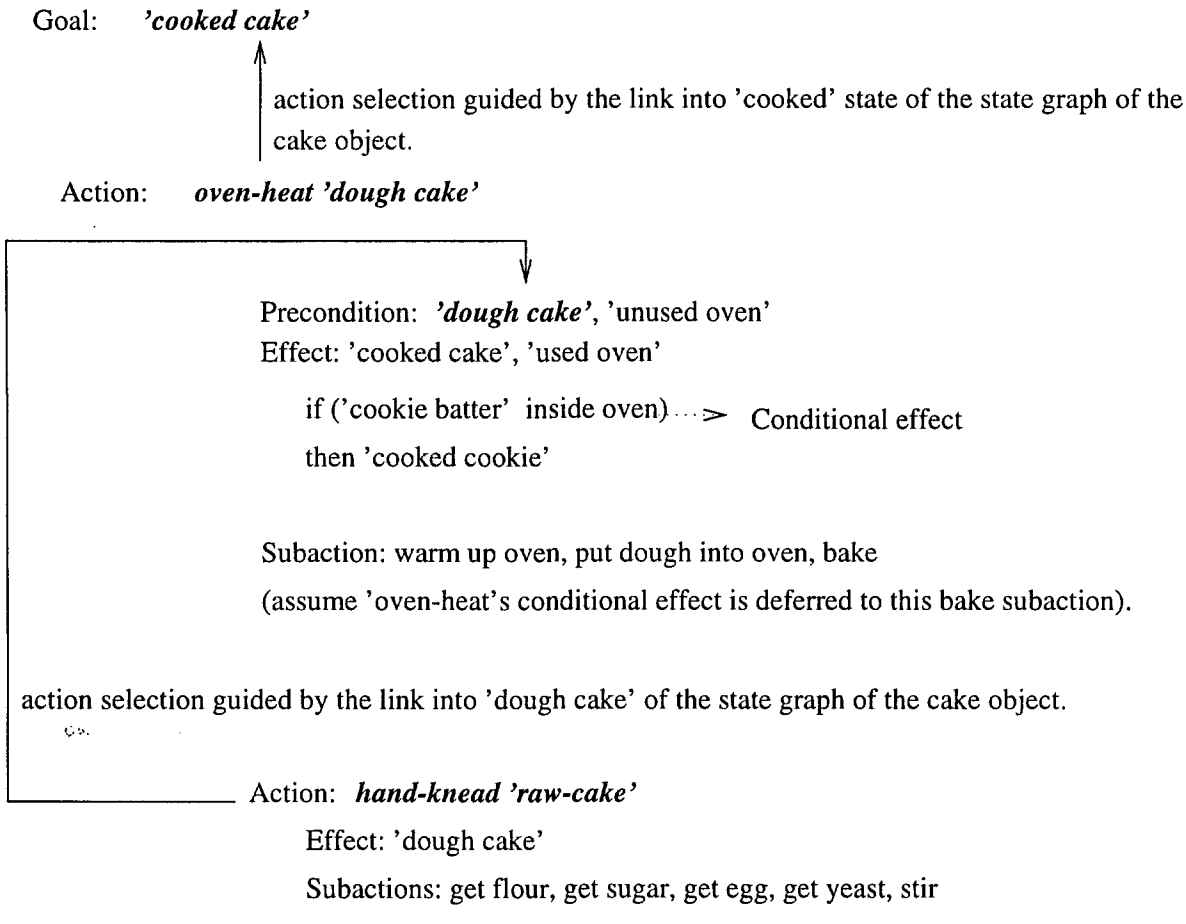


Figure 4: An example

[4] Young, R. Michael and Pollack, Martha E. and Moore, J.D. (1994) Decomposition and causality in partial-order planning. *The proceedings of AIPS-94.*

[5] Beer, R.D. (1995) A dynamical systems perspective on agent-environment. *Artificial Intelligence*, 72, 1-2.

[6] Uschold, Mike and King, Martin, and Moralee, Stuart, and Zorgios, Yannis. (1995) The enterprise ontology.

[7] Chapman, D. (1987) Planning for conjunctive goals. *Artificial Intelligence*, 32.

[8] McAllester, D. and Rosenblitt, D. (1991) Systematic nonlinear planning. *The proceedings of AAAI-91.*

[9] Penberthy, J. and Weld, D. (1992) UCPOP: A sound, complete, partial order planner for ADL. *The proceedings of the conference on knowledge representation and reasoning.*

[10] Yang, Q. (1990) Formalizing planning knowledge for a hierarchical planner. *Computational Intelligence*, 6.

[11] Knoblock, C. A. (1994) Automatically generating abstractions for planning. *Artificial Intelligence.*

[12] Barrett, A. and Weld, D. (1994) Schema parsing: Hierarchical planning for expressive languages. *The proceedings of AAAI-94.*

[13] Hobbs, J. R. (1985) Granularity. *The proceedings of IJCAI-85.*

[14] Bratman, M. E. and Israel, D. J. and Pollack, M.E. (1988) Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4.

[15] Erol, K. and Nau, D.S. and Subrahmanian, V.S. (1995) Complexity, decidability and undecidability results for domain independent planning. *Artificial Intelligence*, 76.

[16] Tate, A. (1977) Generating project networks. *The proceedings of IJCAI-77.*

[17] Currie, K. and Tate, A. (1991) O-Plan: The open planning architecture. *Artificial Intelligence*, 51.

Multi-Agent Systems as a Paradigm for Intelligent System Design

Amal El Fallah Seghrouchni
LIPN - CNRS / URA- 1507 , Institut Galilee
University Paris13- FRANCE
Phone: +33 1 49 40 35 78, Fax: + 33 01 48 26 07 12
E-mail: elfallah@lipn.univ-paris13.fr

Keywords: collaborative problem solving, coordination, distributed planning, interaction, concurrency, multi-agent system, recursive petri nets

Edited by: Se Woo Cheon
Received: May 12, 1997 **Revised:** May 19, 1997 **Accepted:** June 6, 1997

The design of complex and intelligent systems requires an adequate framework that should combine the rigor of formal models, the practicality of existing development methods and the performance analysis of modeling tools. Based on the Multi-Agent System paradigm, this paper proposes an efficient approach to build intelligent systems, the active components of which are cognitive agents (i.e. an agent is endowed with various abilities to solve problems, interact with other agents). In our approach, agent behavior is viewed as an organized collection of abstract actions considered as an abstract control pattern. As agents are willing to cooperate, they exchange their respective plans and coordinate them to cancel negative interactions and to take advantage of helpful ones. The coordination mechanism relies on distributed planning. Concurrent plans are specified through the Recursive Petri Net formalism. A transformation method refines an agent's plan (i.e. control pattern) dynamically thus enabling agents' roles to be predicted and validated formally. The coordination model combines two paradigms: remote and situated coordination that satisfie the following requirements: domain independence, broad coverage of interacting situations, operational coordination semantics and natural expression for the designer.

1 Introduction

Features such as cooperation, interoperability, distribution, data integration and problem solving are fundamental (Bond & Gasser 1988) to build complex and intelligent systems. To support the effective modeling and analysis of inherent complexity, an adequate framework should combine the rigor of formal models, the practicality of existing development methods and the performance analysis of modeling tools. Moreover, it should respect today's accepted main software engineering standards:

- Modularity, to reduce complexity and facilitate development, testing and maintenance.
- Efficiency, to provide rapid executions and inferences by putting independent tasks in parallel.
- Reuse, to prevent redundancies and redefinition.

The major criticism leveled at most approaches to requirement specification is their poor way of capturing and modeling knowledge dealing with all these features. These shortcomings can be attributed partly to the inherent nature of the process of capturing, modeling and verifying domain models and partly to inadequate formalisms used for the representation of these concepts. Requirement specification implies two basic activities: modeling and analysis. Modeling refers to the mapping of real world phenomena onto basic concepts of a requirement specification language. Analysis refers to techniques used to facilitate the communication between requirement engineers and end-users making use of the requirement specification as the basis of that communi-

cation. An adequate approach should rely on the ability to design autonomous and cognitive agents dealing with each of above. Such an approach remains problematic; to respect complex agent system concepts, which are of great importance in Multi-Agent System (MAS) design, the underlying architecture requires three conditions:

- to include different kinds of agents,
- to facilitate the cooperation between agents,
- to ensure coherent activities and validate the global system behavior.

The first requirement may be met through the genericity of an agent entity so as to define a broad range of specialized agents and enable the adequate knowledge representation. The second requirement lies in the coordination mechanisms and efficient interacting methods between agents. The last requirement imposes two additional burdens on intelligent systems: the use of a formal model to specify the agent's activities and formal techniques to verify the agent's interactions.

This paper focuses on the coordination mechanism using the distributed planning alternative. To begin with, we argue that coordination is a necessary mechanism for cognitive agent cooperation specified through concurrent plan coordination. A survey of related work is given and concurrent plan requirements are detailed. An efficient framework for concurrent plan management is then developed. The ordinary Petri Net formalism is extended to the Recursive Petri Net (RPN) (El Fallah Seghrouchni & Haddad 1996b) formalism to meet the distributed planning requirements. An overview of the main features of the RPN formalism is given and the RPN formalism itself is briefly described. Plan management through RPN is illustrated through some modeling aspects which allow the handling of both positive and negative interacting situations. The coordination mechanism is then described through two complementary mechanisms: remote and situated coordination.

2 Coordination as a Bridge for Cooperation

One of the basic postulates of MAS is that no intelligence, and especially no acquisition thereof, is possible without social interactions and without communication with others. Interaction is generally viewed as a collective action allowing agents to cohabit peacefully in the same environment, the sharing of which introduces new problems such as critical resources (consumable or not), negative effects (i.e. action interleaving may cancel or prevent the completion of other agent actions), etc. The main issue addressed by the MAS is how to get a group of agents to carry out coordinated and coherent activities in a dynamic environment. Coordination seems a credible alternative (Osawa & Tokoro 1992) allowing agents both to take advantage of their respective actions (i.e. positive interactions) and to overcome their conflicts (i.e. negative interactions). The main requirements of coordination, such as communication between agents (Searle 1990), recognition of potential interactions between plans (Corkill 1979; Martial 1990; Osawa & Tokoro 1992; Ephrati & Rosenschein 1995), and negotiation between agents (Zlotkin & Rosenschein 1989; Davis & Smith 1983; Bond & Gasser 1988) have already been studied in numerous papers. The main criticisms to be leveled against most of the models proposed in Multi-Agent research are:

- The formal cooperation models are often remote from practical systems and provide only a little help in designing MAS.
- The computing methods used to work out dependencies between agents (between actions, plans, etc.) are often static when they need to be dynamic.
- The coordination process is generally centralized where it should be distributed and implies two agents where it should imply n agents (i.e. not just two by two).

3 Distributed Planning as a Coordination Mechanism

The MAS environment is assumed to be dynamic and shared between agents. Because of the presence of other agents and processes, the environment is subject to continuous changes. A cognitive agent can be considered as an entity continuously receiving perceptual input from the environ-

ment (including other agents) and reasoning by taking actions that affect this environment. Such an agent can be characterized as having a partial representation about the world (e.g. agent's acquaintances), local goals that need to be satisfied, plans to achieve the goals, etc. Hence, a careful coordination and the synchronization of agents' activities (Osawa & Tokoro 1992) are required. The cooperation mechanism proposed here is based on concurrent plan coordination. This approach is in keeping with the general pattern of distributed planning where agents are assumed to be autonomous, i.e. they generate and execute their own plans and the planning process is distributed among them. Distributed planning (i.e. planning-driven agent (Martial 1990)), which concerns the coordination problem constitutes one of the numerous problems highlighted by the design of MAS and may be characterized by two strongly correlated aspects: there is no global plan (i.e. several goals have to be reached simultaneously) and the planning process should be dynamic (i.e. abstract plans and dynamic refinement). Therefore, the interleaving of execution and planning (Ephrati & Rosenschein 1995) for dynamic tasks performed by a group of agents becomes necessary to take into account the dynamic changes (e.g. environment changes, agents' requests, external events, interruption). Different research has been done in distributed planning and in particular incremental planning (Martial 1990) which offers a number of advantages in that the model is a theoretical one which handles both positive and negative interactions. It is however limited by the fact that only two agents (point to point) can be coordinated at a time. In addition, this approach does not provide for the interleaving of planning and execution. The approach put forward by (Alami et al. 1994) is based on the paradigm of plan merging. Although it is robust and handles n agents at a time, it is centralized by the new agent. Moreover, the existing agents are suspended during the coordination process and the new plan is generated in function of existing ones. The approach described in (El Fallah Seghrouchni & Haddad 1996a) proposes a distributed algorithm which coordinates n agents' plans at once but the formalism used may be improved by introducing the notions of abstraction and refinement in the plan. Other research focuses on distributed planning but is based rather on organizational structures (Werner 1989).

One of the major problems in distributed planning remains the definition of an efficient model of actions and plans which can easily and naturally take into account the potential concurrence between the agents' activities (both actions and plans) or between agents and any dynamic processes resulting from the environment. Many up till now well-defined hypotheses in single-agent planning have to be specified again, such as considering an action as a relation between two states of the world, or considering atomic events (Georgeff 1990).

4 Characteristics of Concurrent Plan

The main features of distributed planning is the execution of concurrent plans with overlapping which may concern actions, shared resources, etc. This raises problems such as ordering actions, which is linked to the notion of conflictual resources (i.e. consumable resources or not), non-instantaneous actions, atomic events, etc. Hence, it is necessary to determine whether the plans conflict with each other, i.e. whether they interfere or whether one ruins another's executability conditions. Consequently, coordination requires an adequate model of concurrent plan execution. Such a model should both offer a high expressive power for complex plan representation and provide the necessary mechanisms to control concurrent plan execution.

4.1 Concurrent Plan Representation

A complex plan may be viewed as an organized collection of complex or atomic (i.e. elementary) actions which can be performed sequentially or concurrently in some specific order. A complex plan's actions are subjected to several relations (choice, concurrence, overlapping, etc.) and demand both local and shared resources. A plan model should ensure the standard ways of composing plans such as sequencing, iteration, recursion and choice. In addition, the plan size should remain controllable for the specification phase and plan complexity should remain tractable for validation. This requirement may be attained if

the model enables abstract views by considering a complex plan as being partial and/or abstract where only the relevant information is represented at the earlier phases of plan generation.

Partial Plan: A partial plan may be defined in two ways: the plan actions are detailed but not ordered, or the plan actions are not detailed but are ordered.

Abstract Plan: Abstraction is defined (Tenenberg 1988) as a mapping between representations, from concrete-level to an abstract-level (or to several abstract-levels through repeated mappings). The abstract-level typically does not contain certain concrete-level details which expand the state but which do not have to be considered to obtain an approximate solution. Abstract solutions may correspond to a set of concrete refinements, serving as a basis for plan serving and reuse, reducing the planner's need to solve each problem from first principles. In addition, abstraction reduces the plan complexity since the total complexity is the sum of the complexities of multiple search, and not their product (Minsky 1984).

4.2 Concurrent Plan Execution

In real-world domains, much of the information about how best to achieve a given goal is acquired during plan execution. Two approaches may be considered:

- To build highly conditional plans, which introduces two disadvantages: most of the branches will never be used and the environment is too dynamic, thus requiring replanning.
- To leave low-level tasks to be accomplished by fixed primitive operators that are themselves highly conditional, which relegates the problem to the primitive operators themselves and does not provide any mechanism by which the higher level planner can control their behavior.

The need for models to overcome this problem (at least in part) is acute. The requirement is to develop planning systems that interleave plan formation and plan execution where decisions are deferred until they have to be made. This assumption means that an agent can only acquire more information as time passes and that some degree of deferred decision-making is clearly desirable. A credible alternative is to combine a conditional plan concept at the formation level with a dynamic refinement mechanism at the execution level. Hence, a dynamic refinement may be considered as an elegant way to handle conditional plan as it provides the correspondence between abstract- and concrete-level in the plan. Applied during the plan execution, the refinement displays relevant information dynamically and allows the interleaving of planning and execution which is indispensable for distributed planning.

5 Conceptual Framework

The model proposed here is made up of cognitive agents which represent the active components of the MAS and which evolve in the environment. The positive or negative interactions depend on the agents' behavior and on the environment, the representation of which includes the modeling of resources.

5.1 Agents

Let $Ag = \{g_i, i \in IN\}$ be a set of MAS agents. An agent g_i is defined through the following components:

- the identity of the agent $Id(g_i)$,
- a list of acquaintances $Ac(g_i)$,
- a set of plans,
- an abstract description of the agents' skills $Sk(Ac(g_i))$,
- a set of resources $Res(g_i)$ which includes a sub-set of potential shared resources $SRes(g_i) \subset Res(g_i)$

Remark: Environment may be considered as a special agent the identity of which equals 0 and which shares all its resources with the other agents (i.e. $SRes(g_0) = Res(g_0)$).

5.2 Resources

A resource type represents a domain object which is used to instantiate the variables that appear in an action's conditions. A resource may be shared or not. In addition, many patterns of the same

type of resource may be available. The MAS resources are defined through a set of objects called *Res* such that:
$RES = \{r_i, i \in IN^*\} = \biguplus Res(g_i), g_i \in Ag$,
and two additional attributes:

- the identity (integer) of the resource owner
  $Bel : Res \rightarrow IN$

  $r_i \mapsto j$ where $g_j \in Ag$

- the number of patterns of the resource type
  $Pat : Res \rightarrow IN$

  $r_i \mapsto n$
  where n represents a number of available patterns of the resource r_i.

Remark: This definition includes the extreme cases of consumable resources (i.e. $Pat(r_i) = 0$ after use) and non-consumable resources (i.e. $Pat(r_i) = \infty$).

5.3 Plans

A plan organizes a collection of actions which can be performed sequentially or concurrently in some specific order. Furthermore these actions demand both local and shared resources. A correct plan execution requires that whatever the total order that extends the partial order obtained from the plan, it should remain feasible.

5.4 Actions

A plan involves both elementary actions associated with irreducible tasks (but not instantaneous ones) and complex actions (i.e. abstract views of the task). Semantically, there are three types of actions:

- ◇ An *elementary action* represents an irreducible task which can be performed without any decomposition.
- ◇ An *abstract action*, the execution of which requires its substitution (i.e. refinement) by a new sub-plan. There are two types of abstract actions:
  - ◇ a *parallel action*, which can be performed while another action is being executed,
  - ◇ an *exclusive action*, the execution of which requires its coordination with other current executions.
- ◇ An *end action*, which is necessarily elementary, gives the final results of the performed plan. The plan's goals are implicitly given through its end actions.

5.5 Methods

Intuitively, a method may be viewed as some way to perform an action. Several methods can be associated with an action. A method requires a label, a list of formal parameters to be linked when the method is executed, a set of Pre-Conditions (i.e. the conditions that should be satisfied before the method is executed) and a set of Post-Conditions (i.e. the conditions that will be satisfied after the method has been executed).

Remark: Depending on the action definition, a method may be elementary or abstract. An elementary method calls for a sub-routine to execute the associated action immediately but not instantaneously. An abstract method calls for a sub-plan corresponding to the chosen refinement of the associated abstract action. The refinement occurs so as to detail abstractions and display relevant information. In addition, the planner chooses dynamically the best refinement according to the execution context depending on the pre- and post-conditions.

5.6 Dependence

Several kinds of dependence may be detected since agents evolve in a shared environment. In the following, two types of dependence are distinguished:

- ◇ *Know_How dependence:* such a situation occurs when an agent tries to execute a plan (respectively an action) the refinement of which (respectively the instantiation of the variables that appear in the action conditions) is required. The agent then has to ask another agent for help. He examines the list of his acquaintances and the abstract description of the agents' skills.

◇ *Resource dependence:* an agent tries to execute an action which requires an object (at least one) belonging to another agent. In this case, the agent should know who is the object owner before sending him a request.

6 A Recursive Plan Model

6.1 Motivations

Since the Petri Nets are suitable for modeling, analyzing and prototyping dynamic systems with parallel activities, distributed planning lends itself very well to this formalism. The main contribution we expect from Petri Nets is their ability to improve the representation of complex plans and to allow their dynamic coordination and execution. Applied to distributed planning, the Petri Net model mainly offers the following advantages:

— natural and graphical expression of the synchronization of parallel activities that are the performance of the agents' tasks,

— clear decomposition of processing (transitions) and data sharing (places),

— scheduling of the plans's actions (causal and temporal relations),

— dynamic allocation of tasks,

— qualitative and quantitative analysis of Petri Net modeling of a plan.

The Recursive Petri Net formalism (El Fallah Seghrouchni & Haddad 1996b) is introduced to overcome some limitations of usual categories of Petri Nets (Jensen 1991) (e.g. ordinary Petri Nets, High-Level Petri Nets and even Hierarchical High-Level Petri Nets) that are apparent if one considers a Petri Net as a plan of actions.

6.2 RPN Formalism

This section presents briefly the RPN formalism. Formal definitions are given in (El Fallah & Haddad 1996b). The RPN formalism supports the interleaving of planning and execution since its recursive and hierarchical aspects allow the dynamic refinement of transitions and therefore enable an abstract plan to be considered at multiple levels of abstraction.

6.2.1 Plan Representation through RPN

A plan is represented as an RPN and its behavior is obtained from the firing of the net' transitions.

Recursive Petri Net Definition
An RPN is defined as a tuple $< P, T, Pre, Post, M_0 >$ where:

— P is the set of places

— T is the set of transitions such that:

— $T = T_{elem} \uplus T_{abs} \uplus T_{end}$

— $T_{abs} = T_{par} \uplus T_{exc}$ where $\uplus$ represents the disjoint union

— Pre is the precondition matrix and is a mapping from $P \times T$ to N

— $Post$ is the postcondition matrix and is a mapping from $P \times T$ to N

— M_0 is the initial marking of RPN (mapping from P to N).

Remark: An RPN model represents a plan according to the above definition:

— The state of the RPN (plan) is given by the tokens contained in the places of R.

— A marking represents a current state of the plan and M_0 represents the initial marking which allows the plan execution to start.

— $Pre(p,t)$ (respectively $Post(p,t)$) equals $n > 0$ if the place p is an input (respectively output) place of a transition t and the valuation of the arc linking p to t (respectively t to p) is n. It equals 0 otherwise.

— A transition $t \in T$ models an action of the plan according to the nature of the action (abstract, elementary, etc.).

Remark: The default value of a non-valuated arc equals to 1.

The above definition is augmented to represent an executable plan as follows:

Executable Plan Definition

An executable plan Π is defined as a tuple $< R, Var, Met >$ where:

— R is an RPN which models Π,

— Var is the set of global variables attached to the plan Π,

— Met is the set of method calls associated with the transitions of R.

6.2.2 RPN Semantics: Firing Transition

In RPN formalism, a transition models an action and its firing corresponds to executing an action. The RPN semantics is formally given through the transition firing rules of each type of transition in (El Fallah Seghrouchni & Haddad 1996b). Let us note that an elementary transition firing updates the marking of the net using the pre- and post-matrix as in the case of Ordinary Petri Nets. The effect of an abstract transition firing is to substitute the transition by a sub-net (its refinement) and to update the marking only after the sub-net has been executed (i.e. an end transition has been fired). An end transition firing closes the sub-net and updates the net marking.
A plan is executed by executing its actions. The dynamic refinement of an abstract transition when it has to be fired allows the interleaving of planning and execution.

7 Agent Interaction

Concurrent plan execution imposes two additional burdens on agent coordination which requires both an efficient management of shared resources and an elaborate handling of interacting situations. In our approach, the first requirement will be met through *the remote cooperation mechanism* which allows the allocation of shared resources (simultaneous access to a set of resources) while ensuring both the safety and the liveness of the system. The second requirement leads us to define *a situated coordination mechanism* supported by the RPN semantics. Section 8 shows how to solve the *Know_How dependence* through the situated coordination mechanism. It also presents a distributed algorithm that solves the *resource-dependence*. The algorithm proposed is based on classical techniques often used in distributed operating systems.

7.1 Detection of Interacting Situations

Interacting situations are generally expressed in terms of binary relations between actions and are often detected statically. The interleaving of planning and execution requires both static and dynamic detection of such situations which is ensured through the RPN semantics. The problem now is to answer the two questions:

◇ When should agents' coordination happen? When positive interactions are detected, coordination is desirable and even necessary and may be considered as an optimization of plan execution. On the other hand, when negative interactions are detected coordination becomes indispensable to plan execution.

◇ How may agents' coordination be ensured? In our approach, planning and coordination aspects are merged, thus offering a number of advantages as we shall see in section 8.

7.2 Characterization of Interacting Situations through RPN

The most important interacting situations handled by our approach include both positive and negative interactions.

7.2.1 Positive Interactions

— *Redundant Actions:* actions are redundant if they appear in at least two plans belonging to two different agents, have identical pre-conditions and post-conditions, and the associated methods are instantiated by the same parameters except for the agent parameter who has to perform the action. Hence, coordination assigns action execution to one of the agents. The agent who will perform the action has to provide his results. The others have to modify their plans by including a synchronizing transition.

— *Helpful Actions:* actions are said to be helpful if the execution of one satisfies all or some of the pre-conditions of the others. Their execution will be respectively possible or favored. There are two ways of detecting such a situation:

— dynamic detection: during the execution of an abstract action, the refinement of which encapsulates an elementary action which validates another action' s pre-conditions (i.e. parallel actions),

— static detection: when the execution of one action precedes the execution of another and validates its pre-conditions.

7.2.2 Negative Interactions

— *Harmful Actions:* actions are said to be harmful if the execution of one invalidates all or some of the pre-conditions of the others. Consequently, the execution of the latter will be respectively impossible or at an unfair disadvantage. Such a situation should be detected before the new plan execution starts to predict failure or deadlock. Our coordination mechanism introduces an ordering (i.e. synchronization arcs) between the harmful actions as in (El Fallah Seghrouchni & Haddad 1996a) which provides a coordination algorithm (COA) for handling such interactions between n agents at once.

— *Exclusive Actions:* actions are said to be exclusive if the execution of one momentarily prevents the execution of the others (e.g. their execution requires the same non-consumable resource). Detected dynamically, this situation occurs when an exclusive action has been started since it needs coordination with all other executions. In this case, execution remains possible but is deferred according to the firing rules.

— *Incompatible Actions:* actions are said to be incompatible if the execution of one prevents the execution of the others (e.g. their execution requires the same consumable resource). In our model, such a situation models an alternative (e.g. two transitions share the same input place with one token). In this case, execution remains possible only if the critical resource can be replaced. In our approach, the planner uses heuristics based on two alternatives which may be combined to avoid conflicts: if the conflict concerns an abstract action, the planner tries to substitute the current refinement. Otherwise, the method used will be replaced.

8 The Coordination Mechanism

8.1 Remote Coordination

Imagine that an agent tries to execute an action (i.e. a method) the parameters of which he is unable to instantiate. Ne has two alternatives: to execute the method concerned at a distance or to obtain access to the shared objects. Again, a selective request is addressed to the agent' acquaintances. In the first case, a client/server protocol (based on the Remote Call Procedure (RPC) mechanism) is established between the two agents. In the second case (a shared object), the answer may be permission to access the objects needed as described in the following.

Hypotheses:

Let us assume that the agent g_i requires the set of resources $NR_i \subset Res$ such that:

$$NR_i = \biguplus SR_j \text{ where:}$$

$$SR_j = \{r_i, Bel(r_i) = j \text{ and } r_i \in SRes(g_j)\}$$

The agent g_i sends g_j a request:
$Resource\_Request((r_1, nr_1), \cdots, (r_n, nr_n))$
where: $ir_i \leq Pat(r_i)$

In the following, the variable i will replace g_i to simplify the notation. Let us introduce the necessary concepts (agent state, relations between agents, etc.) for the shared resource management.

- $State(i, r) = \{wait, in, out\}$ with the following semantics:

  wait: the agent g_i is waiting for permission to go into the critical section of r;

  in: g_i is in the critical section of r;

  out: else.

- $Defer(i, r)$ is the set of g_j $(g_j \in Ac(g_i))$ where g_j has sent a *Resource_Request* to g_i the permission for which has been deferred, and $r \in SRes(g_i)$.

- $Use(i, r, j)$ indicates to g_i the number (an overestimation) of resources that are still used by the agent g_j.

– $Priority(i,r)$ is a Boolean which indicates if the agent g_i has priority over the resource r or not.

Let $SR_j = \{x, y\}$ (i.e. g_i needs the resources x and y belonging to the set of potential shared resources of g_j). This algorithm is based on the Lamport'stamp where h_i is a local agent clock. This algorithm is distributed among agents since all the variables and sub-routines are shared between agents.

1. Resource_Request $(< n, x >, < m, y >)$

{The agent updates his stamp before sending a stamped request to all the agents capable of using the required resources, then he waits for permission to go into the critical section.}

begin
$k_i(x) = n$; $k_i(y) = m$;
$last\_stamp(i) = h_i + 1$;
for $z = x, y$ do
$State(i,z) = wait$;
$\forall j \in Ac(Bel(z))$ and $j \neq i$ do
begin
$Use(i,r,j) = Use(i,r,j) + Pat(z)$
Send $Request(z, (last\_stamp(i), i))$ to j
end
wait $(State(i,z) = in)$
endfor
End {Resource_Request}

2. Resource_Free

{The agent releases resources and leaves the critical section. He also informs all the agents for which his answer was deferred.}

begin
for $z = x, y$ do
begin
$State(i,z) = out$
$\forall j \in Defer(i,z)$ do **Send** Free $(k_i(z))$ to j
$Defer(i,z) = \emptyset$
end
end {Resource_Free}.

3. Receive $Free(Y(z))$ **from** j

{The agent updates his perception of other agents (their resource use). If he was waiting and if the access to resources is possible, he enters the critical section.}

begin
for $z = x, y$ do
begin
$Use(i,z,j) = Use(i,z,j) - Y(z)$
if $(State(i,z) = wait)$ and
$(\sum_{1 \leq j \neq i \leq n} Use(i,z,j) + k_i(z) \leq Pat(z))$
then $State(i,z) = in$
endif
end
end {Receive Free}.

4. Receive $Request(z,k,j)$ **from** j

{The agent updates his stamp and checks his priority. If he has priority over the resources, or if he already has the resources, he defers his answer, else he sends his permission.}

begin
$h_i = max(h_i, k)$
for $z = x, y$ do
begin
$Priority(i,z) = (State(i,z) \neq out)$ and
$((last\_stamp(i), i) < (k, j))$
if Not $(Priority(i,z))$ or $(Priority(i,z)$ and $j \in Differ(i,z))$
then **Send** Free $(Pat(z))$ to j
else if $k_i(z) \neq Pat(z)$
then **Send** Free $(Pat(z) - k_i(z))$ to j
endif
$Defer(i,z) = Defer(i,z) \bigcup \{j\}$
endif
endif
end {Receive Free}.

Discussion: This algorithm provides the two fundamental properties often required in distributed systems:

- ◇ Safety which ensures that no deadlock is possible since agents have a perception (updated at each new request) of the use of shared resources and they enter the critical section only if they have permission.
- ◇ Liveness since the stamps used here ensure that each request will be satisfied in a finite time.

8.2 Situated Coordination

An agent starts from his initial plan (i.e. reduced to an abstract transition) and refines it. The coordination mechanism is triggered when an agent (e.g. g_i) has to perform a plan Π_i which is partially instantiated, i.e. some plan methods associated with plan transitions have non-instantiated call parameters. The agent (e.g. g_i) should find

an agent who will execute these methods. He starts a selective communication based on his acquaintances and the description of the agents' skills. Two cases are possible: either the agent receives a positive answer or he never (timeout) receives it. In the first case, he delegates part of his plan and initiates a situated coordination (i.e. coordination takes a place at the agent's who receives a plan) which is based on the plan merging paradigm. In the second case, he initiates a remote coordination mechanism described above (section 8.1). Coordination combines the two complementary mechanisms that are strongly coupled. Let us assume now that a Client g_1 initiates a situated coordination by sending his plan Π_1 to the agent Server. The main phases of situated coordination can be summarized as follows:

- ◇ *Recognition and unification:* the agent who receives Π_1 (e.g. Server) detects the methods that are partially instantiated and then examines his execution tree (i.e. the tree of his current plans) in search of a plan with the same methods. This phase succeeds if he finds an RPN Π_2 where all the methods to be instantiated appear (i.e. non-assigned in Π_1 and assigned in Π_2). Then the Server triggers unification of the methods through their call parameters and instantiation of the variables w.r.t. the two plans. If both unification and instantiation are possible, the Server tries to merge the two plans Π_1 and Π_2.

- ◇ *Structural merging through the transitions:* the Server produces a first merging plan Π_m through the transitions associated with the previous methods and instantiates the call parameters. Then he checks that all the variables have been instantiated and satisfy both the Pre- and Post-Conditions.

- ◇ *Consistency checking:* this phase is the keystone of the coordination mechanism since it checks the feasibility of the new plan which results from the structural merging. It is based on the algorithm using the Pre- and Post Conditions Calculus (PPCC) described in (El Fallah Seghrouchni & Haddad 1996b).

Let us note that negative interactions may appear if the Server receives an other plan from an other Client. This case is not described in this paper. More details about the handling of negative interaction are available in (El Fallah Seghrouchni & Haddad 1996b).

9 Conclusion

This research provides an adequate framework to design intelligent systems. The approach proposed here combines the MAS paradigm and distributed planning. The MAS paradigm allows a modular conception of the system the underlying architecture of which involves cognitive agents. Agent cooperation is ensured through two correlated mechanisms: remote and situated coordination.
The coordination model is dynamic and satisfies the main requirements of MAS design including domain independence, broad coverage of interacting situations, operational coordination semantics and natural expression for the designer.
The RPN formalism offers a number of advantages:

- — representation and reasoning about simultaneous actions and continuous processes (Georgeff 1990) (concurrent actions, choice, alternatives, synchronization, etc.),

- — the formalism is domain-independent and supports complex plans that may contain different levels of abstraction used according to the nature of the operation performed on it,

- — the formalism offers abstraction (i.e. only the relevant information is represented at the earlier phases) and allows modifications with the associated verification (e.g. no structural inconsistency) and valuation methods (e.g. answering time, robustness), before its performance can be continued,

- — recursivity and dynamicity, that are necessary to take into account the interleaving of execution and planning according to environment changes,

- — plan reuse, which allows agents to be able to bypass the planning process in similar situations w.r.t. the execution context (library of abstract plans which represent the basic building blocks of the new plans),

— agents can skip some of the planning actions, detect conflicts in early phases and reduce communication costs,

— execution control is dynamic (i.e. depends on the associated refinement) and therefore minimizes the set of revocable choices (Barrett & Weld 1993) because the instantiation of actions can be deferred,

— plan size remains controllable for the specification phase and plan complexity remains tractable for validation.

References

[1] Alami R., Robert F., Ingrand F., & Suzuku S. (1994). A Paradigm for Plan-Merging and its use for Multi-Robot Cooperation. *in Proceedings of IEEE International Conference on Systems, Man and Cybernetics.* San Antonio, Texas (USA).

[2] Barrett A. & Weld D.S. (1993) Characterizing Subgoal Interactions for Planning. *in Proceedings of IJCAI-93, pp 1388-1393.*

[3] Bond A.H. & Gasser L. (1988) *Reading on DAI.* Morgan Kauffman Publishers, Inc.

[4] Corkill D.D. (1979) Hierarchical Planning in a Distributed Environment.*in Proceedings of IJCAI-79.*

[5] Davis R. & Smith R.(1983) Negotiation as a Metaphor for distributed Problem Solving. *in Proceedings of Artificial Intelligence, vol 20, pp 63-109.*

[6] Decker K.S. & Lesser V.R. (1992) Generalizing the Partial Global Planning Algorithm. *in International Journal on Intelligent Cooperative Information Systems .*

[7] Durfee E.H. & Lesser V.R. (1987) Using partial Global Plans to Coordinate distributed Problem Solvers. *in Proceedings of IJCAI-87.*

[8] Georgeff M.P. (1990). Planning. *Readings in Planning.* Morgan Kaufmann Publishers, Inc. San Mateo, California.

[9] El Fallah Seghrouchni A. & Haddad S. (1996a) A Coordination Algorithm for Multi-Agent Planning. *Proceedings of MAAMAW'96 Workshop.* Published in the series of LNAI:1038. Ed. Springer Verlag. Eindhoven, Netherlands.

[10] El Fallah Seghrouchni A. & Haddad S. (1996b) A Recursive Model for Distributed Planning. *in Proceedings of ICMAS'96 Conference.* AAAI Press, Kyoto, Japan.

[11] El Fallah Seghrouchni A. (1996) Rational Agent Cooperation through Concurrent Plan Coordination. *in Proceedings of DAIMAS'96 Workshop.* Xalapa, Mexico.

[12] Ephrati E. & Rosenschein J.S. (1995) A framework for the interleaving of Execution and Planning for Dynamic Tasks by Multiple Agents. *in Proceedings of ATAL'95.*

[13] Jensen K. (1991) High-level Petri Nets, Theory and Application. Ed. Springer-Verlag.

[14] von Martial F. (1990). Coordination of Plans in a Multi-Agent World by Taking Advantage of the Favor Relation. *in Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence.*

[15] Minsky M. (1984) Steps towards artificial intelligence. *in E.A. Feigenbaum and J. Feldman (Eds).* Computers and Taught (Addison Welsey, Reading, MA).

[16] Osawa E. & Tokoro M. (1992) Collaborative Plan Construction for Multi-Agent Mutual Planning. /it Werner E. & Demazeau Y., DECENTRALIZED A.I.3, Elsevier/North. Holland.

[17] Parnas D. (1972) On the criteria to be used in decomposing systems into modules, commun. *ACM, 15, 12, 1035-1058* .

[18] Searle J.R. (1990) Collective intentions and actions. *in PR Cohen, J. Morgan and ME Pollac (eds.),* Intentions in Communication, pp 401-416. MIT Press.

[19] Tenenberg J. D. (1988) Abstraction in planning. Ph.D. diss and TR 250, Computer Science Dept., Univ. Rochester, New York.

[20] Werner E. (1989) Cooperating agents: a unified theory of communication and social structure. *In L. Gasser and M.N. Huhns (eds.)*, Distributed Artificial Intelligence. Vol II, pp 3-36. Pitman.

[21] Zlotkin G. & Rosenschein J.S.(1989) Negociation and Task Sharing in a Cooperative Domain. In Proceedings of the ninth Workshop on Distributed Artificial Intelligence.

Internet Information Brokering: A Re-Configurable Database Navigation, Data Filter and Export System

Syed Sibte Raza Abidi
School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia
Phone: +60 4 6573335, Fax: +60 4 6573335
E-mail: sraza@cs.usm.my

Keywords: information broker, database navigation, database virtual hierarchy

Edited by: Se Woo Cheon
Received: May 12, 1997 **Revised:** May 19, 1997 **Accepted:** June 6, 1997

The abundant use of the Internet has promoted it as an effective medium for global information sharing, for instance accessing information stored in remote databases that are inter-linked by the Internet. We present an Internet-based database application, a 'Re-configurable Internet Information Broker' (RIIB) that provides (a) the functionality to dynamically connect and interact with remote databases via the Internet; and (b) an 'intelligent' database navigation engine which is based on the notion of database virtual hierarchies – a database navigation mechanism that allows users to navigate the database by envisaging it as a user-defined hierarchical structure.

1 Introduction

The explosive growth of the Internet has radically transformed the norms of information aceess and processing. Growing number of business organisations, government institutions, academics, students and a variety of other users are now embracing the Internet as an apt medium for electronic, infomation-centered communication. Advances in Internet based technologies have generated tremendous opportunities for information-sharing by making publically available a plethora of applications, software and document archives for various disciplines. In the background of such technological advances, now, it remains of interest to explore the possibilities for transpiring 'intelligence' to typical Internet oriented activities. One possible area that demands research attention is the extraction of information from remote databases via the Internet. This should be achieved by utilising 'intelligent' Internet based database navigation mechanisms.

Indeed, today the need to have access to information that is both correct and complete is very real. For strategic reasons, such demands disregard geographical and time constraints – information/data should be available from any database site in the world. Furthermore, the desired data should be found and made available as and when required with the shortest possible time-delay. (Upton & McAfee 1996). To address such demands the research proposal here, then, entails dealing with two key technologies – (a) the Internet and (b) database technology.

In this paper, we present an Internet based database application, a 'Re-configurable Internet Information Broker' (RIIB) – a confluence of Internet protocols, database technology and artificial intelligence (AI) techniques. RIIB provides (a) the functionality to dynamically connect and interact in real-time with any remote database via the Internet; and (b) an 'intelligent' database access engine that incorporates artificial intelligence based database navigation mechanisms to extract relevant information from a variety of databases (Abidi 1996).

The proposed Re-configurable Internet Information Broker is a generic database access and manipulation tool with an open-ended interface that permits it to dynamically connect and retrieve information from a variety of databases. RIIB's open-ended interface is implemented by exploiting Open DataBase Connectivity (ODBC) technology that permits a seamless interaction

with databases that may vary both in structure and implementation platform. Most attractively, the incorporation of Internet protocols allows RIIB to access databases stored at remote sites. On the database navigation front, RIIB is an intuitive, easy-to-use, visual solution to complex data extraction requirements. RIIB provides end-users a completely customisable information access environment that facilitates data extraction from databases without requiring the knowledge of any query language and the need for programming. RIIB queries databases in a way that end-users understand: there is no need to remember table or field names, no typing of queries and no submission of erroneous field values. By way of an user-friendly interactive session, users can build complex compound queries by simply clicking and choosing the various data fields and specifying whatever constraints deemed relevant. Moreover, RIIB introduces the notion of a database 'virtual hierarchy' and the automatic creation of 'child' databases.

2 A Novel Database Navigation Mechanism Based on a Database 'Virtual Hierarchy'

From a database navigation perspective, we argue that a database can be envisaged at two levels (i) the physical level – the flat structure of the database where all records are stored at the same level, and (ii) the 'virtual' hierarchical level, whereby the flat structure of the database is transformed to a tree-like organisation – a virtual taxonomy or a virtual hierarchy, where various fields are hierarchically organised. The term 'virtual taxonomy' has been defined by Woods (1991) in the context of description of concepts in knowledge representation systems such that whenever a system "constructs an explicit collection of concept nodes ... the result is a subgraph of the virtual taxonomy". Woods' motivation for viewing a collection of 'descriptions' this way is that "although its structure is important, one never wants to make it explicit in the memory of a computer".

For database navigational purposes, we introduce a shift towards a dynamic virtual database hierarchy. RIIB provides users the provision to 'virtually' organise the database as a hierarchical structure based on user-specific priorities (Holmes-Higgin, Abidi & Ahmad 1994). Each level of the database virtual hierarchy corresponds to a field of the records stored in the database. Database navigation can then be carried out by following appropriate branches through the virtual hierarchy, without needing to consider the database as a whole. In simple terms, the virtual database hierarchy can be envisaged as the (attribute-oriented) criteria for selecting relevant records from a database. The word virtual is indicative of the fact that the physical structure of a database is not altered, rather the database is just to be viewed as a (virtual) hierarchical structure for navigation purposes. Since the hierarchy is virtual for all purposes, users can dynamically define numerous hierarchical structures of the same database in a variety of ways, i.e., a number of profiles of the same database can be realised by specifying different database virtual hierarchies.

Specification of a user-specific database virtual hierarchy is carried out during a user-friendly dialogue session. First, the user is provided a list of fields available in the database fields. Next, the user can specify the database's virtual hierarchy by simply selecting the fields of interest, and additionally suggesting the relevant values of the selected fields. Fig. 1 shows the start of a database virtual hierarchy specification session, whereas Fig. 2 shows a partially specified database virtual hierarchy (The right-side view shows the selected fields together with the chosen values).

The database's virtual hierarchy is simultaneously built according to the order in which the fields are selected: the top level of the hierarchy is the first selected field and so on. Indeed, different users would like to 'intelligently' view the same database for varying purposes and this can be achieved by specifying different user-specific hierarchies. For instance, if a database containing records describing the attributes of texts (documents) is to be used by a language translator then the top level of the database virtual hierarchy should be the attribute Language (a typical hierarchy for this purpose may look like the one shown in Fig. 3a), whereas if the same database is to be used for studying the nature of specialised texts then the top level is desired to be the attribute Research Area (as shown in Fig. 3b).

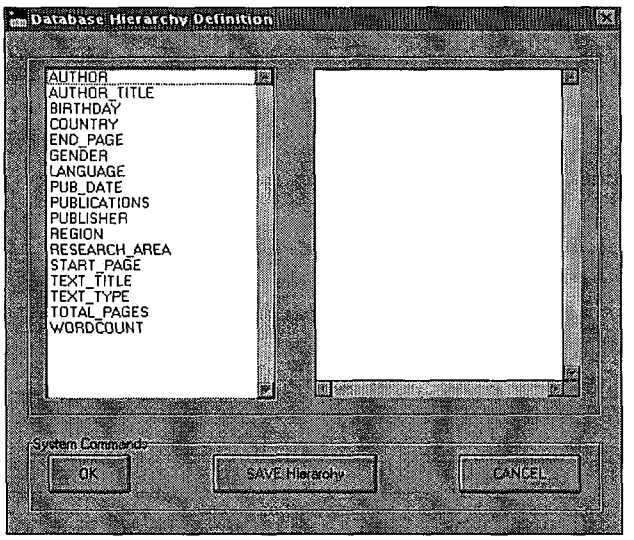


Figure 1: Dialog-box for the specification of a database virtual hierarchy.

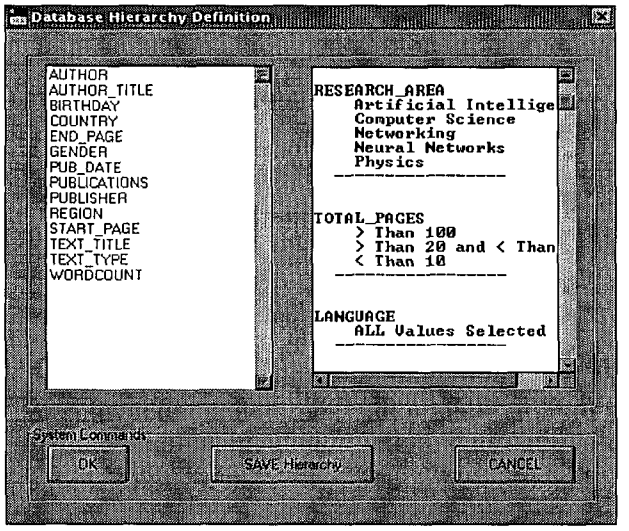


Figure 2: Specification of a database virtual hierarchy in progress.

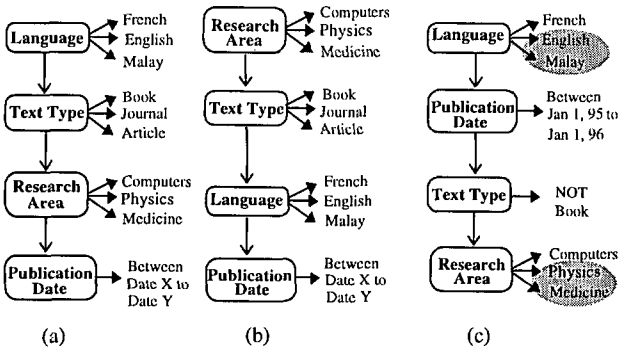


Figure 3: Exemplar virtual hierarchies for a database.

Database navigation initiates from the top level of the database's virtual hierarchy and subsequently proceeds to the next lower level and so on. At the top level (say L1) of the database virtual hierarchy the scope of possible records that can be selected is the entire database. However, as the search proceeds to the next lower level (say L2), the scope of selectable records is reduced to the set of records found at level L1 (i.e. only those records that have satisfied the constraints implemeted at L1 of the database hierarchy). This strategy can be envisaged as the filtering of records (based on the constraints imposed by the hierarchy) as we move down the database's virtual hierarchy. Such a navigation scheme ensures a fine-grained retrieval of highly specific records at the lowest level of the hierarchy.

To elucidate the nature of the navigation mechanism and the scope of record selection consider the database virtual hierarchy shown in Fig. 3c: the top level is the attribute Language with two selected values - English and Malay. The navigation mechanism works as follows: Initially, all records from the entire database that have Language = English OR Malay are selected: say, the retrieved set of records is called Record Set 1. At the next lower level of the hierarchy, i.e. Publication Date, the scope of records to be searched is Record Set 1 and the search mechanism would retrieve from Record Set-1 all records that have an Publication Date = BETWEEN Jan 1 1995 to 1 Jan 1996, yielding say Record Set-2. Therefore, Record-Set 2 is a subset of Record-Set 1. Next, all records that have a Text Type = NOT Book are selected from Record-Set 2, forming say Record Set 3. Finally, at the lowest level of the database's virtual hierarchy, from Record-Set 3 all records that have Research Area = Physics OR Medicine are collected to realise the final collection of records retrieved based on the database's virtual hierarchy. An exemplar database navigation session (using the database hierarchy shown in Fig. 2) is shown in Fig. 4 and Fig. 5. In both these figures, the left-hand view shows the database virtual hierarchy, whereas the right-hand view shows the field values (the highlighted ones are chosen for browsing down). In Fig. 5 it may be noted that as we browse down (the third level of the database virtual hierarchy is shown here) the matched record count decreases, thus

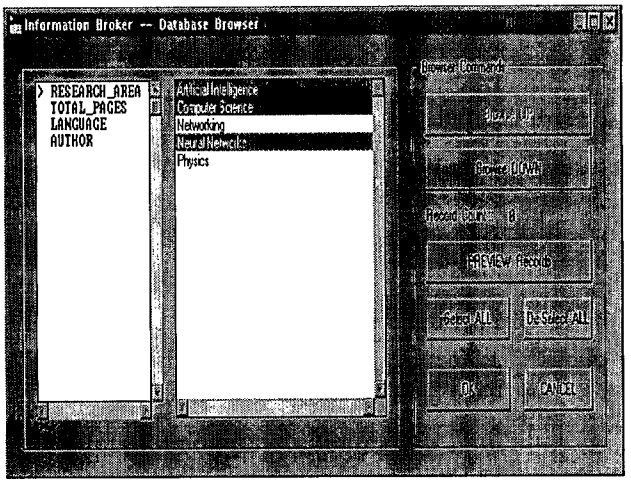


Figure 4: Database navigation based on a virtual database hierarchy.

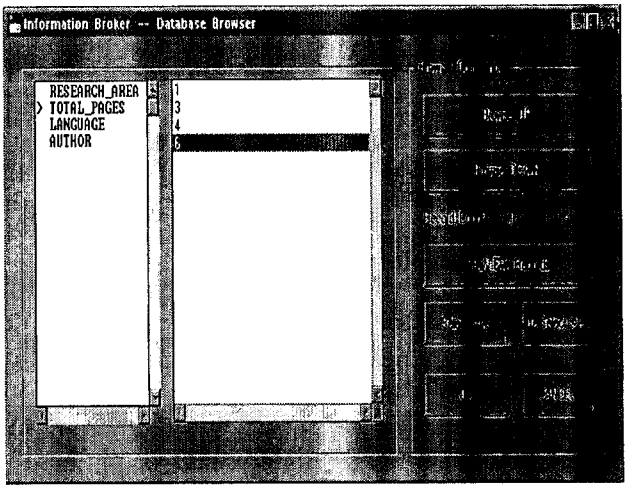


Figure 5: Database navigation: Showing the 3rd level of the database hierarchy.

signifying a fine-grained retrieval of records.

More attractively, database navigation is interactive thus allowing the following features: (i) users can view and select, all or some, the records collected at each level of the hierarchy (the Preview Record button in the database browser shown in Fig. 4 facilitates this option); (ii) users can dynamically customise the already specified database's virtual hierarchy. For instance, the hierarchy in Fig. 3c has Language = English and Malay. Whilst database navigation, the user can constrain the search by selecting only Language = English and not consider the other specified values of the Language attribute. Fig. 4 shows how the database virtual hierarchy is further constrained by selecting only pertinent field values; and (iii) from each level the user can browse both in a forward (next level down in the hierarchy) and backward (previous higher level in the hierarchy) direction, much like web-browsers. We therefore believe that, database navigation based on a database virtual hierarchy facilitates a more systematic visualisation and understanding of data stored in a database.

3 Architecture Of The Re-configurable Internet Information Broker

In architectural terms, RIIB can be envisaged to comprise three main components: (i) Internet Connector; (ii) Metadata Editor and (iii) Database Navigator. RIIB also implements a Database Manager, providing a suite of database management facilities. We briefly discuss below each component. Fig. 6 illustrates the architecture of RIIB.

(i) Internet Connector provides RIIB the necessary interface to connect with the Internet and interact with various 'client' databases. The implementation of open standard protocols facilitates Internet-based transactions. The Common Gateway Interface (CGI) layer, working closely with the Metadata Editor, provides an 'intelligent', dynamically re-configurable Graphical User Interface (GUI) to interact with any connected database.

(ii) Metadata Editor facilitates an internal, functional interface between the physical database and the Database Navigator. The ODBC layer allows the RIIB to connect with a variety of different databases. Other than facilitating the dynamic connection to a variety of databases, the main task of the ODBC layer is to determine the structure of the connected database, i.e. all inherent tables, fields, relationships, keys, etc. This entails the dynamic creation of two files: (i) Data Dictionary file which contains the specification (tables, fields and types) of the connected database; and (ii) Database Map file that contains Prolog based facts describing the entire specification of the database. The Database Map is the most important file as it is responsible for all database navigation operations. Furthermore, the Database Map is also used for

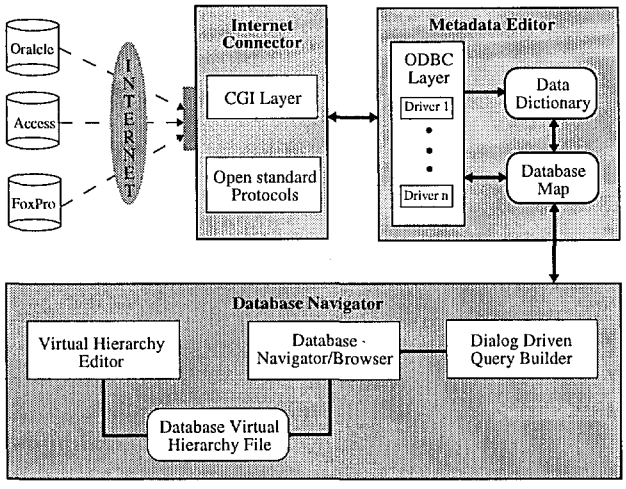


Figure 6: The architecture of the Re-configurable Internet Information Broker.

the creation of the GUI specific to the connected database. The combined usage of the Data Dictionary and the Database Map files thus allows RIIB to reconfigure (on the fly) according to the specification of any connected database system.

(iii) Database Navigator provides an intelligent database navigation engine based on either a virtual hierarchy or user-defined queries. It comprises three main modules: (i) Virtual Hierarchy Editor - a user-friendly dialogue box that allows the specification of user-specific database virtual hierarchies (manifested as the Database Virtual Hierarchy File) shown in Fig. 1 and Fig. 2; (ii) Dialogue Driven Query Builder which provides a dialogue based interface for building compound queries; and (iii) Database Navigator/Browser (shown in Fig. 4 and Fig. 5) which provides a visual illustration of the navigation through the database. When navigating based on a virtual hierarchy, the Database Browser shows (a) the database hierarchy; (b)the current level of the database hierarchy; (c) the values of the field specified at the level in the hierarchy (with the provision to choose the various values in order to dynamically customise the search); (d) records retrieved at each level and (e) other navigational information and options.

Database Manager implements various database management routines. A Electronic Form based approach based on CGI is used for database management.

The Internet Information Broker, near completition, is implemented on the PC platform under the MS Windows environment. The software is developed using the LPA Win-Prolog (version 3.1.) package.

4 Salient Features Of The Re-configurable Internet Information Broker

We discuss below the important database navigation and management features supported by RIIB.

(i) Dialogue Driven Query Building - Click and Choose Database navigation is guided by the constraints imposed by the user's query, however database users vary greatly in their level of database manipulating skills. RIIB relieves the user from the need to know any database query language. Rather, in RIIB both a database virtual hierarchy and a compound query is built in an user-friendly interactive session where dialogue boxes showing the data fields are presented to the user. The user is just required to choose the data fields of interest and specify the constraints, if any. Firstly, to assist an efficient and informed transversal of the database, RIIB makes available all necessary information right at the fingertips of the user. Next, in an intuitive manner, the user can build a database navigation scheme – a virtual database hierarchy or a compound query by simply clicking and choosing the various data fields in a systematic manner. Again, the desird values of the chosen fields can be specified in a Click and Choose manner, thus minimising the specifiation of erroneous values. Fig. 7 and Fig. 8 show the typical dialog-boxes for specifiying constraint values for text and date type fields.

(ii) Open-Ended Interface RIIB provides an open-ended interface that allows it to dynamicaly connect to a variety of database platforms. By exploiting ODBC technology, RIIB can be configured (on the fly) to the specification of any database format and database platform (for instance Oracle, MS Access, FoxPro, etc.). This allows end-users to design their databases using whatever database products and suitable database specification but still be able to exploit the functionality of RIIB. Furthermore, at one time RIIB can interact with more than one

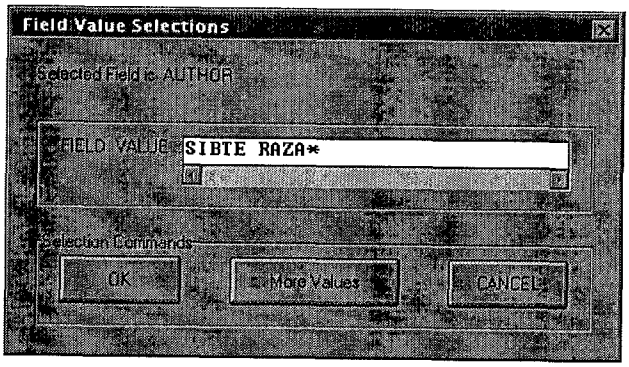


Figure 7: Dialog-box for specifying values for text type fields.

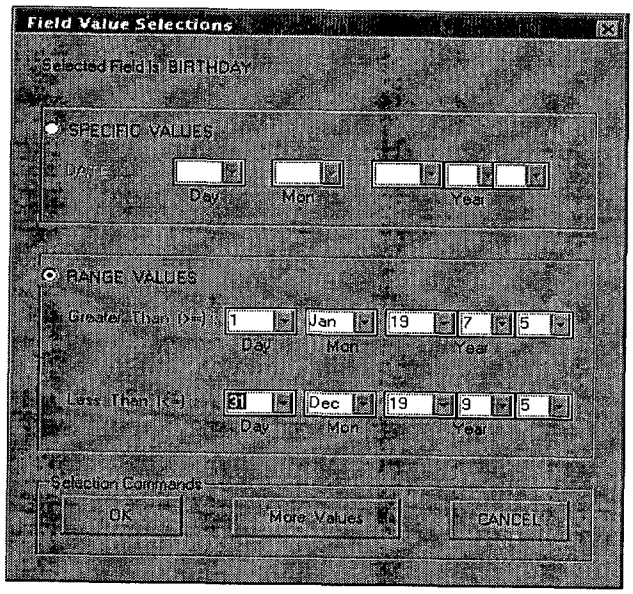


Figure 8: Dialog-box for specifying values for date type fields.

database without any restrictions on the database platforms. For instance, at the same time RIIB can be connected with Oracle, MS Access and Ingres databases, to the extent of transparent data exchange between different database platforms.

(iii) Creation of 'Child' Databases Information brokering requires the efficient transportation of information from the 'parent' source to a secondary source - a 'child' database. RIIB allows the interactive creation of a 'child' database that can be easily exported across various database and word processing platforms. Users may select the records to be exported whilst browsing the database and then proceed to export them in a variety of formats to either a child database, spreadsheet or text document. The 'Preview records' button in the database browser shown in Figs. 4 and 5 can be used to export records to the desired destination.

5 Conclusion

In conclusion, the problem of extracting information from databases is quite real (Geppert & Dittrich 1994) more so if the database in question is situated off-site. We believe that the highest level of functionality one can associate with our Re-configurable Internet Information Broker is **telepresence** - a capability that permits Internet users to access information from a variety of databases resident on computers that may be either on- or off-site via a communication channel of their choice, i.e. from a normal phone to a high-speed connection. More attractively, the RIIB provides a graphical gateway to databases (that may vary both in structure and platforms), thus making the use of databases more easy, efficient, informative and 'intelligent'.

We have proposed the notion of database virtual hierarchies. By incorporating virtual database hierarchies users can impose their own structure on an existing database to retrieve relevant and correct information. The aim of the virtual database hierarchy approach is to provide the flexibility of the traditional attribute-based information retrieval approach, but with the intuitive functionality of an explicit taxonomy approach. For that matter, we have demonstrated how database navigation based on a database's virtual hierarchy can be more informed, user-specific and interactive. We believe that, applications of RIIB could be numerous, such as an 'Electronic' Computer Shopping Mall (a relational database of databases connected via a computer network) or a front-end to a data mining system.

References

[1] Abidi S. S. R. (1996) Towards Information Brokering: An Intelligent Database Navigation and Management System. *Proceedings of the Research & Development in Computer Science and Applications Conference*, Kuala Lumpur.

[2] Geppert, A, Dittrich, K. R. (1994) Constructing the Next 100 Database Management Systems: Like the Handyman or Like the Engineer?. *ACM SIGMOD Record*, 23,1.

[3] Holmes-Higgin, P., Abidi, S. S. R., & Ahmad, K. (1994) Virtual Text Corpora and Their Management. *Proceedings of the in Sixth EURALEX Int. Congress '94*, Amsterdam.

[4] Upton, D. M. & McAfee, A. (1996) The Real Virtual Factory. *Harvard Business Review*, July-August.

[5] Woods, W. A. (1991) Understanding Subsumption and Taxonomy: A Framework for Progress. *John. W. Sowa (eds) Principles of Semantic Networks.* Morgan Kaufmann: California.

MANIS: A Multi-Agent System for Network Information Services

Xiaoqiang Kang and Chunyi Shi
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
Phone: +86 10 6278 5592, Fax: +86 10 6256 2463
E-mail: kangxq@public.bta.net.cn

Keywords: distributed AI, multi-agent systems, network information service

Edited by: Se Woo Cheon
Received: April 23, 1997 **Revised:** May 14, 1997 **Accepted:** June 6, 1997

Network information service has been an important ground for distributed AI research. In this paper, we present a multi-agent system, called MANIS, which integrates several methods for text understanding and offers the text understanding services over network. MANIS is organized as a market-like system. Broker agents can choose among a variety of services offered by server agents based on the sharing information in the service market. On the other hand, server agents can reorganize themselves via decomposition and composition in order to increase the utility of their limited computational resources. MANIS provides an open framework for building multi-agent systems for network information services and will serve as a testbed for our future researches.

1 Introduction

Current users of the Internet are longing for more intelligent information services while they witness an explosive growth in the number and kind of information services offered. For example, a task can be delegated to an agent, and then not only data-level information, but also knowledge-level information can be provided through cooperative distributed problem solving among intelligent agents. Distributed AI (DAI), which is concerned with the cooperative solution of problems in multi-agent systems, can certainly provide theories and methodologies for building systems for network information services. At the same time, network information service is becoming an important ground for DAI research.

Intelligent agents presented in the Internet can be classified as follows:

- *User Assistants*, which assist users in a range of daily activities autonomously (or semi-autonomously). For example, the email agents (Lashkari et al., 1994), and the userbots (Kautz et al., 1994). These agents can be viewed as information consumers.
- *Information Providers*, which offer some kinds of information services to user assistants, users or information brokers (discussed below). The agents, which provide knowledge-level information, may be based on expert systems or knowledge base systems. We refer to information providers as server agents below.
- *Information Brokers*, which are responsible for organizing information and act as media among information providers and consumers. An information broker may provide a limited index of the subject network, or even accepts tasks from user assistants, forms agent teams to solve these tasks and acts as a manager during the cooperative distributed problem solving process. For example, the SHADE and COINS matchmaker (Kuokka & Harada, 1995), which supports content-based information routing.

In this paper, we present a multi-agent system, called MANIS (Multi-Agent Network Information Service), which has integrated several methods for text understanding and offers the text understanding services over network. Our focus is mainly on two problems: how can a broker agent choose among a variety of services offered by server agents and how can a server agent change the amount of the computational resources it owns according to the task load dynamically.

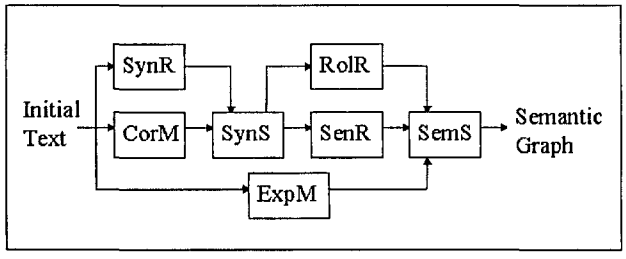


Figure 1: The text understanding methods in MANIS

MANIS is organized as a market-like system. For the first problem, a broker agent in MANIS can choose among services based on the sharing information in the service market. For the second problem, a server agent can reorganize via decomposition and composition. We also present a social agent framework for self-organization, in which an agent is viewed as an organization of intelligent entities competing for the sharing computational resources. This agent framework integrates the social organization theory of DAI and the modeling and implementation theory of Object-Based Concurrent Programming(OBCP) (Gasser & Briot, 1992), and satisfies the needs in the design and implementation of multi-agent systems, especially the systems in the computational task environment as mentioned in (Decker & Lesser, 1993).

The paper is organized as follows: Section 2 describes the methods for text understanding, Section 3 presents the market-like architecture, Section 4 describes the methods for the choice of services, Section 5 presents the social agent framework and the self-organizing process, Section 6 makes a comparison with some related works, and finally Section 7 gives conclusions.

2 Text Understanding Background

Distributed natural language processing (DNLP) can not only provide the solution of higher quality based on cooperation and conflict resolution but also make full use of the geographically distributed resources of natural language processing.

MANIS has integrated several methods for Chinese text understanding, as shown in Figure 1. *SynR* and *CorM* are responsible for syntactic analysis. If the text is news-oriented and not very complex, *SynR* can analyze fast and give the solution of high quality, otherwise, it may give a wrong solution or just fail. As opposed to *SynR*, *CorM* has a large corpus base and processes open corpora. *SynS* synthesizes the solutions provided by *SynR* and *CorM* and thus improves both the quality and efficiency. *SenR*, *RolR* and *ExpM* are responsible for semantic analysis. *SenR* finds the semantic relations among sentences, *RolR* finds the main role of the text and *ExpM* derives the semantic representation of the text according to the pre-defined semantic schemes. *SemS* synthesizes the solutions provided by these three methods and derives the semantic graph of the text based on which agents can offer users some higher level services. For example, more intelligent email sorting services.

3 System Architecture

3.1 Market-Like Architecture

MANIS is organized as a market-like system. This is for the reason of the large scale of such a system and the autonomy of intelligent agents. It has four components: broker agents, service organizations, the service market management subsystem and the computational resource market management subsystem.

Broker agents, referred to simply as brokers below, are responsible for interacting with users and choosing among services offered by service organizations corresponding to information providers.

Every service organization has a representative agent, i.e., the manager. After it has contracted a task with a broker, the manager allocates them to the agents, called solvers, which are responsible for problem solving, and finally sends results back to the broker or the successive service organizations. A degenerated service organization only has a single agent which has all of the basic functions a service organization should have. From the perspective of brokers, the service organization is just a sever agent, referred to as servers below, identified by the manager.

A broker can be formed as an organization of agents too, and an agent can have all of the basic functions of brokers and servers. These are not shown in Figure 2 for simply.

Agents in the service market management subsystem gather information from servers, such as

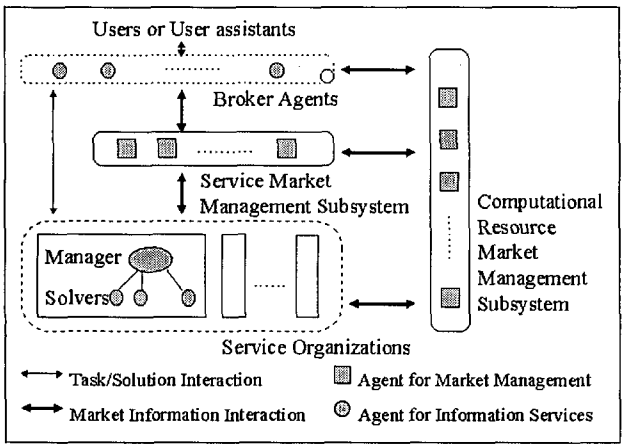


Figure 2: The market-like architecture of MANIS

service price, time, and quality. Every kind of services is corresponding to a submarket which has the current information of that service, including the global information and the local information of every server. Broker agents choose among services based on the sharing information in the service market.

Agents in the computational resource market management subsystem are responsible for gathering information on computational resources including computing speed, network bandwidth, and etc. Currently, we focus on computing speed. Every agent should tradeoff between computing speed and cost. Agents in MANIS can make self-reorganizing decisions based on the sharing information in the computational resource market in order to increase the utility of their limited computational resources.

3.2 Communication and Computational Resource Management

Two kinds of system agents are not shown in Figure 2. They are agents for communication, referred to as *CAs*, and agents for computational resource management, referred to as *RAs*. Every physical host has a *CA* and a *RA* at least, and application agents, including all the agents mentioned above, link to them in a federation architecture, as shown in Figure 3.

Communication among agents is based on message passing. Every agent has a mailbox and *CAs* form a post subsystem responsible for delivering every mail to the mailbox of the destination agent.

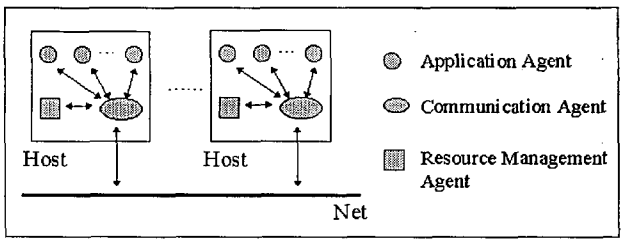


Figure 3: Communication relations among agents in MANIS

There are two approaches of communication. The first one is address-based. In MANIS, the source agent can give multiple addresses in one message. The second approach is content-based which is discussed a lot recently, such as matchmaking based on KQML (Kuokka & Harada, 1995). *CAs* don't support this approach at present, but the service market management subsystem supports it to some extent. For example, a broker can send a mail to the service market in order to announce a task to unknown servers.

Comparing with the agents in the computational resource market management subsystem, *RAs* have more basic management functions and all of the operations on computational resources will really change the current resource state of the system. These resources include the operating system processes, the mailboxes for communication and the sharing memories. *RAs* form a subsystem for basic resource management and support the function of initiating new agents in remote hosts which is important for self-organization.

4 The Choice of Services

In MANIS, we concern with the tasks that need to be solved by multiple agents working cooperatively. The broker, which has such a task, can choose among services based on the information in the service market. Here, we assume that all of the agents are honest, i.e., the information they send to the market or other agents is the true reflection of their states.

The broker chooses among services according to the following steps:

(*S1*) When a broker receives a task t, it derives a service plan according to the demands on quality. The plan, which includes all of the ser-

vices possibly needed, is represented as a graph G_t which is a subgraph of that shown in Figure 1.

(*S2*) The broker decides the initial constraints on time, quality and payment for every service in G_t based on the global information of that service in the service market.

(*S3*) For every service in G_t, the broker tries to find all of the possible servers based on these constraints and the information in the service market, and then chooses among them according to the strategy discussed below in order to maximize its utility.

(*S4*) The broker deletes the redundant services,i.e., the redundant subgraph in G_t.

(*S5*) Now, the broker can evaluate the result. If the result is acceptable, the broker will send it to all of the servers, otherwise, if the constraints can not be adjusted, it will choose the best result from all of the results it has derived and send it, otherwise, it will adjust the constraints and go back to Step 3.

The utility function u_b of brokers is defined as:

$u_b(t,q,p,t_0,q_0,p_0,U_0) = U_0 - k_{t1} \cdot e^{k_{t2}\cdot(t-t_0)} - k_{q1} \cdot e^{-k_{q2}\cdot(q-q_0)} - k_{p1} \cdot e^{k_{p2}\cdot(p-p_0)}$

where, U_0 is the basic utility value, t, q and p are time, quality and payment, t_0, q_0 and p_0 are the basic values of t, q and p, k_{t1}, k_{t2}, k_{q1}, k_{q2}, k_{p1} and k_{p2} are constant coefficients which are decided based on experiments. According to u_b, if t and p are fixed, the utility value increases slowly when q is larger than q_0 and increases, but decreases rapidly when q is smaller than q_0 and decreases.

The strategy for the choice of servers, called $\delta HMC^{\star}$, evolves from the strategies that have been formally defined and experimentally analyzed in the TUMIT testbed (Wang & Shi, 1995). Here, we assume that the information on quality and payment in the service market is exact while the information on computing speed is delayed.

Let x_{ij}^k be server j's computing speed obtained by broker i when it interacts with the service market in the $k-th$ time, x_{ij}^n be server j's current computing speed, and v_{ij}^n be server j's computing speed estimated by broker i , v_{ij}^n can be calculated as:

$$n = 1, v_{ij}^1 = x_{ij}^1;$$
$$n > 1, v_{ij}^n = 1/(f_n/v_{ij}^{n-1} + (1-f_n)/x_{ij}^n),$$
where, $f_n = \delta \cdot (n-1)/n$.

According to v_{ij}^n, broker i can estimate the solving time, and then calculate the utility value. Finally, it will choose the server with the highest utility value. If $\delta = 0$, the broker only allows for the current state of the service market which can not reflect the true state of servers because of the communication delay. If the delay is large enough, the system will oscillate. If $\delta = 1$, the broker allows for all of the historical information. This can lighten the side effect of the communication delay but the choice can not adapt to the environment which changes rapidly. If δ is selected properly in $(0,1)$, $\delta HMC^{\star}$ will have both of the advantages of $0HMC^{\star}$ and $1HMC^{\star}$.

5 Self-organization

All of the agents in MANIS, as shown in Figure 2, can interact with the computational resource market to make the decisions on self-organization. Here, we discuss the self-organization of servers. Based on the social agent framework presented below, servers can reorganize via decomposition and composition. As a result, the distribution of services, including the distribution of agent resources, will be adaptive to the distribution of tasks.

5.1 The Social Agent Framework

We focus on three problems in the design of the agent framework of MANIS. The first one is the sociality of intelligence, that is, we must allow for the existence of multiple participants, multiple selves of an individual, multiple contexts of knowledge and action, and multiple representations. The second one is the resource-limited activities of agents. Because of this, it is important for an agent to have the ability to change the amount of its resources adapting to the task load dynamically. This leads to the third problem, the adaptability of agents.

Let $MAS = (A, E, M, RS, TS, crs, cts)$ be the model of a multi-agent system, where, A is the set of agents, E is the set of intelligent entities, M is the set of methods, RS is the set of possible computational resource states, TS is the set of possible task states, crs is the current resource state, and cts is the current task state.

$M = M_1 \cup M_2 \cup \cdots \cup M_m$, where, m is the number of intelligent entities. For every p in M,

p is represented as a function $p : RS \times TS \longrightarrow RS \times TS$. Here, we use method instead of action because we concern with computational tasks.

$E = E_1 \cup E_2 \cup \cdots \cup E_n$, where, n is the number of agents. For every ent in E, $ent = (ces_e, RE_e, M_e)$, where, ces_e is the current mental state of ent which includes the task state and the resource state inheriting from the agent it belongs to, RE_e is the set of entities in relation with ent, and M_e is the set of methods of ent. There is a scheduling method, i.e., the entity engine, in M_e for scheduling other methods based on ces_e.

$A = \{agt_1, agt_2, \cdots, agt_n\}$, For every agt in A, $agt = (E_a, crs_a, cts_a)$, where, E_a is the set of entities running in agt, crs_a is the current resource state of agt, and cts_a is the current task state of agt. There is a special entity, called RME, in every agent for managing the resources of the agent, especially scheduling the entities according to crs_a and cts_a. This means that entities in an agent compete for the sharing resources of the agent with each other and update crs_a and cts_a, on the other hand, RME schedules these entities, including itself, according to the result of competition instead of some higher level knowledge.

Ideas on designing such a social agent framework are explained as follows:

A multi-agent system is a society of agents, and from a more essential perspective, it is a society of concurrent knowledge and action with limited computational resources, such as computing speed. The social knowledge and action can be represented as intelligent entities. An entity is the individual of the society which has some abstract knowledge and action, especially has the capabilities of combining with the limited computational resources and instantiating itself, but actually hasn't any resource. Therefore, an entity needs the carrier which owns resources to give the full play to its knowledge and action. The carrier is agent. An agent is represented as an organization of some instantiated entities competing for the sharing resources, and engages in organizational computing. From the perspective of other agents, an agent is such an individual that has limited resources and acts as multiple roles in the society of agents. An agent organization is just the expression of the instantiated entity organization in all of the agents involved. Thus, an agent can naturally participate in multiple organizations simultaneously.

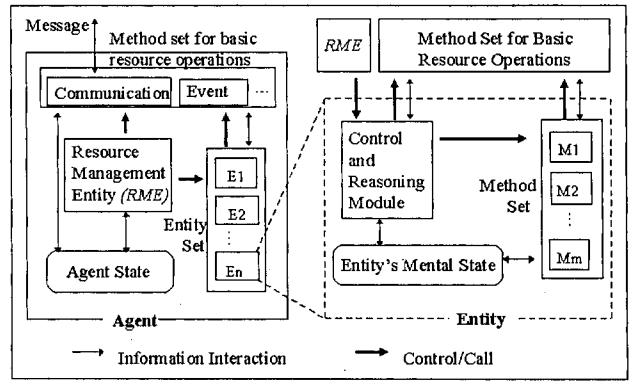


Figure 4: The social agent framework of MANIS

These ideas embody the character of sociality and facilitate the implementation of multi-granularity agents and self-organization. For example, an entity can move from an agent to another agent for more resources, thus the decomposition and composition takes place. On the other hand, entities must instantiate themselves and combine with the computational resources of the agent dynamically, therefore, the problem of resource-limited activities must be considered carefully.

The implementation of the social agent framework in MANIS is shown in Figure 4. Comparing with the typical agent framework, the social agent framework in MANIS engages in organizational computing. The resource management entity needn't have higher level knowledge for scheduling other entities. It just schedules them based on the resource state which is changed by the entities themselves calling for the methods for basic resource operations, which are encapsulated in the class of the resource management entity RME. So entities, including RME, will compete with each other for the sharing resources, and negotiate for conflict resolution.

The basic working loop of RME is as follows:

do{
GetMessageFromMailbox(); // For every message in the mailbox, creates an event and inserts it into the event queue;
GetEventFromEventQueue();
ScheduleEntity(); // Schedules the entity that is responsible for this event;
}*while*(not EXIT);

The basic working loop of entities is as follows:

// Running in the scheduled entity;

switch(Event_Flag){
case MESSAGE_EVENT: // New message from another agent;
CreatTaskEvents(); // Creates new events, which may conflict with other events, to solve the corresponding task;
break;
case EMERGENT_EVENT: // *RME* has found conflicts ;
NegotiateWithOtherEntities();
break;
case NORMAL_EVENT: // An event created before ;
SolvTask(); // May create new events ;
break;
}

The methods for basic resource operations include: (1) methods for gathering information of the resource state of the agent; (2) methods for communication, which are based on the post subsystem, and support both of the inter-agent and intra-agent communication in the same calling forms; (3) methods for the processing resources of the agent, which maintain a event queue and a description of the allocation of the processing time.

The resource management entity *RME* has some private methods for basic resource operations. Based on all these basic operations, it offers some higher level services for other entities, such as initiating, scheduling, gathering information, decomposition and composition, and etc. Decisions on decomposition and composition are made by entities themselves, and *RME* only offers some related services, for example, creating/deleting entities from the description of the agent, and redelivering mails to the new address of that entity which has moved to another agent.

5.2 The Self-Organizing Process

The utility function of servers can be defined as $u_s(T, R, P, S)$, where, T is the task set, R is the distribution of computational resources in time and space, P is the description of the price of resources, and S is the distribution of services. The objective of self-organization is trying to make the actual distribution of resources be as approximate as possible to $R^\star$ and the actual distribution of services be as approximate as possible to $S^\star$, where,

$R^\star = argmax_R(u_s(T_c, R, P_c, S) - u_s(T_c, R_c, P_c, S_c) - cost_s(R_c, R, S_c, S, P_c, T_c))$,
$S^\star = argmax_S(u_s(T_c, R, P_c, S) - u_s(T_c, R_c, P_c, S_c) - cost_s(R_c, R, S_c, S, P_c, T_c))$,

R_c is the current distribution of computational resources, S_c is the current distribution of services, T_c is the current task set, and P_c is the current price of resources, $cost_s(R_c, R, S_c, S, P_c, T_c)$ is the cost of self-organization.

When R_c and S_c are unacceptable, the manager of a service organization will reorganize according to the following steps:

(*S1*) Based on R_c, S_c, P_c and T_c, fixing $R = R_c$, the manager searches for $S^\star$. If $S^\star$ is acceptable, some of the solvers will decompose, and then some of the entities in these solvers will merge with other solvers, go to Step 3.

(*S2*) Based on R_c, S_c, P_c and T_c, the manager searches for $S^\star$ and $R^\star$. If they are acceptable, new solvers may be created, and after the decomposition and composition process as that in Step 1, some solvers may be deleted. In this step, the manager will cooperate with other servers if the requests of these servers in the resource market satisfy its needs.

(*S3*) The manager tries to find the possibilities of cooperating with other servers in the future, i.e., sends the requests for buying or selling resources less than an agent should have at least. The process stops.

When the manager receives a cooperative request for reorganization, it will go directly to Step 2.

The computational resource market, here, not only has the requests for resources, but also offers the price information on resources which is changed according to the resource state and all of the requests.

6 Some related works

DAI foundations and agent frameworks have been considered a lot. Hewitt,C. presented the open information systems semantics theory as the foundation of DAI and emphasized the importance of manageability, scalability, robustness and organization to an open system (Hewitt,1991; Hewitt & Inman, 1991). Gasser L. discussed several principles for DAI, especially emphasized the sociality of knowledge and action, and the resource-limited activities (Gasser,1991). Russell, S. presents the

concept of bounded optimality for defining intelligence that explicitly allows for the limited computational resources of real agents (Russell, 1995). Our social agent framework is influenced by these works and embodies the characters they emphasized. Currently, AOP framework is a typical agent framework. Shoham, Y. presented the AGENT0 language which described situated agents based on the concepts of belief, goal, commitment, etc. (Shoham, 1993). Rao, R.S. and Geogeff, M.P. studied on the BDI framework and is trying to apply it to practice (Rao & Geogeff, 1995). Decker, K. and Lesser, V.R. presented a framework for modeling complex computational task environments (Decker & Lesser, 1993). Comparing with these frameworks, the social agent framework of MANIS views agent as an organization of intelligence entities. OBCP has been suggested as a basis to construct multi-agent systems. For example, in REACTALK system (Giroux & Sentini, 1991), agents are modeled as an organization of three internal actors (duty, actual behavior, and will) and there is only recursive organizational levels. In our framework, intelligence entities are similar to actors, but are not the concurrent units owning computational resources. Thus, the difficulty of implementation in the ACTOR model is reduced.

The most famous work on market-like mechanisms is the Contract Net Protocol (Davis & Smith,1983), which is extended via incorporating some real economic concepts, such as marginal cost (Sandholm, 1993). In (Kraus, 1993), the problem of contracting to a group of agents is also presented but not analyzed in detail. Real computational market is suggested as a general model for distributed computations and is recognized as useful for multi-agent systems (Miller & Drexler,1988). Most of the studies on computational market mechanisms for multi-agent systems are concerned with resource allocation problems. For example, Market-Oriented Programming is for finding an allocation involving multiple interrelated resources (Wellman, 1993). In MANIS, we use market mechanisms for choosing among services, and supporting the decisions on self-organization.

In (Mullen & Wellman, 1995), a simple computational market is proposed, which is focused on where and when to setup mirror sites in the Internet in order to reduce the costs of information transmission. Our focus, however, is on choosing and reorganizing services through market-based coordination. Further more, the services offered by MANIS need to be solved by multiple agents. This is related to the research on cooperative distributed expert systems. In (Itoh et al., 1995), self-organization in the learning phase is analyzed, while we concern with the problem solving phase.

7 Conclusion

Network information service is an important ground for DAI research. The Objective of MANIS is to build a multi-agent system for network information services as well as a testbed for the experimental research on DAI. MANIS is an ongoing project that has integrated several methods for text understanding and has implemented a prototypic system for network information services programmed in C++ language on the platform of Solaris2.4. The works of MANIS are mainly on three areas:

- *Agent frameworks*: a social agent framework is presented which integrates the DAI social organization theory and the OBCP theory by viewing agent as an organization of intelligent entities.
- *Market-like organization and market-based coordination*: agents in MANIS are organized as a market-like system in which a service market and a computational resource market is offered, and broker agents and server agents make decisions on the choice of services and self-organization based on the market information. Thus, the rational allocation of tasks and the rational distribution of services can be achieved based on the local decisions of agents.
- *Network information services*: the services offered by MANIS is on knowledge-level, which need the cooperation of multiple intelligent agents. The methods for text understanding integrated in MANIS can provide the semantic graph of a text containing both the semantic and syntactic information, based on which further processes on the text can be implemented, such as email sorting.

Future works are in progress: (1) the formal description of the social agent framework; (2) the integration with the general agent communication languages, such as KQML; (3) the coordination strategies; (4) the services not limited to text un-

derstanding.

References

[1] Davis, R. & Smith, R. (1983) Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, 20, p. 63-109.

[2] Decker, K. & Lesser, V. (1993) Quantitative Modeling of Complex Computational Task Environments. *Proceedings of the AAAI'94 Conference*, p. 217-224.

[3] Gasser, L. (1991) Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems Semantics. *Artificial Intelligence*, 47, p. 79-106.

[4] Gasser, L. & Briot, J. (1992) Object-Based Concurrent Programming and Distributed Artificial Intelligence. *Distributed Artificial Intelligence: Theory and Praxis*, Kluwer Academy Publish, p. 81-107.

[5] Giroux, S. & Sentini, A. (1991) A Distributed Artificial Intelligence Approach to Behavioral Simulation. *Proceedings of the European Simulation Multiconference'91*, Copenhagan, Denmark.

[6] Hewitt, C. (1991a) Open Information Systems Semantics for Distributed Artificial intelligence. *Artificial Intelligence*, 47, p. 79-106.

[7] Hewitt, C. & Inman, J. (1991b). DAI Betwixt and Between: From Intelligent Agents to Open Systems Science. *IEEE Trans. on System, Man, and Cybernetics*, 21, p. 1409-1419.

[8] Itoh, T., Watanabe, T. & Yamaguchi, T. (1995) Self-Organizational Approach for Integration of Distributed Expert Systems. *Proceedings of the ICMAS'95 Conference*, Sanfrancisco, USA, p. 209-216.

[9] Lashkari,Y., Metral, M. & Maes, P. (1994) Collaborative Interface Agents. *Proceedings of the AAAI'94 Conference*, p. 444-449.

[10] Kraus, S. (1993) Agent Contracting Task in Non-Collaborative Environments. *Proceedings of the AAAI'93 Conference*, p.243-248.

[11] Kuokka, D. & Harada L. (1995) Matchmaking for Information Agents. *Proceedings of the IJCAI'95 Conference*, p. 672-678.

[12] Kautz, H. & Selman, H. (1994) An Experiment in the Design of Software Agents. *Proceedings of the AAAI'94 Conference*, p. 438-443.

[13] Miller, M.S. & Drexler, K.E. (1988) Markets and Computation: Agoric Open Systems. *The Ecology of Computation*, North-Holland Publishing Company, Amsterdam, p. 177-205.

[14] Mullen, T. & Wellman, M. P. (1995) A Simple Computational Market for Network Information Services. *Proceedings of the ICMAS'95 Conference*, Sanfrancisco, USA, p.283-289.

[15] Rao, A. S. & Geogeff, M. P. (1995) BDI Agent: From Theory to Practice. *Proceedings of the ICMAS'95 Conference*, p. 312-319.

[16] Russel, S. J. (1995) Rationality and Intelligence. *Proceedings of the IJCAI-95 Conference*, p.950-957.

[17] Sandholm, T. (1993) An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. *Proceedings of the AAAI'93 Conference*, p. 256-262.

[18] Shoham, Y. (1993) Agent Oriented Programming. *Artificial Intelligence*, 60, p. 51-92.

[19] Wang, X. & Shi, C. (1995) A Text Understanding Oriented Multiagent Dynamic Interaction Testbed: Theoretic Framework, System Architecture and Experimentation. *Proceedings of IEEE International Conference on System, Man and Cybernetics*, p. 800-805.

[20] Wellman, M. P. (1993) A Market-Oriented Programming Environment and Its Application to Distributed Multicommodity Flow Problems. *Journal of Artificial Intelligence Research*, 1, p. 1-22.

Cognitive Simulation of Operator's Diagnostic Strategies in Nuclear Power Plants

Se Woo Cheon, Jeong Woon Lee, and Bong Sik Sim
Korea Atomic Energy Research Institute
Yu Song P.O. Box 105, Taejon 305-600, Korea
Phone: +82 42 868 2261, Fax: +82 42 868 8357
E-mail: swcheon@nanum.kaeri.re.kr
AND
Jin Kyun Park and Soon Heung Chang
Korea Advanced Institute of Science and Technology
Kusong Dong 373-1, Yu Song Gu, Taejon 305-701, Korea

Keywords: cognitive simulation, diagnostic strategies, blackboard architecture, nuclear power plants

Edited by: Matjaž Gams
Received: May 5, 1997 **Revised:** May 12, 1997 **Accepted:** June 6, 1997

This paper describes an approach to simulate operator's diagnostic strategies under emergency situation of nuclear power plants. To assess operator's cognitive workload during the diagnosis, we have developed a diagnostic model, which is an embedded module of a task simulation analyzer, SACOM. This model is based on a blackboard architecture and uses four types of diagnostic agents. The diagnostic agents can simulate typical diagnostic strategies used by human operators: i.e., data-driven search, symptomatic search, hypothesis-driven search, and topographic search. The model is developed using a G2 expert system tool.

1 Introduction

Computer modeling of operator's cognitive behavior using knowledge-based techniques is a promising approach to study human factors and to assess operator's performance (Cacciabue et al., 1992; Corker & Smith, 1993; Fujita et al., 1993; Furuta & Kondo, 1993; Hollnagel & Cacciabue, 1993; Schryver & Palko, 1988; Woods et al., 1987). Especially, modeling of operator's diagnostic behavior is very important in the area of cognitive models.

In this paper, we describe our approach to simulate operator's diagnostic behavior under emergency situation of nuclear power plants. The diagnostic model is an embedded module of a task simulation analyzer, SACOM (Cheon et al., 1995 & 1997). The objective of SACOM is to assess operator's task performance (i.e., both the cognitive workload and the physical workload) at the main control rooms of nuclear power plants.

In the diagnostic model, a blackboard architecture is employed as an inferential architecture (Hayes-Roth, 1985; Park & Wilkins, 1992). The blackboard is a global database containing input data, partial solutions, and other data in various problem-solving states.

The diagnostic behavior of human operators can be simulated by employing at least four types of diagnostic strategies: i.e., data-driven search, symptomatic search, hypothesis-driven search, and topographic search (Meister & Hogg, 1995). The diagnostic model should exhibit such diverse diagnostic behavior by implementing these strategies. The diagnostic strategies are implemented on the agents that consist of a set of meta-rules.

The model has been implemented on a HP 9000 workstation using a G2 expert system tool (GENSYM, 1992). G2 is a tool for developing and running real-time expert systems for complex applications that require continuous and intelligent monitoring, diagnosis, and control.

Section 2 describes the diagnostic strategies of operators. Section 3 describes the development of

a diagnostic model including structure of the diagnostic model, implementation of the diagnostic strategies, the simulation scenario, and the diagnostic knowledge. Prototype simulation follows in Sec. 4. Finally, conclusions and further works are drawn in Sec. 5.

2 Diagnostic Strategies of Operators

Figure 1 shows a typical process of operator's diagnostic and recovery tasks. The cognitive behavior of an operator is typically decided along the long route via monitoring, observation, plant state identification, goal selection, procedure selection, and procedure execution (Rasmussen, 1986). During the plant state identification, operators use the following four types of diagnostic strategies (Rasmussen, 1991; Meister & Hogg, 1995).

- *Data-driven search*: Operators collect information from the plant state, and examines a piece of data to determine if it is worth closer attention. This examination is done by comparing the data to expected plant state behavior. The data-driven search is made when the operator was not quite sure what was happening or where problem was, although something was happening. This strategy may be time-consuming and dependent on the quality of information the operator receives from the plant process.
- *Symptomatic search*: Operators search through a number of remembered event symptom patterns to try to match an observed symptom pattern. This is done until the plant state associated with the symptom pattern can be identified. If there is ambiguity, more observations need to be collected. One drawback of this strategy is that the symptoms of a fault may be very similar to the symptoms of other faults.
- *Hypothesis-driven search*: Operators may generate a set of plausible hypotheses according to observed malfunctioned states and their own heuristic knowledge, evaluate them in a knowledge-based way, and select a diagnostic action based on the selected best hypothesis. The hypotheses using this strategy can be generated from insufficient or incorrect data.
- *Topographic search*: Operators start with a normal model of the system in a plant rather than

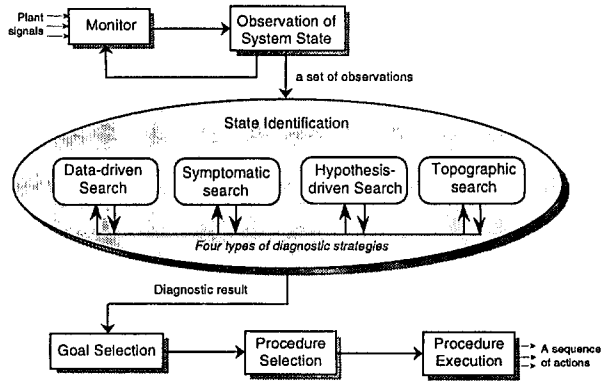


Figure 1: A schematic diagram of an operator's diagnostic and recovery tasks

models of malfunctions. This strategy is used to select the next field of attention. Each field is judged good or bad through observation of the different parts of the system's function. There may be functional distance between the symptom and its cause.

Generally, operators can change between the diagnostic strategies while he/she is solving the problem. Also, operators can use more than one strategy at the same time, at different levels, and they are switching from one to another during their work on the plant.

3 Development of a Diagnostic Model

3.1 Structure of the Diagnostic Model

As shown in Fig. 2, the diagnostic model consists of four components: a blackboard, a controller, agents for diagnostic strategies, a strategy controller, and a diagnostic knowledge base.

The blackboard is a global database containing input data, partial solutions, and other data in various problem-solving states. The controller monitors the changes on the blackboard and decides what actions to take next. It uses various kinds of schedule information to determine the focus of attention.

The agents are independent modules that contain the knowledge needed to solve the problem. The model can simulate the four types of the diagnostic strategies by using the agents. A selected agent requests to the strategy controller to access the diagnostic knowledge base.

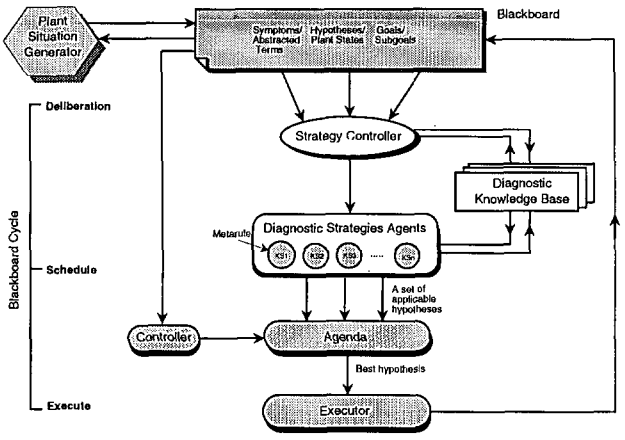


Figure 2: The structure of the diagnostic model

The strategy controller evaluates each of the diagnostic strategies by applying knowledge stored in the diagnostic knowledge base and selecting one agent to pursue in a given state. All the results of monitoring values, observations, and inference results are posted on the blackboard which will be accessed by the strategy controller to make a decision on the selection of an agent.

The model uses a 'generate-evaluate-activate' method. First, based on the current diagnostic state, a set of characteristic features are generated. Second, each diagnostic strategy is evaluated based on the generated features. Third, the selected strategy is activated to change the context of the blackboard which represents the dynamic state of the diagnosis.

3.2 Implementation of the Diagnostic Strategies

For the implementation of the diagnostic strategies, we use about 50 strategic meta-rules to collect symptoms, generate/refine/group hypotheses, generate active hypotheses, and differentiate active hypotheses.

A diagnostic agent is composed of a set of meta-rules. The meta-rules used in the agents are listed in Table 1. The 'clarify_finding' meta-rule finds out more specific and clarifying information based on a selected finding. The 'differentiate_hypotheses' meta-rule attempts to seek for a finding that can discriminate between two active hypotheses.

Each meta-rule is composed of three parts: a triggering body, a precondition body and an action body. In the triggering body, the initial conditions by which the meta-rule can help the reasoning process are evaluated. A meta-rule, even if triggered by some event, is not always able to perform its action body directly; supplementary data may have to be acquired to fulfill the conditions of execution. These required conditions are defined in the precondition body of the meta-rule. The action body consists of the knowledge on how to perform the inference mechanism on the data that match its triggering and precondition parts, respectively.

Table 1: Meta-rules used in the agents

metarules \ diagnostic agents	Hypothesis-driven	Symptomatic Search	Data-driven Search	Topographic Search
collect_initial_data	O	O	O	O
clarify_finding	O	O	O	O
abstract_finding	O	O	O	O
apply_rule	O	O	O	O
generate_hypotheses	O			
test_hypothesis	O			
refine_hypothesis	O			
group_hypothesis	O			
differentiate_hypotheses	O			
abstract situation		O		
generate_similar_cases		O		
test_case		O		
differentiate_case		O		
apply_decision_table			O	
select_next_location				O
find_location_in_system				O
do_test				O
observe_test				O
make_judgement				O

3.3 The Simulation Scenario

The fundamental task of operators in emergency operation is to diagnose the plant state and implement the correct recovery strategy through emergency operating procedures (EOPs) which are like a prescribed action sequence for operator's behavior (WESTINGHOUSE, 1987).

For the simulation of the diagnostic model, an interfacing-system loss of coolant accident (ISLOCA) was implemented. Figure 3 shows a system diagram related to the ISLOCA scenario. This scenario is initiated by an internal leak of two isolation valves from the high pressure reactor coolant system (RCS) to the low pressure residual heat removal system (RHRS). Because initial symptoms are typical of a loss of coolant accident (LOCA) inside containment, this scenario may be difficult from the point of view of diagnosis.

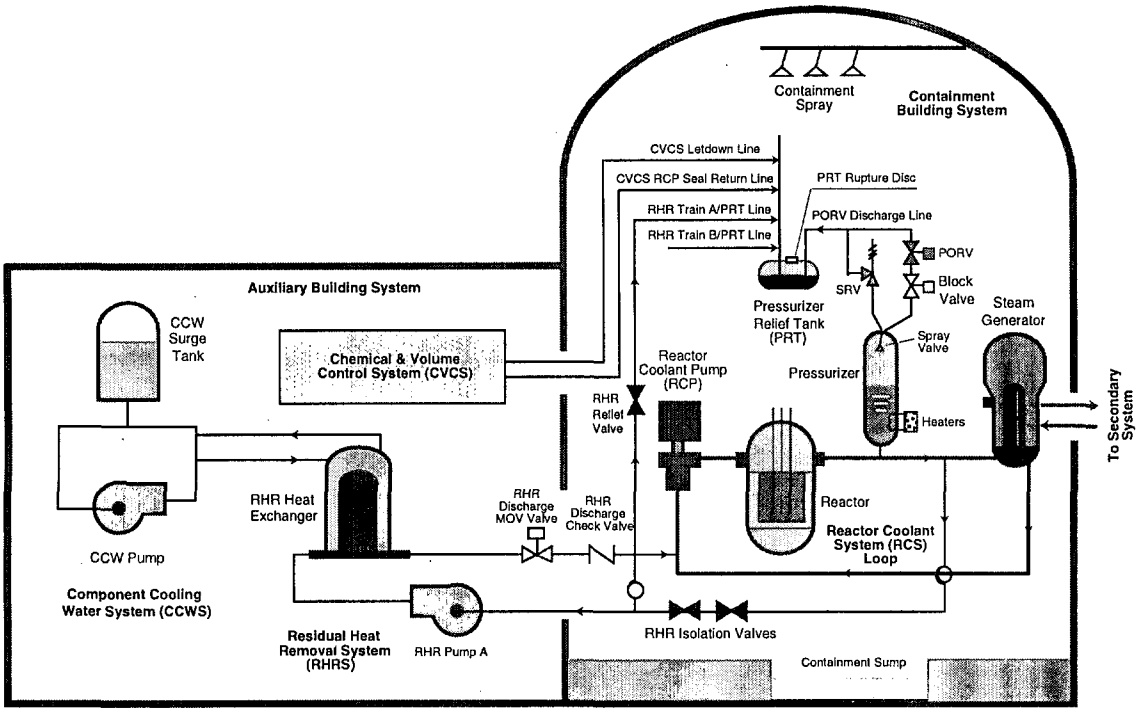


Figure 3: A system diagram related to the ISLOCA scenario

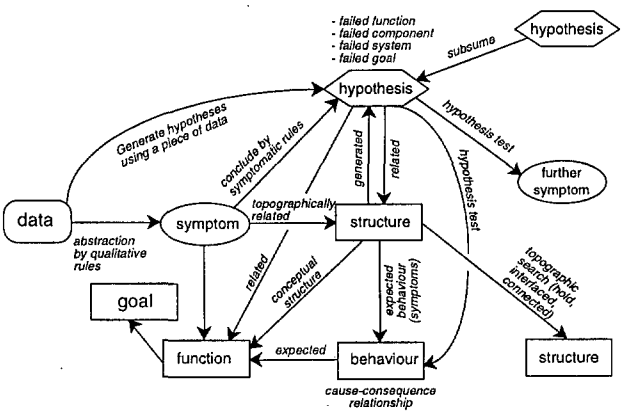


Figure 4: A diagram depicting the diagnostic reasoning flow

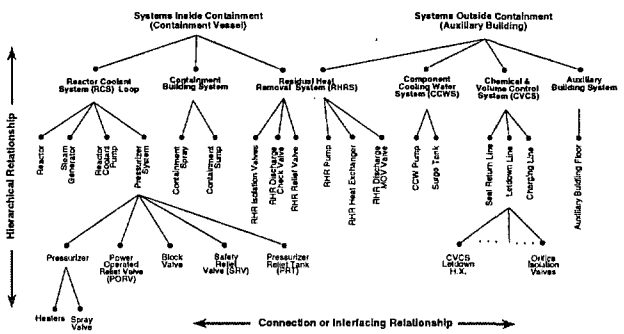


Figure 5: A conceptual structural model for the ISLOCA scenario

3.4 The Diagnostic Knowledge

The diagnostic knowledge base consists of various types of knowledge to simulate the four types of the diagnostic strategies. Figure 4 shows a diagram depicting the diagnostic reasoning flow in the model.

Data-driven frames represent the decision-tables of the operator's heuristics about the relation between a piece of observed data and a hypothesis. Symptomatic rules compare a number of remembered symptom patterns with observed symptom patterns and generate possible hypotheses.

To test hypotheses and to search the failure field of the system topographically, the diagnostic model incorporates three types of knowledge; i.e., the structural, the functional and the behavioral knowledge.

The structural knowledge includes the physical relationships among the parts of the system, commonly called connectivity, and the manner in which the individual parts of the system are constructed. Figure 5 shows a conceptual structural model for the ISLOCA scenario. This model is constructed based on the concept of operator's

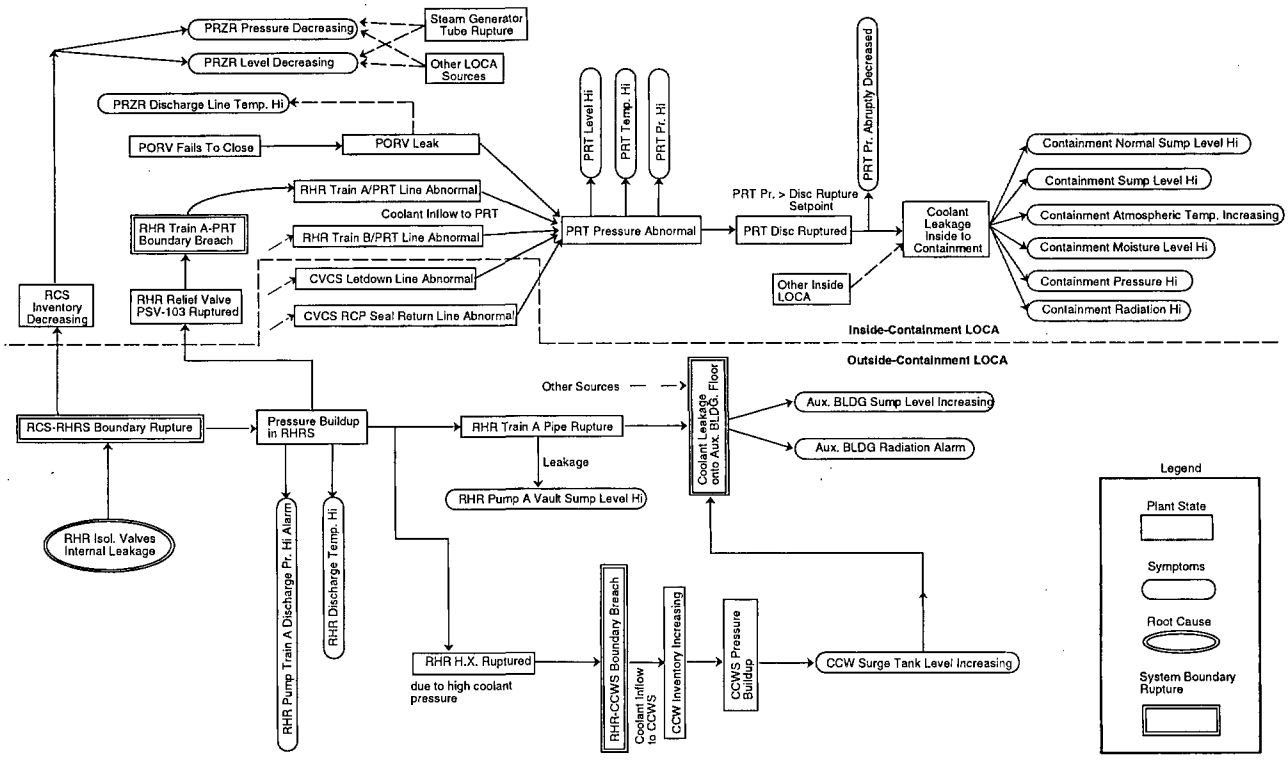


Figure 6: A cause-consequence diagram for the ISLOCA scenario

hierarchical abstraction for the domain system. Each element in the structural model has its attributes such as connectivity information (e.g., peripheral components around its flow path or other interfacing systems) and hierarchical information (e.g., subsystem or components).

The functional knowledge is related to the idea of the intended structure, i.e., a conceptual structure relating a goal to the means to reach it. Failure of a component corresponds to its malfunction. For example, one of the required functions to accomplish the goal 'maintain_mass_balance' in the reactor coolant system is 'pressure_control', where the function is associated with the pressurizer system in the structural model. When a power operated relief valve (PORV) in this system was leaked, the function of 'pressure_control' is malfunctioned.

The behavioral knowledge represents the knowledge of how the parts of the system behave, which is represented as production rules, based on the cause-consequence relationships in Fig. 6. This knowledge describes qualitatively the causal relationships holding between physical variables, component operating states, operating modes, expected functions, and symptoms.

4 Prototype Simulation

Figure 7 shows the prototype run-mode display of SACOM. The system can diagnose a plant malfunction using the diagnostic model, and can simulate a sequence of the operator's emergency tasks that are described in the EOPs.

The inferential status on the blackboard (e.g., data, symptomatic hypotheses, agenda, major symptoms, and structure related hypotheses) can be monitored during the simulation.

In the simulation of the ISLOCA scenario, the diagnostic model shows the following results.

1. At the early stage, the system first diagnoses the possible accident as an inside-containment LOCA by using the symptomatic search strategy.

2. As several symptoms occur, indicating malfunctioning of outside-containment systems, the system tries to identify the root cause of the scenario. A number of hypotheses are generated and tested by using further symptoms and the structural knowledge. The system narrows down the possible failure fields in question, and finally diagnoses the failed component as RHR isolation valves.

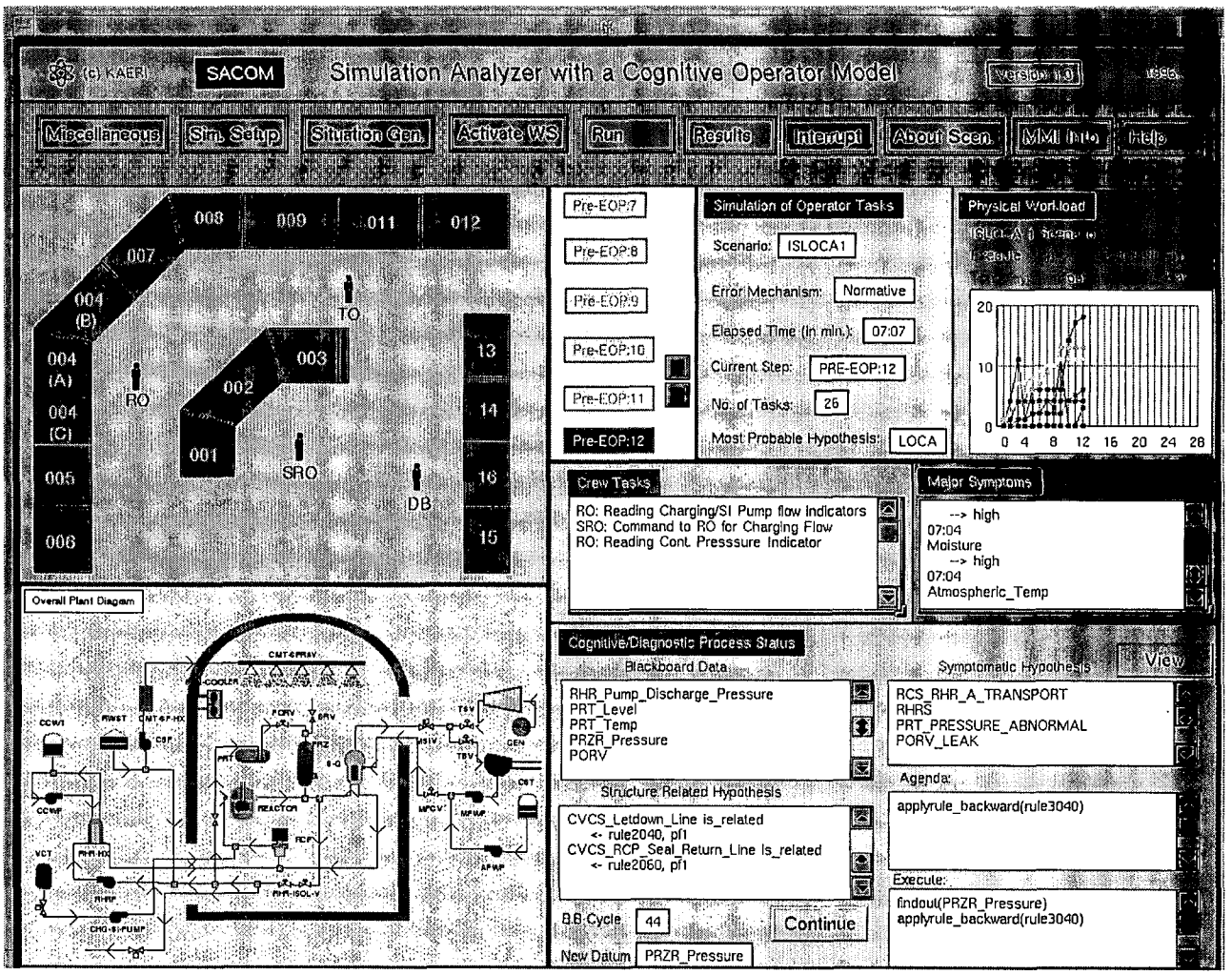


Figure 7: The prototype display of SACOM

5 Conclusions and Further Works

This paper has presented the current status of the work in developing the diagnostic model in SACOM. The system can assess operator's cognitive workload by simulating the diagnostic behavior of operators using the diagnostic model. The blackboard structure in the model is powerful to implement several diagnostic strategies.

The model has employed four types of the diagnostic strategies: i.e., data-driven search, symptomatic search, hypothesis-driven search, and topographic search. The inferential routines of those strategies were implemented on the diagnostic agents.

For further works, it will be required to expand the scope of the diagnostic domain to simulate a number of different scenarios. In this way, the system will be capable of representing various characteristics of operator's diagnostic behavior.

At the current stage, the selection criteria for using the diagnostic strategies are simply dependent on the context of a domain scenario, i.e., observations and the knowledge base. For better assessment of operator's cognitive workload, it is important to derive the selection criteria from the interviews with operators. Also, the diagnostic model should have a function to activate one diagnostic strategy while another strategy is being used in the diagnostic process.

Acknowledgements

We would like to acknowledge the financial support of the Korea Ministry of Science and Technology (MOST) for this work.

References

[1] Cacciabue, P. C., Decortis, F., Drozdowicz, B., Masson, M. & Nordvik, J-P. (1992) COSIMO: a Cognitive Simulation Model of Human Decision Making and Behavior in Accident Management of Complex Plants. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 5, p. 1058-74.

[2] Cheon, S. W., Sur, S. M., Lee, Y. H., Lee, J. W., Park, Y. T. & Lee, K. R. (1995) Development of a Cognitive Task Simulation Analyzer of Operators in Nuclear Power Plants Using Blackboard Techniques. *Proceedings of the 1995 Pacific-Asian Conference on Expert Systems*, Huangshan, China.

[3] Cheon, S. W., Sur, S. M., Park, Y. T., Lee, J. W. & Sim, B. S. (1997) Simulation of Operator's Diagnostic Strategies Under Emergency Operation of Nuclear Power Plants. *Proceedings of the Joint 1997 Pacific Asian Conference on Expert Systems/Singapore International Conference on Intelligent Systems*, Singapore, p. 106-13.

[4] Corker, K.M. & Smith, B. R. (1993) An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation. *Proceedings of the AIAA Computing in Aerospace Ninth Conference*, San Diego, California.

[5] Fujita, Y., Yanagisawa, I., Nakata, K., Itoh, J., Yamane, N., Kubota, R. & Tani, M. (1993) Modeling Operator with Task Analysis in Mind. *Proceedings of the Topical Meeting on Nuclear Plant Instrumentation, Control, and Man-Machine Interface Technologies*, Oak Ridge, Tennessee, USA., p. 505-12.

[6] Furuta, K. & Kondo, S. (1993) An Approach to Assessment of Plant Man-machine Systems by Computer Simulation of an Operator's Cognitive Behavior. *Int'l. J. of Man-Machine Studies*, 39, p. 473-93.

[7] GENSYM (1992) G2 Reference Manual, Ver. 3.0. *Gensym Co., Inc.*, Cambridge, MA, USA.

[8] Hayes-Roth, B. (1985) A Blackboard Architecture for Control. *Artificial Intelligence*, 26, p. 251-321.

[9] Hollnagel, E. & Cacciabue, P. C. (1993) Investigating the Consequences of Incorrect Procedures Execution in Aerospace: the System Response Generator. *Proceedings of the Fourth International Conference on Human-machine Interaction and Artificial Intelligence in Aerospace*, Toulouse.

[10] Meister, D & Hogg, D. (1995) HWR-379: Development of a Task Descriptive Model of the Operator's Fault Diagnosis: a Framework for Interpreting Empirical Data within the Human Error Project. *OECD HALDEN Reactor Project Report.*

[11] Park, Y. T. & Wilkins, D. C. (1992) Explicit Control of Reasoning in a Generic Task. *Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence*, Seoul.

[12] Rasmussen, J. (1986) Information Processing and Human-machine Interaction: an Approach to Cognitive Engineering. *Elsevier Science Publishing Co., Inc.*, Amsterdam.

[13] Rasmussen, J. (1991) Technical Report Risø-M-2952: Diagnostic Reasoning in Action. *RisøNational Laboratory*, Roskilde, Denmark.

[14] Roth, E. M., Woods, D. D. & Pople, H. E. (1992) Cognitive Simulation As a Tool for Cognitive Task Analysis. *Ergonomics*, 35, 10, p.1163-98.

[15] Schryver, J. C. & Palko, L. E. (1988) Knowledge Enhanced Network Simulation Modelling of the Nuclear Power Plant Operator. *Proceedings of Society for Computer Simulation 1988 Multiconference*, San Diego, CA, USA.

[16] WESTINGHOUSE (1987). Emergency Response Guidelines, HP-Rev.1A. *Westinghouse Owner's Group*, USA.

[17] Woods, D. D., Roth, E. M. & Pople, H. (1987) NUREG/CR-4862: An Artificial Intelligence Based Cognitive Environment Simulation for Human Performance Assessment. *U.S. Nuclear Regulatory Commission*, Washington D. C., USA.

Modelling Human Cognition in Problem Diagnosis: A Hybrid Case-Based Reasoning Methodology

Yuh Foong David Law
National University of Singapore
Singapore 119260
Phone: (65)-7726875, Fax: (65)-7791117
E-mail: mcblawyf@leonis.nus.sg

Keywords: hybrid case-based reasoning, cognitive processes, fault diagnosis, help desk

Edited by: Se Woo Cheon
Received: May 5, 1997 **Revised:** May 12, 1997 **Accepted:** June 6, 1997

As modern business functions become more knowledge-intensive, the demand for intelligent and knowledge-based systems has increased. Part of the challenge of developing quality expert systems lies in the effective modelling of cognitive processes of human experts and representation of various forms of related knowledge in a domain. This paper describes a cognitive-driven methodology for the development of expert systems based on a hybrid reasoning approach. The methodology incorporates a hybrid case-based reasoning (CBR) framework of techniques which include case memory organisation networks, case indexing and retrieval schemes; and an interactive and incremental style of reasoning. The paper will discuss the design and implementation of an expert system in a problem response help desk environment of a local bank, as an example of a successful application of this methodology.

1 Introduction and General Motivation

Construction of a knowledge base has often proved to be an intractable task for conventional expert systems. This problem arises because much of the knowledge such as domain knowledge, expertise, heuristics and experience are implicit. A knowledge engineer, usually a novice in the domain, must obtain additional knowledge through self-study and observation, a process that takes inordinate amount of time and effort. Hence, the process of incorporating new knowledge into an existing knowledge base may be difficult. One of the challenges of developing quality expert systems lies in the effective modelling of cognitive processes of human experts and representation of various forms of related knowledge in a domain.

Case-based reasoning (CBR) is a method of solving a current problem by analysing the solutions to previous, similar problems. This technique is based upon our understanding of the nature of human information processing system in some problem areas; human experts depend heavily upon their memory of past experience when solving problems. This is particularly true in areas like law, medical diagnosis and strategic planning. In conventional intelligent systems, knowledge is usually represented in the form of rules. CBR technology promises something different. It stores knowledge in the form of cases, which seems to be a more natural way in problem resolution as it better maps the mental model of domain experts. This is also advantageous as much of the knowledge required already existed as past cases or instances of experience.

Indeed, CBR approach presents a viable way to emulate the thought process of experts in solving problems. Moreover, knowledge can be better acquired and represented, and the effort required for maintaining and managing case bases is also relatively easier compared to rule-based systems (Vargas and Raj 1993). Hence, CBR is appropriate for real world knowledge-based systems because it addresses many of the shortcomings of standard rule-based and frame-based systems.

The rest of this paper describes a cognitive-driven methodology for the development of expert systems based on a hybrid reasoning and knowledge representation approach. The methodology incorporates a hybrid CBR framework of techniques which include case memory organisation networks (discrimination networks and shared-featured networks), case indexing and retrieval schemes (nearest-neighbour similarity matching and knowledge-guided indexing); and an interactive and incremental style of reasoning.

In particular, the methodology hopes to model the cognitive processes and knowledge of a human expert, as discussed in Bolger (1995), which include: cognitive skills (eg. problem-solving strategies), personality and affective traits (eg. relative confidence in judgement), "declarative" domain knowledge (eg. facts, rules), and "procedural" domain knowledge (eg. heuristics). The paper will discuss the design and implementation of an expert system, the Intelligent Help desk Facilitator (IHDF), in a computer problem response help desk environment of a local bank. This system is an example of a successful application of this hybrid methodology.

2 Overview of Application Domain

2.1 Help Desk Operations

The application domain for our system is centred around a computer hotline help desk service operated manually by a small team of technical and help desk operators at the computer problem response centre of a local bank. This hotline team manning the help desk deals with the coordination of a large number of request for service and information. In-house computer end-users of the bank from all branches in Singapore will call a hotline number whenever they encounter hardware, software or network related problems.

The hotline expert will attempt to identify and advise on the problem across the phone. If this is unsuccessful due to unavailability of information or inadequate information provided by the user, the hotline technical expert will need to visit the end-user personally, either at another department or at an off-site branch to inspect the fault. He will try his best to resolve the fault, but if unsuccessful, he will need to carry out tedious fault isolation and troubleshooting procedures. If the fault cannot be rectified within a short period of time, he will just have to continue troubleshooting until it is resolved.

2.2 Challenges in Computer and Network Troubleshooting

The hotline team used to encounter a host of problems and limitations in performing fault isolation and diagnosis tasks manually. To begin with, hotline problem response operations were usually urgent and mission-critical to the day-to-day operations of the bank. The hotline team comprised only two computer and network experts and one operator serving more than 60 remote (off-site) branches and departments of the bank in Singapore.

The process of fault isolation was tedious and time consuming. In the case of a network fault, it could take days to isolate in the complex network environment of the bank. Fault coverage was large and undefined, which made the job of a troubleshooter even more daunting. A simplified network diagram is shown in figure 1 which illustrates the complexity of the connections between a branch and the HQ of the bank. A network problem could arise from a fault in any communications device and equipment, or the connectivity of cables and lines between any two devices, or a combination of both. In most scenarios, the problem symptoms reported were rather general. Different types of computer or network faults could have similar symptom descriptions which further complicated the problem resolution process.

There were also no proper call or fault logging, status monitoring and reporting procedures. In addition, staff turnover would result in the loss of valuable expertise and experience. The process of training a new staff could take at least several months.

2.3 Appropriateness of CBR to Help Desk Environment

CBR technology is increasingly popular in commercial domains (Harmon and Hall 1993), especially in applications where efficient information processing and knowledge retrieval needs are urgent. A help desk deals with the coordination

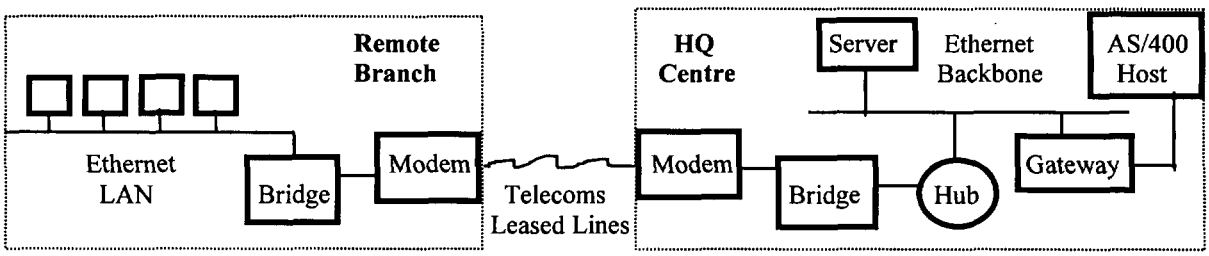


Figure 1: Network connections between a remote branch and the HQ of the bank

of a large number of request for service or information. The help desk operators need to diagnose the problem so that they can dispatch appropriate help. CBR systems could help streamline help desk support operations and improve customer satisfaction. In case-based fault diagnosis, the goal is to recognise faults by using diagnostic cases containing the experience of previous fault detection (Deters 1995). CBR technology will be suitable here where troubleshooting and fault resolution of hardware and network are critical operations, which are usually ill-defined but experience-rich domains. Furthermore, case-based systems can facilitate the process of knowledge acquisition by easing the tasks of case base updates and maintenance.

In most situations, the domain experts will be responsible for the maintenance of the case base through new knowledge input and refinement. They are accustomed to and proficient at considering cases and communicating about expertise through the consideration of cases (Webb 1996). This paper will illustrate the relevance of CBR to problem response help desk, and the effectiveness of adopting a hybrid methodology for expert systems development especially for diagnostic types of applications.

3 Knowledge Representation and Retrieval

IHDF has been developed using a CBR development tool *CBR Express*. The tool offers various useful features for developing CBR applications which include nearest-neighbour matching, confirmatory and discriminatory scoring of questions, facilities for construction of simple rules, and interactive reasoning capability. However the real challenge here is the ability to model the problem domain at hand and to implement an intelligent reasoning strategy which closely resembles the cognitive models of humans in problem resolution. The system should also be useful to users with different levels of experience and expertise. The conceptual design and implmentation of the system will be discussed in the rest of this paper.

3.1 Organisational Structure of the Case Library

Before the case library is implemented, all cases collected through knowledge acquisition process have been sorted out and categorised. This is important as it will affect the efficiency and effectiveness of the case retrieval process. It is advantageous to group related cases into categories to enable the system to discriminate between similar manifestations of ambiguous problems rather than make a gross-level diagnosis (Small and Yoshimoto 1995). If the case base ever reaches a size where run-time performance is a problem, it can be partitioned according to the various categories to reduce the search space for similarity matching. In order to facilitate speedy and accurate case retrieval from the expanding case base library, cases could be conceptually partitioned based on a combination of *discrimination* and *shared-featured* memory organisation network structures (Kolodner 1993).

In IHDF, cases are physically stored in a sequential manner in memory but are indexed logically according to these two memory organisation structures, as illustrated in figures 2a and 2b below.

Discrimination network structure provides an

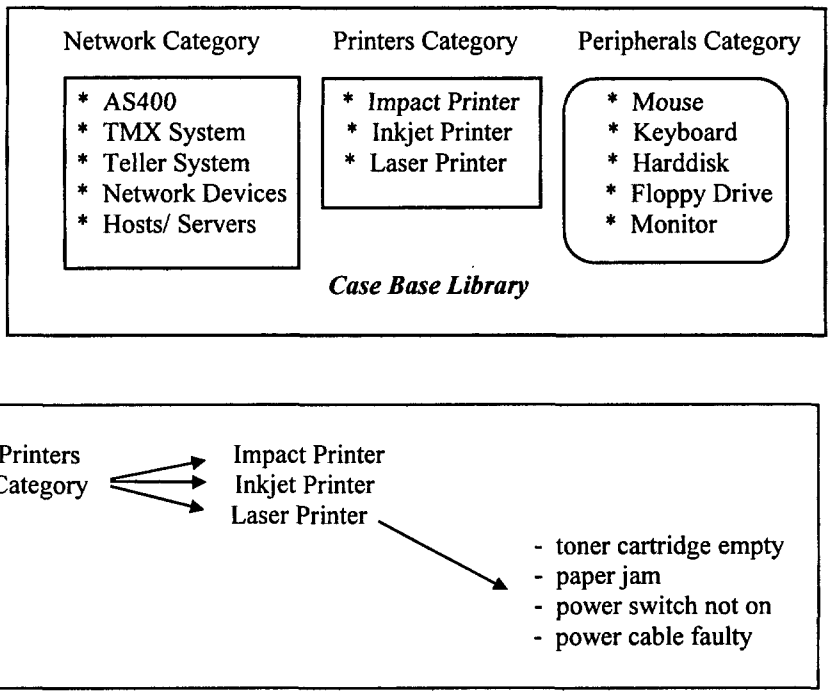


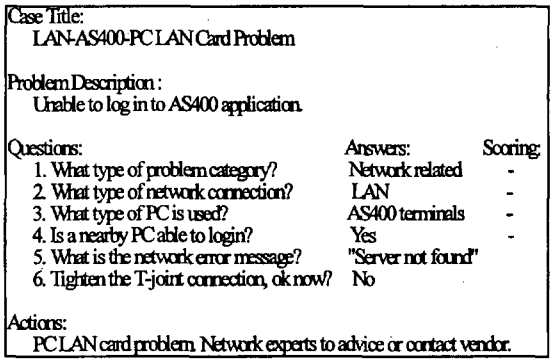
Figure 2: Main categories of cases in the case base (a), Conceptual case hierarchy (b)

effective first cut into the case base. In figure 2a, discrimination network structure is used to distinguish among the main categories of cases. The relevant category of cases will be retrieved and the other categories will be discriminated against. Within each category, there are similar or related groups of cases (figure 2b), mostly sharing some common attributes. These are organised using a combination of discrimination and share-featured network structures. Once a category of cases is isolated, the system can further shrink the cluster based on different ranking and scoring of individual cases to select one or a few optimal case solutions.

Cases in different categories are indexed and grouped differently, depending on the nature of the problem. For instance, the Network category will contain cases which are network-related and they usually have very similar problem descriptions. These cases will be grouped according to common problem scenarios or fault symptoms. Cases in other categories are usually hardware dependent, and are therefore more appropriately grouped according to specifc equipment. Discrimination network structures can be implemented using discriminatory scoring of questions.

3.2 Knowledge Representation Structures

Domain knowledge can be represented as cases and rules. Cases can represent instances and episodic types of knowledge while rules are useful for "declarative" and "procedural" types of knowledge. A case adopts a frame-based representation comprising of four main parts (figure 3a): *case title* indicates the category which it belongs to; *problem description* contains fault symptoms as reported by a caller, including key words used to facilitate fuzzy text search and indexing; *case attributes* are the main indexing elements which take the form of questionnaires posed to the user; and *action* will contain a set of documented expert advice and recommendations. Rules are used to facilitate knowledge-guided reasoning via automatic answering of questions, based on a deductive domain model. A rule (figure 3b) could have one or a conjunct of premises and conclusions. A premise of a rule is actually implemented as the number label of a question (attribute) with its corresponding answer (value). Similarly, the conclusion contains other questions with deduced answers.


Case Title:
LAN-AS400-PC LAN Card Problem

Problem Description:
Unable to log in to AS400 application.

Questions:	Answers:	Scoring:
1. What type of problem category?	Network related	-
2. What type of network connection?	LAN	-
3. What type of PC is used?	AS400 terminals	-
4. Is a nearby PC able to login?	Yes	-
5. What is the network error message?	"Server not found"	
6. Tighten the T-joint connection, ok now?	No	

Actions:
PC LAN card problem. Network experts to advice or contact vendor.

Question labels	*Answers to expect or deduce*
2	LAN
0	AS400
0	log on
===⟩	
3	AS400 terminals

Figure 3: Example of a case (a), rule (b)

3.3 Case-Base Indexing and Retrieval Schemes

Two indexing and retrieval approaches are offered by CBR Express: *nearest-neighbour* and *knowledge-guided indexing.* The nearest-neighbour algorithm (Kolodner 1993) is able to perform: partial case matching, using a numeric evaluation function (figure 4); and case ranking based on degree of similarity match. In the numeric evaluation function, a set of weightages (w) is assigned to attribute dimensions (f) of a case, which reflects the relative importance of each attribute as indicated by the domain expert. Every attribute in the given case (f_i^I) is matched to its corresponding attribute in the retrieved case (f_i^R), the degree of match for each pair is computed using the function $sim()$. Based on weightage assigned to each attribute, an aggregate match score is computed which represents the aggregate degree of match of the retrieved case to the given one. Next, cases are ranked according to their scores. Cases with higher matching scores indicating higher degrees of similarity will be ranked higher, and hence more likely to be retrieved. The use of attribute weightage, score computation and ranking are effective means of modelling a human expert's biases and confidence in judgement.

One of the main benefits of nearest-neighbour matching is its ability to deal with incomplete information input, and will always be able to find an optimal solution even though an exact-matched case may not necessarily be available. However, one limitation of this indexing scheme is the retrieval of cases without the use of domain knowledge. This makes it susceptible to incorrect or out-of-context case retrieval. In knowledge-guided indexing, domain knowledge in the form of rules, are used to index cases directly.

This is useful and it complements nearest-neighbour indexing. The hybrid indexing approach has significantly improved the speed and accuracy of case retrieval. This synergy of these two techniques will be further discussed in section 4.

4 System Implementation

4.1 Hybrid Reasoning Strategy

In diagnostic types of applications, assessments based only upon rule-based knowledge are subjected to incompleteness (eg. a rule base may not have included the exceptions or the less well defined knowledge), while assessments made only from case-based knowledge may be less systematic (eg. some important rules or algorithms may not be used in existing cases) according to Xu (1995). A CBR system, basically an inductive reasoning approach, draws inferences of a new case by comparing it with cases in the case library. On the other hand, a rule-based reasoning (RBR) system, based on deductive reasoning, usually employs a rich domain theory which can guide the problem solver to reach a given goal (Chi and Kiang 1993). Ketler (1993) summarised various advantages of CBR over RBR. These include the close resemblance of CBR to human decision processes, the automation of the process of incorporating new knowledge into an existing knowledge base, the rapid creation of a case base and retrieval of cases, the ability to provide better explanation and justification, and the use of CBR in problems with

$$\frac{\sum_{i=1}^{n} w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i}$$

where w_i is the importance of attribute dimension (slot) i,
Sim() is the similarity function for attributes,
f_i^I and f_i^R are the values for attribute f_i in the given and retrieved cases, respectively.

Figure 4: Numeric evaluation function for case similarity assessment

poorly understood domains.

However in practice, most problem solving processes require both types of reasoning paradigms while previous research tends to emphasise either one of the approaches. This observation encourages the combinations of these two approaches to form an integrated system which has the strength of both systems and can be applied in a knowledge-poor and experience-poor problem domain. Domain knowledge (both empirical and theoretical knowledge) are important and are essential to the reasoning process (Bichindaritz 1994). Cases can be used to represent empirical knowledge while rules can be used to represent both theoretical and some empirical knowledge. Hence, if rule-based and case-based systems are integrated into a single hybrid system, it could have the potential to use all available information to achieve better results and performance. In IHDF, a hybrid reasoning architecture has been implemented using the indexing schemes described earlier. This is further illustrated in figure 6.

4.2 Expert System Architecture

The expert system consists of two main modules: the Consultation module and the Maintenance module. The system represents an integration of the case library designs, retrieval and indexing techniques, and other considerations discussed earlier for computer and network fault diagnosis.

The hotline operator (a novice troubleshooter) will consult the system by entering free-text description of a problem and answering a few questions prompted by the system. A set of optimal solutions (most probable cases) will be retrieved, fully documented with experts' comments and advice. There is also a hypertext component which will provide on-line help and documentation facilities. Documented domain knowledge, explanation of experts' heuristics and actions will serve as a knowledge base for tutoring and training new inexperienced staff in fault resolution, with minimal human expert supervision.

4.3 Modelling the Reasoning Process

A human expert usually tackles a problem from a top-down manner, in a systematic and incremental manner, relying on intermediate results at different stages of the diagnostic process to narrow down the search space. The expert system maintains a highly interactive mode with its user, employing a step-by-step incremental reasoning approach. The computer-assisted problem resolution process begins with the entry of problem symptoms and descriptions reported. This is followed by the retrieval of relevant cases with questions displayed to the user to be answered one-by-one, based on caller feedback over the phone. Whenever available, knowledge-guiding rules will be fired to automatically answer further questions. Finally the system will narrow down to the closest matched cases (optimal solution). Appropriate expert advice documented in the retrieved cases will be available. If the problem situation is new (not available in the case library), it will be escalated to the human expert for further analysis and advice. The case base administrator (usually the human experts) will update new cases into the case library.

There are several benefits of adopting an interactive and incremental reasoning approach. Firstly, different cases may have different questions (case attributes) and hence at different states of the reasoning process (figures 6 and 7) cases will have different score rankings which will determine the order of questions displayed to the user. Secondly, the user would not be able to know upfront which are the right questions to ask, and would therefore have to rely on the system to suggest probable questions at different reasoning states. Thirdly, the caller may not always be able

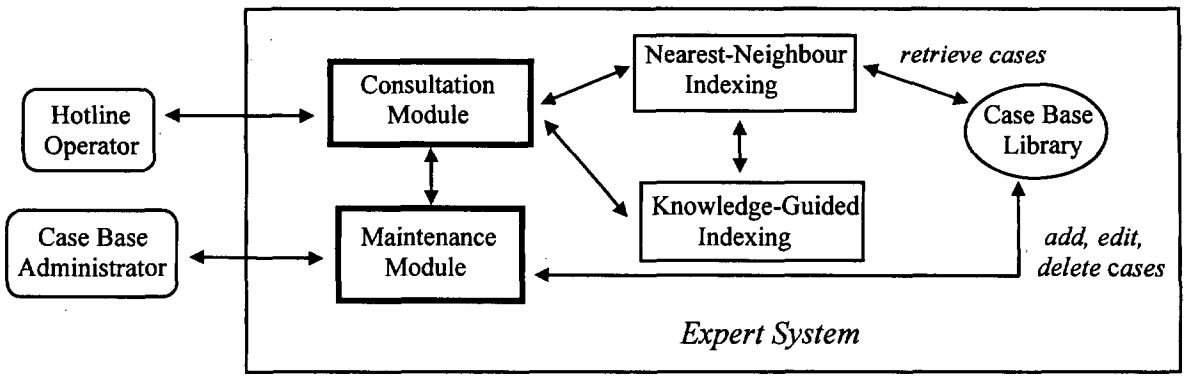


Figure 5: System architecture of the IHDF system for hotline operations management

to provide immediate answers to all questions suggested by the system. This is so as some questions may require the caller to check and confirm fault symptoms. Accurate answers will be feedback to the system user over the phone before the system can proceed to further reasoning states. Fourthly, the system is able to allow the user to back-track to previous reasoning states should he decide to unanswer a question due to erroneous input from the caller.

A human expert will tend to approach a problem systematically with a intuitive problem-solving scheme, and proceed to performing fault isolation by elimination strategy. To begin a case-based consultation process (state 0), the system user will enter some free-text problem symptoms as described by the caller over the phone. This serves as the first-cut into the case base memory (state 1), where cases with similar problem description will be retrieved while others will be discriminated against.

The system performs free-text matching based on a character-matching technique called *Trigrams* (matching every 3-letter sequence in the text). It is effective and robust, tolerant to minor misspellings, though computationally intensive (Small and Yoshimoto 1995). The system then conducts a question-and-answer session with the user, similar to a human expert asking relevant questions (with feedbacks from the caller) to help narrow down to the relevant cluster of cases (from state 1 to n). As questions are answered, the system recomputes the closest matched cases and continuously displays the fresh matching scores to the user (Yoon et al. 1993) as shown in figure 7. These matched cases represent the system's current best guess on the classification of the problem statement.

The user will continue with the question-and-answer session until he feels the closest matched case retrieved represents the current problem (optimal solution found). He will then examine the actions and recommendations associated with the closest matched case to determine how to resolve the current problem. Sometimes the user may detect missing knowledge and experience as evidenced by the retrieval of optimal case solutions with low score rankings (Klahr and Vrooman 1991). In such circumstances, these new situations will be refined and incorporated into the case base library as new cases by the case base administrator.

Figure 7 shows the usefulness of rules (automatic answering of questions) where the number of reasoning states can be effectively reduced to four. In this example, the system is able to answer the first few questions based on certain keywords present in the problem description. Answers to certain questions may also be used to trigger off answers to further questions. In this way, rules empower the system with a certain degree of domain understanding as they enable the system to reason within the confines of a domain model, serving a knowledge-guiding function. Not only does this accelerates the case retrieval process, it also enhances the accuracy of cases retrieved, with minimal human input errors. The hybrid CBR and RBR architecture reflects how a human expert could combine his knowledge from previous experience and his heuristics of expertise to effec-

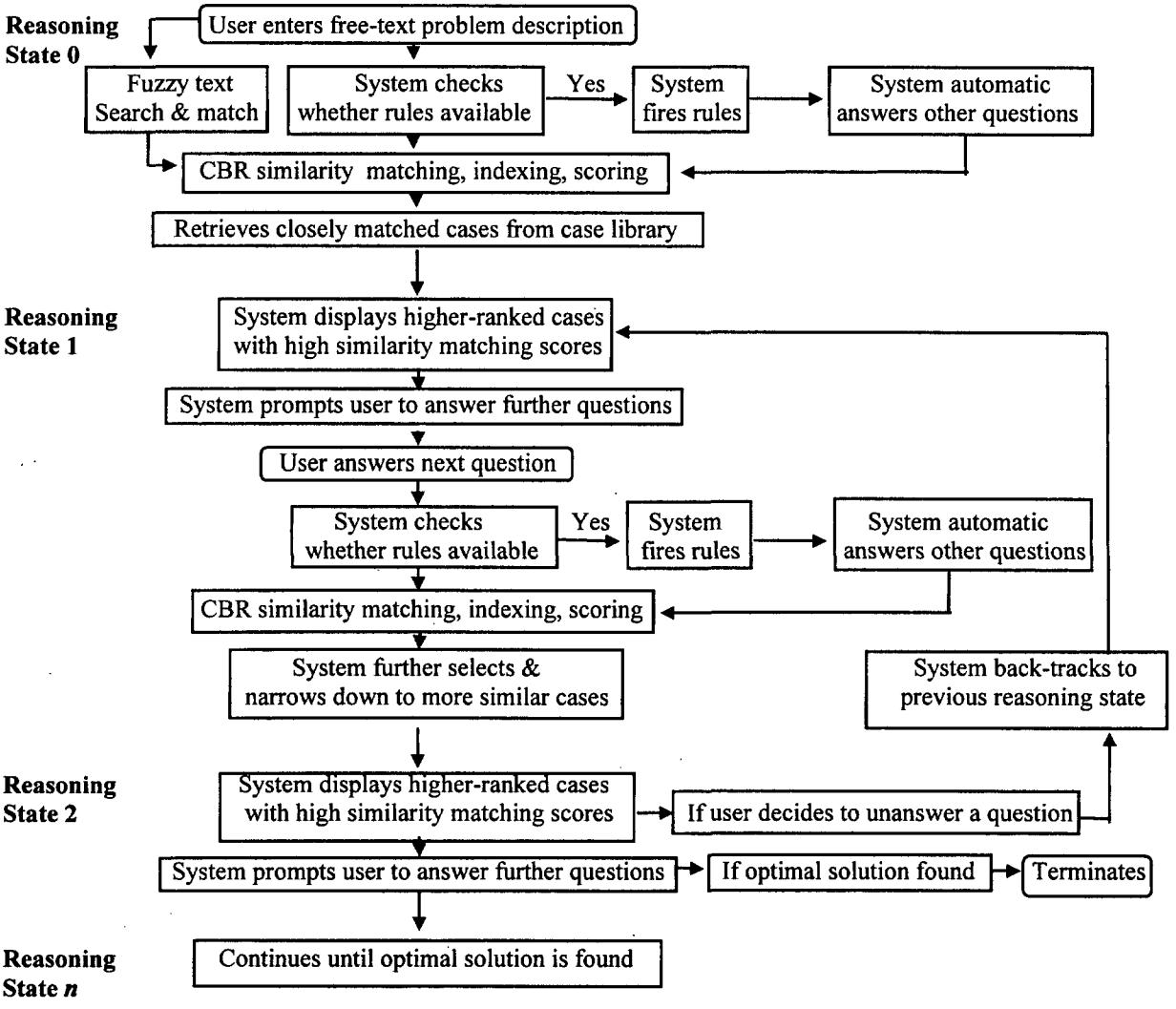


Figure 6: An interactive and incremental hybrid CBR reasoning process

tively tackle a new problem situation.

5 Discussion And Conclusion

The CBR expert system was developed to augment the problem response capability of help desk staff, and has been successfully deployed in a hotline response centre. The system is based on a cognitive hybrid CBR framework which demonstrates the synergy of various reasoning techniques and methodologies to organise, represent and reason about knowledge. Such an integrated, hybrid methodology is effective in solving real world problems and closely reflects the cognitive model of a human expert in problem resolution.

A post-implementation assessment revealed several interesting issues and benefits of the system. Manual fault tracking and monitoring has been reduced. The hotline operator, with little experience in troubleshooting, was able to use the system to perform fault isolation of common problems, most of which were previously handled only by the human experts. Timely, quality and consistent response could be provided to callers. The experts themselves have relatively more time to plan and manage the hotline operation. Also, the knowledge documented in the system has facilitated in-house training of new staff. On one occasion, one of the two hotline experts left the team. The system, which has previously retained some of his expertise, was able to minimise temporary disruptions to the hotline workflow until a new staff was recruited. Valuable knowledge captured in the system was effectively exploited to allevi-

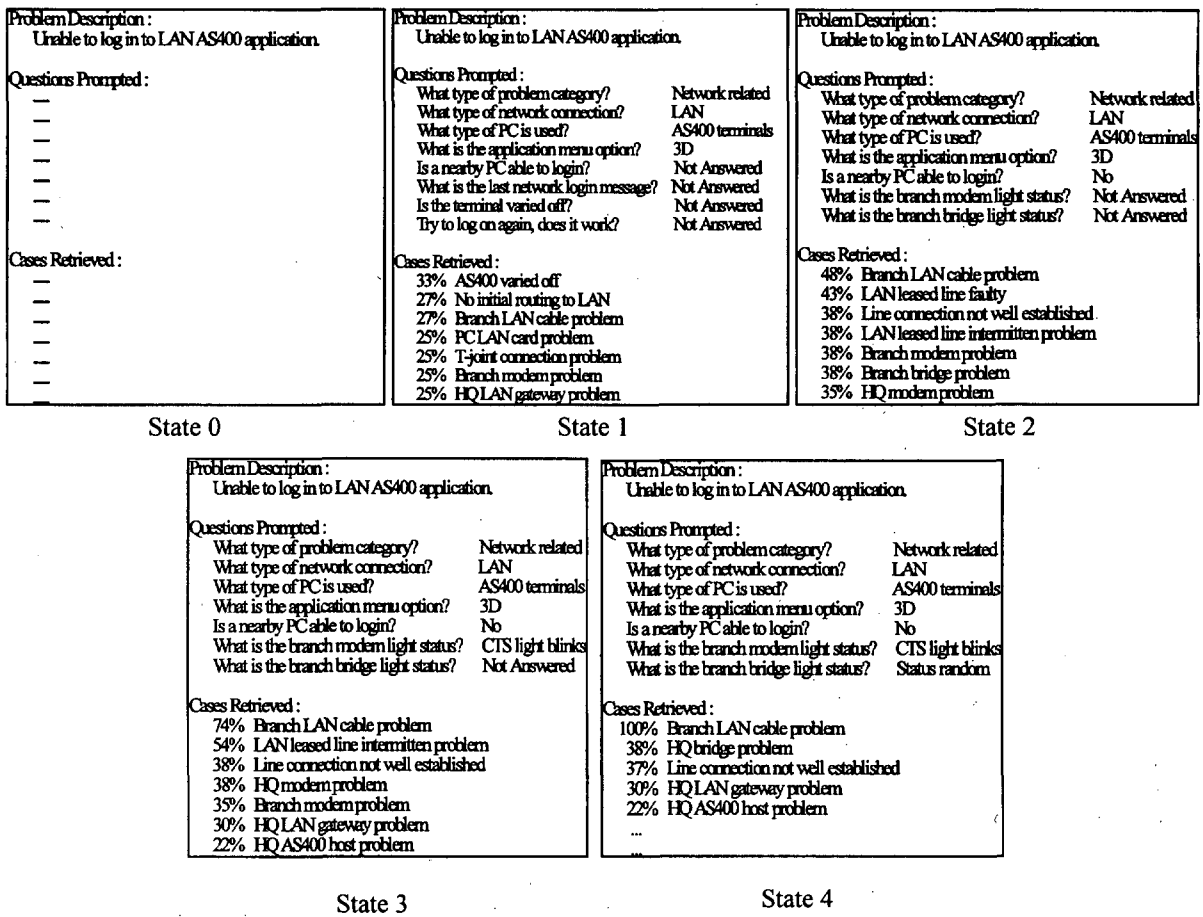


Figure 7: A computer network fault isolation situation which illustrates the intermediate reasoning states

ate possible bottlenecks in hotline response due to the lack of expertise.

In essence, the expert system has helped to streamline the management of hotline operations and enabled the team to function at a high state of operational readiness in response to mission-critical problem situations. We present the cognitive-driven CBR methodology as an effective paradigm for the development of quality expert systems.

References

[1] Bichindaritz, I. (1994) A Case-Based Reasoning System Using a Control Case-Base, *11th European Conference on Artificial Intelligence (ECAI-94)*, Amsterdam, pp.38-42.

[2] Bolger, F. (1995) Cognitive Expertise Research and Knowledge Engineering, *The Knowledge Engineering Review*, vol.10, no.1, pp.3-19.

[3] Chi, R.T.H and Kiang, M.Y. (1993) Reasoning by Coordination: An Integration of Case-Based and Rule-Based Reasoning Systems, *Knowledge-Based Systems*, vol.6, no.2, pp.103-113.

[4] Deters, R. (1995) Case-Based Diagnosis of Multiple Faults, *Proceedings of the 1st Inter-*

national Conference on Case-Based Reasoning (ICCBR-95), Portugal, pp.411-420.

[5] Harmon, P. and Hall, C. (1993) *Intelligent Software Systems Development: An IS Manager's Guide*, Wiley.

[6] Ketler, K. (1993) Case-Based Reasoning: An Introduction, *Expert Systems with Applications*, vol.6, no.1, pp.3-8.

[7] Klahr, P. and Vrooman, G. (1991) Commercializing Case-Based Reasoning Technology, *Research and Development in Expert Systems VIII: Proceedings of Expert Systems 91, the 11th Annual Technical Conference of the BCS Specialist Group on Expert Systems*, London, pp.18-24.

[8] Kolodner, J. (1993) Case-Based Reasoning, Morgan Kaufmam.

[9] Small, R.A. and Yoshimoto, B. (1995) The VLS Tech-Assist Expert System, *AI Magazine*, vol.16, no.1, pp.41-50.

[10] Vargas, J.E. and Raj, S. (1993) Developing Maintainable Expert Systems using Case-Based Reasoning, *Expert Systems*, vol.1, no.4, pp.219-226.

[11] Webb, G.I. (1996) Integrating Machine Learning with Knowledge Acquisition through Direct Interaction with Domain Expert, *Knowledge-Based Systems*,vol.9, no.4, pp.253-266.

[12] Xu, L.D. (1995) Case-Based Reasoning for AIDS Initial Assessment, *Knowledge-Based Systems*, vol.8, no.1, pp.32-38.

[13] Yoon, Y., Acree, A.D and Peterson, L.L. (1993) Development of a Case-Based Expert System: Application to a Service Coordination Problem, *Expert Systems with Applications*, vol.6, no.1, pp.77-85.

Medical Decision Support System for the Management of Hypertension

Young Moon Chae, Seung Hee Ho, and Mi Young Bae
Department of Health Management Information System, Yonsei University
C.P.O.Box 8044, Seoul Korea,
Phone:+82-2-361-5048, Fax:+82-2-392-7734
E-mail: ymchae@hitel.kol.co.kr
Sang Hyun Lee
Department of Family Medicine, Inha University
AND
Hee Choul Ohrr
Department of Preventive Medicine, Yonsei University

Keywords: hypertension, decision support system, expert system, CAI (computer assisted instruction), neural network, case-based reasoning, discriminant analysis

Edited by: Se Woo Cheon
Received: February 24, 1997 **Revised:** May 12, 1997 **Accepted:** June 6, 1997

This paper describes the development of a medical decision support system (MDSS) for the management of essential hypertension. While most MDSS's have been developed for doctors in university hospitals, this system is developed for primary health care doctors and health workers in community health centers. The MDSS consists of three modules: health management information system (HMIS) for health center, logistic regression model, and an expert system. HMIS maintains hypertension database for two modules. Logistic regression was used to identify risk factors for essential hypertension and three knowledge models were compared to determine the best method for predicting the severity of hypertension and the prognosis of treatment. Out of all the cases, discriminant analysis showed the highest predicitive power. To improve user interface, computer assisted instruction (CAI) capability also incorporated in the system.

1 Introduction

Hypertension is a major contributor to coronary heart disease and the leading cause of death in American adults (Jaffe & Seely, 1995). In Korea, 15% of the adult population in Korea have hypertension (Huh, 1992); and of those with hypertension, 85-90% have essential hypertension while the remaining have secondary hypertension. This study is focused on the management of essential hypertension because it affects the majority of the population. When the patients are suspected of suffering from secondary hypertension, they are usually referred to the tirtiary hospitals for further examination and specialized treatment for the underlying diseases such as primary aldosteronism, Cushing's disease, and pheochromocytoma.

Various factors have been implicated in the pathogenesis of essential hypertension, although the exact cause of essential hypertension is still unknown (Garg & Pepper, 1995). In patients with essential hypertension, heavy cigarette smoking (greater than or equal to 20 cigarettes/day) is associated with a definite increase in left ventricular mass through a rise in whole-day blood pressure (Verdecchia et al., 1995). Since the pathogenesis of essential hypertension is multifactorial, diagnosis as well as the causal therapy of hypertension remain a great challenge.

Considerable variation exists among both general practitioners and physicians in their stated assessment and management of hypertension. Overall, general practitioners and physicians have similar stated thresholds for treating hypertension

but differ in their choice of first line therapy (Ford & Asghar, 1995). This suggests that effective interventions in addition to clinical guidelines need to be developed if hypertension is to be consistently managed.

While there have also been many studies on a MDSS or medical expert system which provides such information to help doctors improve their decision-making, most of them are applicable to hospital use. Since hypertension is the most prevalent disease in the community and needs to be continuously managed, a MDSS was developed in this study mainly for general practitioners and health workers in community health centers to improve the transfer of knowledge from the university hospital to the community health centers. In addition, three knowledge models were compared to determine the best method for predicting the severity of hypertension and the prognosis of treatment in a community setting. CAI capability was also incorporated in the system to improve user interface for health workers.

2 Subjects and Methods

2.1 Subjects

The subjects of the study were classified into two large groups: community resident data and hospital data. The community data was comprised of 171 Kangwha county residents who participated in the Cohort Study(Ohrr, 1993) on hypertension, and the hospital data were composed of 65 patients with hypertension who received treatment at the Yonsei University Severance Hospital from 1986 to 1995. Community data was used in the determination of risk factors for hypertension in a community setting; whereas the hospital data was used in the development of knowledge models for predicting the severity of hypertension and the prognosis of treatment.

2.2 Methods

2.2.1 Structure of the MDSS for Hypertension Management

As shown in Figure 1, the hypertension management information system consists of three modules: health management information system (HMIS), logistic regression model, and expert system. HMIS is an information system for health center that provides administrative information to assist internal management and maintains health database for residents including hypertension data. Logistic regression model provides information on risk factors for hypertension in the community. Finally, expert system provides diagnostic and therapeutic information using three knowledge models.

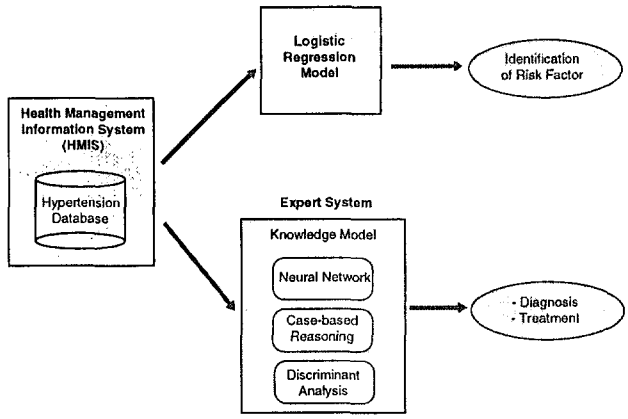


Figure 1: Structure of the MDSS for hypertension management

2.2.2 Logistic Regression Model

Logistic regression model is a statistical method for predicting a dichotomous dependent variable using independent variables (David & Lemeshow, 1989). Logistic regression was performed to identify risk factors of hypertension in community using patient charateristics, history, lifestyle, and test results as independent variables and the hypentension status as dependent variable. The model can be expressed as follows:

$$\log(p/(1-p)) = a + b_1X_1 + b_2X_2 + ... + b_kX_k$$

where $p(1-p)$: odds of development hypertension, and

$\log(p/(1-p))$: log odds of developing hypertension

2.2.3 Knowledge Models

A. Neural Networks

A back-propagation method with sigmoid function was used in this network. In the case of the diagnosis model, the input layer was composed

of 30 input nodes including the patient characteristics, while the output layer was composed of one output node presenting a severity of hypertension which is a dichotomous variable ('hypertension group' for those in stage 1, 2, 3 of hypertension; and 'severe-hypertension group' for those in stage 4). The prognosis of treatment was performed using the same input nodes and one output node presenting treatment efficiency with a hidden layer including 5 nodes similar to the diagnosis model. Prognosis of treatment was divided into two large groups: effective and non-effective. After pharmacologic therapy, the effective group was determined by a decrease in systolic blood pressure below 140 mmHg and diastolic blood pressure below 90 mmHg. This study examined the relative predictive importance of the input nodes by partitioning the sum of effects on the output layer. These are represented by the "shares" which is the percentage of all output weights attributable to the given input nodes (Garson, 1991).

$$\sum_i^{n_h}\left(\frac{Iv_j}{\sum_k^{n_h} Iv_j} O_j\right) / \sum_i^{n_v}\left(\sum_j^{n_h}\left(\frac{Iv_j}{\sum_k^{n_v} Iv_j} O_j\right)\right)$$

For each j of n_h hidden nodes, sum the product formed by multiplying the input-to-hidden connection weight of the input node I of variable v for hidden node j times the connection weight of output node O for hidden node j; then divide by the sum of such quantities for all variables. The result is the percentage of all output weights attributable to the given independent variable and thus represents the relative importance of the independent variable. In short, this process partitions the hidden-to-output connection weights of each hidden node into components associated with each input node shares.

B. Case-based Reasoning

A CBR(case-based reasoning) draws its predicting power from a large case library, and therefore, to be successful, cases should be organized in a manner that only relevant cases be retrieved from memory, and previous cases be effectively adapted to new problem (Garson, 1991). In this study, a case was constructed to represent a hypertension condition with features for describing the symptoms associated with a condition as well as features for treatment. A stored case matches a presented case when it contains similar features. The degree of match depends on the degree of similarity. In this model, the best matching cases were retrieved using the case score based on how closely the features of the presented-case schema match the features of the stored-case.

$$case\ score\ (new, old) =$$
$$2/(2 - normalized\ similarity\ score) - 1$$
where normalized similarity score
$$= (sum\ of\ match\ score + sum\ of\ penalties) / (sum\ of\ max\ weights)$$

Scores ranged from 1.0 for cases that exactly match to -1.0 for cases that completely mismatch. In this study, case scores were obtained based on two alternative match scores: scores based on the doctors' clinical judgement and the "shares" from the neural network. The latter approach is an effective way to integrate the neural network and CBR.

C. Discriminant Analysis

Discriminat analysis is a statistical method for predicting a certain event which is a categorical variable. In this study, discriminant analysis was used to predict a severity of hypertension and the prognosis of its treatment using patient charateristics, history, and test results as independent variables.

3 Results

3.1 Identification of the risk factors of hypertension

In the results, the logistic model was statistically significant (p=0.0036). The significant variables at 5% level were : past history of blood pressure, stop drinking, and WBC(white blood cell). In case that variable is classified with male and female represented the same results. When continuous variables were classified in the regular range, only 'past history of high blood pressure' was significant (Odds Ratio=11.4067, p=0.0071) for female. However, the model was not statistically fitted for male.

Table 1: Odds ratios and parameter estimates for BP(blood pressure) by patient's characteristics

patient's chracteristics	Odds Ratio	Parameter estimate	p value
past history of BP	17.80	2.88	0.00
smoking	0.11	-2.17	0.29
drinking	4.63	1.53	0.16
stop drinking	140.72	4.94	0.03
amounts of smoking	1.00	-0.004	0.18
pulse	1.10	0.09	0.16
waist / hip	0.00	-8.77	0.12
GOT	0.90	-0.11	0.14
GPT	0.93	-0.07	0.28
TR	1.02	0.02	0.17
WBC	1.92	0.65	0.02

(GOT: plutamic oxaloacetic transminase, GPT: plutamic pyrubic trasminase, TR: tricuspid regurgitation)

Table 2: Five most important "share"s for the severity of hypertension

Variable	Share
Pulse rate	3.958
Alcohol	3.928
Glucose	3.873
Symptom - palpitation	3.841
BUN(Blood Urea Nitrogen)	3.824

3.2 Knowledge Models

3.2.1 Neural Networks

Five most important inputs influencing the severity of hypertension for the neural network were presented in Table 2. They were: pulse rate, alcohol, glucose, sysmptom (pappitation), and BUN(blood urea nitrogen). Five most important inputs influencing the prognosis of treatment for the neural network were presented in Table 3. They were: family history, glucose(U/A), symptom (headache), symptom (dyspnea), and EKG(electrocardiogram).

To validate each knowledge model, the predictive value decided by the experimental result of each knowledge model was compared with the actual value decided in the clinic. In predicting the severity of hypertension, the capability of the neural networks model for 15 test samples were: 60% for severe-hypertension, 90% for hypertension, and 80% for the total (Table 4). In predicting the prognosis of treatment, the capability of the neural networks model for 20 test samples were: 50% for effective, 80% for non-effective, and 65% for the total (Table 5).

Table 3: Five most important "share"s for the prognosis of treatment

Variable	Share
Family history	3.425
Glucose(U/A)	3.360
Symptom-headache	3.327
Symptom-dyspnea	3.307
EKG	3.294

Table 4: Comparison of three knowledge models for the severity of hypertention

Actual diagnosis made by doctor		No. of matching cases		
		NN(%)	CBR(%)	DA(%)
Hypertention	10	9 (90)	8 (80)	9 (90)
Severe hypertention	5	3 (60)	2 (40)	5 (100)
Total	15	12 (80)	10 (67)	14 (93)

(NN: neural network, CBR: case-based reasoning, DA: discriminant analysis)

3.2.2 Case-based reasoning

The match score which consisted of matching and mismatching scores in each slot were decided upon by a doctor and the "share" obtained from the neural networks. In comparison with the accuracy of the two methods, the method using the match score by a human expert produced a higher score than by the "share".

In predicting the severity of hypertension, the capability of CBR for the total (67%), for severe-hypertension (40%), and for hypertension (80%) was the lowest among three models (Table 4). Similarly, in predicting the treatment results, the

Table 5: Comparison of three knowledge models for the prognosis of treatment

Actual treatment result		No. of matching cases		
		NN(%)	CBR(%)	DA(%)
Non-effective	10	8 (80)	7 (70)	9 (90)
Effective	10	5 (50)	4 (40)	9 (90)
Total	20	13 (65)	11 (55)	18 (90)

Table 6: Summary of stepwise discriminant anlysis for the severity of hypertension

Step	Discriminant variable	Partial R^2	Wilk's Lambda	p value
1	Alcohol	0.1443	0.856	0.0018
2	Sex	0.0510	0.812	0.0016
3	BUN	0.0727	0.756	0.0006
4	EKG	0.0674	0.705	0.0003
5	Protein	0.0716	0.655	0.0001
6	K	0.0438	0.626	0.0001
7	TG	0.0371	0.603	0.0001

(Wilk's Lambda: 0.60, p=0.0001)

Table 7: Summary of stepwise discriminant analysis for the prognosis of treatment

Step	Discriminant variable	Partial R^2	Wilk's Lambda	p value
1	EKG	0.0868	0.913	0.0172
2	TG	0.0728	0.847	0.0058
3	Complication	0.0489	0.794	0.0027
4	Smoking	0.0469	0.757	0.0020
5	CA	0.0469	0.721	0.0014
6	BMI	0.0441	0.690	0.0011
7	PH	0.0487	0.656	0.0007
8	K	0.0610	0.632	0.0003
9	HCT	0.0437	0.605	0.0003

(Wilk's Lambda: 0.59, p=0.0003)

capability of CBR for the total (55%), for effective(40%), and for non-effective (70%) were the lowest among three models (Table 5).

3.2.3 Discriminant Analysis

Stepwise discriminant analysis was performed for predicting the severity of hypertension and the progsosis of treatment. As seen in Table 6 and Table 7, significant variables were: alcohol, sex, BUN, EKG, Protein, K(kalium), TG(thyroglobulin) for the severity of hypertension; and EKG, TG, complication, smoking, CA(coronary angiography), BMI(body mass image), PH(pulmonary hypertension), K, HCT(hematocrit) for the prognosis of treatment, respectively. In predicting the severity of hypertension with test cases, the capability of the discriminant model for the total (93%), for severe hypertension (100%), and for hypertension (90%) was the highest among three models (Table 4). Similarly, in predicting the treatment results, the capability of the discriminant model for the total (90%), for effective (90%), and for non-effective (90%) was the highest among three models (Table 5).

3.3 Development of CAI-based Expert System

The expert system for hypertension was designed to enhance the availability and efficiency of hypertension management in health centers and health subcenters. To increase educational efficiency, CAI capabilities such as tutorial and Encyclopedia were included in the system. In addition, several medical images such as a chest PA(chest X-ray), coronary angiography, retina, and EKG were also included in the system. A summary screen for the prognosis of treatment is shown in Figure 2.

4 Discussions and Conclusions

Diagnosis and therapeutic planning of hypertension are a multifactorial process involving the assessment of symptoms, severity, personal characteristics, lifestyle, past and family history, laboratory tests, and a physical examination. There have been many studies on identifying the relationship among these various factors to improve effectiveness in the management of hypertension.

Diagnostic capabilities for the three knowledge models varied. The discriminant analysis had the best overall prediction rate (93%) and the 100% prediction rate for severe hypertension; the neural network had equally the best prediction rate for hypertension (90%); and the CBR had the lowest prediction rates in all categories. Similarly, in predicting the prognosis of treatment, discriminant analysis also had the best prediction rate (90%); the neural network had much lower prediction rate (65%); and the CBR again had the lowest prediction rate (55%). These findings were consistent to the study on Allergic Rhinitis (Chae et al.,1992). Therefore, the discriminant model, which has not been widely used in the field of expert systems, should be given more application as a knowledge model and be used as an important reference in the management of hypertension.

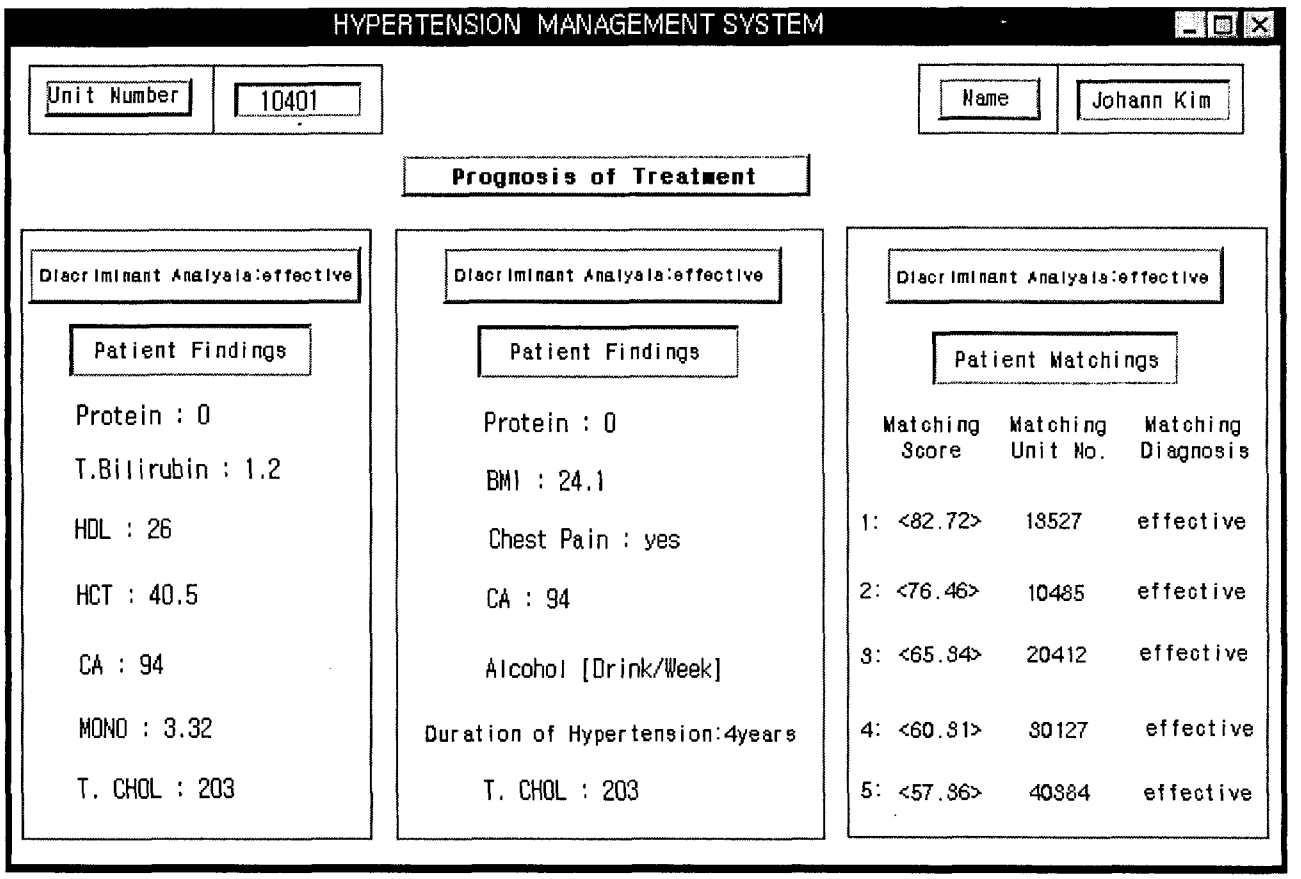


Figure 2: Summary screen for the prognosis of treatment

Overall capabilities for the three models can be improved as more hypertension data is used in the analysis. The system can further be improved by learning to identify the utility of different combinations of techniques. This function may be implemented by combining various techniques from neural networks, CBR, and statistics.

This system also has a CAI capability to provide a realistic learning environment whereby health workers can practice skills for managing hypertensive patients and learn problem solving skills. Although widely used, traditional CAI programs have certain disadvantages. These programs offer the student a pre-recorded knowledge but cannot explain how or why a task is performed (Aegerter et al., 1992).

There were some limitations that should be dealt with to further enhance a capability of MDSS. First, number of cases were insufficient to the developement of knowledge models. Second, integration among HMIS, logistics regression model and expert system was inadequate to allow smooth information flow.

References

[1] Aegerter, P., Auvert, B., Gilbos, V. et al. (1992) An Intellegent Computer-assisted Instruction System Designed for Rural Health Workers in Developing Countries. *Method for Informatics in Medicine*, 31, p. 193-203.

[2] Chae, Y. M., Jang, T. Y. , Jung, S. K. & Park, M. Y. (1992) A Development of Decision Support System for Diagnosing Nasal Allergy. *Yonsei Medical Journal*, 33, 1, p. 72-80.

[3] David, W. & Lemeshow, S. (1989) Applied Logistic Regression. *Wiley-interscience Publication.*

[4] Ford, G. A. & Asghar, M. N. (1995) Management of Hypertension in the Eledery. *British Journal of Clinical Pharmacology*, 39, 5, p. 465-9.

[5] Garg, V. K. & Pepper, G. M. (1995) Essential Hypertension. *Medical Hypothesis*, 45, 3, p. 287-291.

[6] Garson, G. D. (1991) Interpreting Neural-network Connection Weights. *AI Expert*, 6, p. 47-51.

[7] Huh, K. P. (1992) Hypertension and Malmetabolism and Malendocrime. *Korea Journal of Medical Science*, 35, 2, p. 186-92.

[8] Jaffe, L. S. & Seely, G. M. (1995) Hypertension in Woman-extent and Limit of Our Knowledge, Current Problems in Obstetrics. *Gynecology & Fertility*, 18, 1, p. 2-35.

[9] Ohrr, H. C. (1993) Smoking and Total Mortality: Kanghwa Cohort Study, 6-year Follow up. *Yonsei Medical Journal*, 34, 3, p. 212-222.

[10] Rich, E. & Knight, K. (1991) *Artificial Intelligence, 2nd edition*, McGraw-Hill.

[11] Verdecchia, P., Schillaci, G., Borgioni, C. et al. (1995) Cigarette Smoking, Ambulatory Blood Pressure and Cardiac Hypertrophy in Essential Hypertension. *Journal of Hypertension*, 13, 10, p. 1209-15.

Lattice-based knowledge discovery in network management data

F.J. Venter, G.D. Oosthuizen and J.D. Roos
Department of Computer Science, University of Pretoria, 0002, South Africa
Phone: +2712 420 2361, Fax: +2712 436454
E-mail: fventer@cs.up.ac.za, goosthui@cs.up.ac.za, jroos@cs.up.ac.za

Keywords: knowledge discovery, data mining, network management, concept lattice

Edited by: Se Woo Cheon
Received: April 22, 1997 **Revised:** May 16, 1997 **Accepted:** June 6, 1997

Complex and large systems can be hard to manage if the elements that they comprise have dynamic causal relationships. Management of such a system usually entails monitoring and interpreting data generated by various elements of the systems. This data needs to be processed into meaningful information, so that a manager can use the information to get enough insight into the dynamic state of the system to be able to proactively influence the run-time behavior of the system. However, if the amount of data generated for a system is large enough and the complexity of the relationships between elements of the system is high enough, then traditional information analysis and presentation methods do not suffice to give the manager of the system a comprehensive understanding of the state of the system. What is needed is more intelligent analysis of the possibly enormous corpuses of data streaming to the management system in real time. The field that is well positioned to give this sort of analysis is Knowledge Discovery in Databases and Data Mining (KDD) - a process that has the goal to extract knowledge from data. A range of techniques, including neural networks, rule-based systems, case-based reasoning, machine learning, statistics, etc. can be applied to the problem. We will discuss a KDD approach that employs a concept lattice to determine dependencies between elements of a computer network. We will firstly discuss the problem form a network management (NM) point of view and then show how KDD and specifically lattice-based KDD will address the problem

1 Introduction

In order to understand a system, its elements and the way they interact need to be understood. It is only when this understanding is as comprehensive as possible, that successful operation and management of such a system is possible. In many cases, such as small systems, their characteristics are well understood by their inventors or designers and they do not present a problem that traditional information processing cannot solve. But a complex system comprising vast numbers of elements with complex behavior can become quite unwieldy and incomprehensible, especially when the dynamic inter-element relationships cannot be predicted by only considering the domain theory or design of the system. To monitor and interpret the behavior of such a system, data generated by monitoring agents needs to be collected on a regular basis, processed and transformed into a presentable format, so that a system operator/manager can quickly assess the most important inter-element relationships of the system. The format of the information that the investigator uses will depend on the type of system and can range from textual database query results to graphical representations and the results of elaborate statistical analysis. In large systems the number of variables to monitor and volumes of data that can be generated can become too large to interpret using traditional data analysis techniques. The goal of the exercise is at a different level: that of discovering the behavior and characteristics of the system from vast amounts

of input data. When the number of variables measured across all the elements in the system are vast, even the best statistical methods alone cannot give a clear picture of all the possible dependencies between them. What is needed is to go one step further, i.e. from information to knowledge. The overwhelming number of records of data and vast number of variables defined for a given corpus of data, makes a knowledge discovery approach to the problem more applicable than traditional information processing techniques. It is against the backdrop of a frustrated, data drenched community, flooded by large and complex data sets, that this field of knowledge discovery in databases(KDD) was born. KDD will be discussed in greater detail in a later chapter.

This paper will discuss the problem of knowledge discovery in the dynamic data of a computer network. The goal of knowledge discovery in this environment will be to enhance network management(NM) systems to such an extent that dependencies between managed objects(MOs) can be found and managers can understand and diagnose their networks better. An experiment was conducted to investigate the feasibility of a KDD technique called "Lattice based knowledge discovery" (Oosthuizen 1991) applied to the determination of dependencies between Simple Network Management Protocol (SNMP) variables. This lattice based technique uses a concept lattice as a framework for the knowledge discovery process.

2 A first look at the problem

The task of network management is to manage a networked computer system efficiently. Its importance is indisputable and a large world community is working hard at solutions for the network management problem. A consequence of this global effort is that protocols and models have emerged by which network elements can be monitored and controlled. This process includes the exchange and storage of great corpuses of management data. The "Internet", an existing wide area network is already supplying the globe with vast interconnectivity and the "Open Systems Interconnection" (OSI) standards of "The International Organization for Standardization"(ISO) promise to do even better(when/if they arrive). On a smaller scale corporate enterprise networks exist, which together with their WAN counterparts make a possibly massive body of data available to NM systems. Network management systems thus have as one of their tasks to present the network manager with the most comprehensive view of the network and its elements as possible. Some techniques used to help the network manager to visualize the state of the network are statistics, graphics, etc. But the NM system user still needs to make his own conclusions with respect to the state of his network. This is an example of many domains in which people need to cope with the "forests" when they are bombarded with millions of "trees". An example of a "forest" that needs to be seen is the true cause of a network error, in the midst of a plethora of cascading events that are propagated through the network. The computer industry is a prime culprit when it comes to this "tree cancer", but there are also other domains, like the retail industry, scientific research, etc. in which people need to induce the characteristics of the part of the world they are investigating - e.g. characteristics that are hidden in databases full of trend information or tables full of experimental values. It seems that the best that traditional information processing can provide is statistical analysis and glamorous graphical visualization techniques to help lift the lost investigator a few more meters above the data-labyrinth's the hedge. Even when the aim of the investigation is clear, e.g. "what causes this error", if the data set is massive and a lot of irrelevant patterns exist, finding the desired dependencies is hard. There are no shining diamonds in the pile of rubble. The problem may even be worse, however. We may not know what we are looking for in the first place, i.e. all interesting phenomena in the data set must be discovered even without a first-order definition of "interesting". This will make the amount of effort afforded to the human investigation orders of magnitude greater.

What is needed, is more than the results of statistical analysis or elaborate information presentation. Computers need to be equipped with some of the cognitive skills of humans to discover interesting dependencies while traversing great amounts of data. This means firstly that they need to exploit their ability to go far beyond the "seven plus minus two concepts at a time" limitation of humans. Secondly, they should also be

equipped with the logical analytical skills of which humans are still the only true masters. This will bring the best of two worlds - the computer's numeric processing power and the human cognition - together to take computer based information processing into the realm of true knowledge discovery.

The methods developed to solve this data mining problem need to be adequately exhaustive with respect to the patterns and dependencies that are discovered. This means, given a set of network values, all possible dependencies or patterns that exist among these values need to be generated. The resulting set of dependencies need to be presented in such a way that the user of these dependencies does not find himself in a new maze - the maze of dependencies. The set of dependencies will have to be presented in a well ordered structure. Traversing this structure should also be feasible for the human information investigator, who is in our case the network management technician.

3 A brief definition of KDD

According to (Frawley et. al. 1992), KDD is the extraction of previously unknown and potentially useful information from data. In a set of facts (data) D, a language L, and some measure of certainty C, a pattern is a statement S in L that describes relationships among a subset D_S of D with a certainty c, such that S is simpler (in some sense) than the enumeration of all facts in D_S. All patterns that are interesting to the user and are certain enough according to the user are collectively called knowledge. All patterns according to this definition that a program generates from a given data set, is called discovered knowledge.

This definition of KDD concentrates on the format of the output of the process and not the process itself. Key issues arise from the KDD process and therefore each of the elements of the process should be examined to get a more holistic view of KDD.

The KDD process consists of the following major phases:

- **Data preparation, cleaning and warehousing.** This is where issues such as noisy data, consistent formats, etc. are addressed. Prepared data should be maintained on a distributed database, allowing location transparent distribution up to relational level.

- **Data-driven exploration.** The analyzer needs to peruse the content of the data before starting the mining phase in order to identify possible interesting subsets of the data. This will help to reduce computing costs of later more complex and processing intensive phases. It is often the case that the analyzer does not even have a clear vision of the knowledge discovery goal. A first peek at the content and some exposition of the structure of the data can help to trigger deeper exploration into interesting areas.

- **Requirements analysis.** Since it is often not the same person that will be doing the data analysis that will be using the discovered knowledge, the requirements of the knowledge user need to be defined.

- **Search for interesting patterns(Data Mining).** During this phase, the clean data set is mined for possible patterns according to criteria set by the user. These criteria include the level of accuracy of found patterns and the user's biases with respect to interestingness of possible output patterns. This phase is also referred to as the "dredging" phase.

- **Presentation and navigation of patterns.** When the KDD process is interactive, intermediate results are presented to the user, mining parameters are refined and the search for patterns re-iterated, until the desired findings are reported. A more data-driven or bottom-up dredging of raw facts can be used to discover dependencies with less interaction with the user. In this case only the final results will be displayed for the user to interrogate. The user needs to understand the output knowledge. Therefore the discovered patterns need to be depicted in a high level language. A graphical paradigm should be used to depict the structure of the discovered knowledge space. The analyzer should have the ability to "browse" this space in order to incrementally dissect the content and implications of the discovered patterns.

4 KDD from a NM perspective

In order to complete the contextual prelude to the investigation that was done, we need to look at the benefits and results we aspired to achieve from a Network Management (NM) point of view. This means that the rationale for using NM as a suitable field of application for lattice based knowledge discovery needs to be clearly stated. In order to define this rationale, some core characteristics of networks and typical goals of NM will be pointed out, so that it can be clearly shown in later chapters how our lattice based knowledge discovery approach addresses these aspects of NM.

4.1 NM information overload

The vast number of network variables available through the Simple Network Management Protocol (SNMP) or Common Management Information Protocol(CMIP) makes a rich set of network values available to use as raw data source for subsequent dependency discovery. In order to supply the network manager with a comprehensive model of his network, this base of management information can be "mined" for interesting dependencies, that can be used for diagnostic or other purposes. This means that NM information bases can serve as excellent sources of raw data for interesting KDD and from the viewpoint of NM the exercise is well worth the while if the achieved results mean that networks can be made more understandable.

4.2 An integration model

The field of network management(NM) came into being as a consequence of networked computer systems. For the purposes of this paper, a networked computer system can be viewed as a system that consists of a number of computers connected to each other via a system of transmission media and intermediate connecting systems. However a more holistic approach to introduce networks should start at a description of the functionality of the systems that networks form part of. Such networked systems fulfill a set of requirements, that implicitly or explicitly allude to a need for communication of entities that are geographically apart. Examples of such systems are distributed systems(such as distributed operating systems), human-to-human communication systems(such as e-mail), client-server systems(such as automatic bank tellers) and many more. Each networked system comprises a unique collection of software and hardware elements that are interconnected and interact in a unique way. The dynamic behavior of the networked system can be expressed as a dynamic model of the elements in the system. Such a model would consist of well known inter-element relationships(derived from the theory of network system design) and unique system-specific inter-element relationships. Of particular interest is the way that the complete set of elements or factors are dependent on each other.

Having such a model would enhance the encapsulation of the real world devices (that current NM models encapsulate as managed objects (MOs)) with automatically derived dependencies between MOs, or on a finer grain, between properties of MOs. The following figure depicts a conceptual view of this new layer of "knowledge about element dependencies".

In the device layer are the physical devices of which certain low level variables can be monitored or set. The spheres depict the devices and the logical measurable variables per device are shown as "Var1", "Var2", "Var3" ... "VarN". Access to these variables are actually implemented in the information layer's agents. This picture just gives a logical exposition of where the device variables "reside". The information layer gives a managed object view of the device variables in the OSI framework. In the Internet framework (using SNMP), the variables are managed according to a management information base (MIB), a hierarchical ordering of variables.

We introduce a new layer called the "knowledge layer". It represents knowledge about dependencies between different NM variables. It should also supply a complete set of services to the application layer, so that applications can easily transform the encoded dependencies into an intuitive interface between the user and the knowledge. This means that apart from the traditional NM functionality that NM systems provide, they can present the higher level network state knowledge to the user. It could also provide a way that the user can navigate through the new knowledge space, which means that the traditional ways of viewing networks, such as a

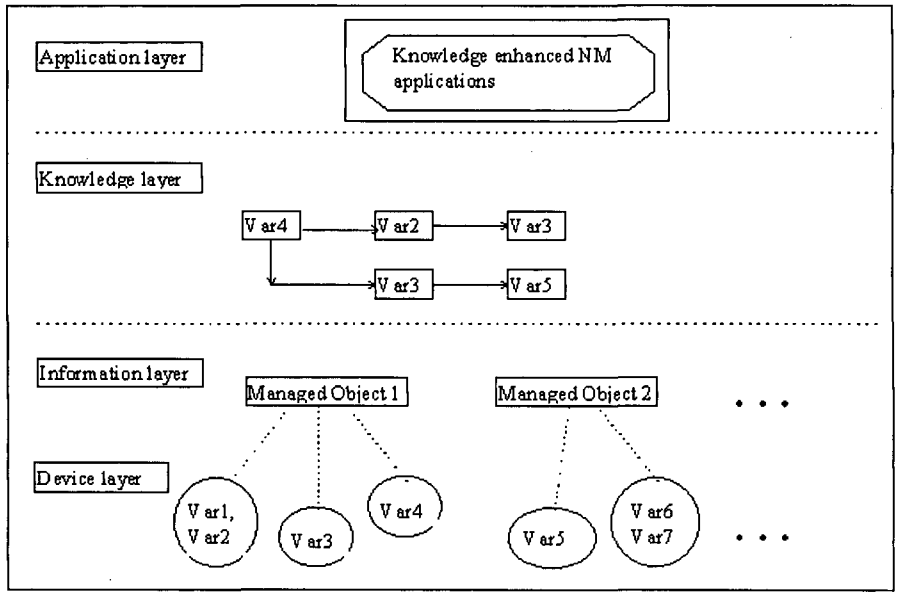


Figure 1: A NM knowledge integration model

topology maps, could be supplemented with alternate, higher level views of networks.

The knowledge in the knowledge layer can be derived using two sources of information:

- a. Network theoretical and technical domain knowledge.
- b. Machine discovery of inherent dependencies (using KDD).

KDD has a clear role to play in this framework: to (a) extend the theoretical and network specific domain knowledge: (b) dynamically and automatically.

5 The concept lattice approach

Although our approach to KDD stems from a machine induction technique developed by Oosthuizen (Oosthuizen & McGregor 1988), we have exploited the graphical nature of the base construct (the "concept lattice") used in the technique, so that we can address more of the processes mentioned above than just the "mining" step of KDD. This means that we have taken a machine learning technique as a basis and extended it to a comprehensive KDD solution. A set of software tools have been developed to demonstrate some of the points we are trying to make in this paper. We try to use a unified lattice based methodology to realize a subset of the processes of KDD. This approach is in contrast to a methodology where different parts of the KDD cycle is realized using different techniques/tools. When explaining our approach, we will start with the basic definitions of concept lattices, underpinning the knowledge representational basis of the approach, after which we will explain the machine learning technique that constitutes the "data mining" part of the process. Our description of concept lattices will be brief, since there are other references that give more complete expositions on concept lattices, e.g. (Oosthuizen & McGregor 1988),(Carpineto & Romano 1993),(Godin et. al. 1991) and (Oosthuizen 1991). Finally we will show how we propose to extend the approach to address some of the other processes of KDD.

5.1 Defining the Concept Lattice

When discussing any intelligent system, classically the first issue to address is how *knowledge is represented.* As mentioned above our technique represents knowledge in terms of *concept lattices.*

A lattice is a *directed acyclic graph* (DAG) in which every two nodes have a unique nearest common descendant - or *meet* - and a unique nearest common ancestor - their *join.* The lattices discussed here are of a special kind, called *concept lattices* (Wille 1982), that have the following ad-

ditional properties:

- 1. Apart from the children of the universal node at the top and the parents of the NULL node at the bottom, no other nodes in the graph have exactly one parent or exactly one child.
- 2. No node has a parent (i.e. no node is directly linked to another node) to which it is also indirectly linked by means of a path that goes via one or more other nodes.

Although Wille formally introduced the notion of a concept lattice (Wille 1982), the particular organization of data described here also corresponds to the so-called *cladistic* approach to classification used by biologists and linguists for some time (Hoenigswald & Wiener 1993). In fact, the fundamental idea behind concept lattices dates back to Aristotle who noted the inverse relation between the number of properties required to define a concept and the number of entities to which the concept applied. This is referred to as *the duality of intention and extension* (Sowa 1984). Each node represents a highly repetitive pattern (set of features/attributes of the input data).

The concept lattice provides us with one unified structure that contains many tangled, optimally integrated trees (hierarchies) of nodes. For each data set there is a unique, minimal (in number of nodes) concept lattice. These characteristics makes it possible to derive all n-ary relationships between all attributes of the input data set.

The following figure illustrates an example of a very small, simple concept lattice.

Note that we distinguish between three types of nodes: attribute (or feature) nodes; concept nodes and entity nodes. The attribute nodes are all single-valued symbolic assertions about any data-entity, e.g. hair=blonde, eyes=blue, eyes=brown, complexion=dark, etc. Entities are the input data rows that we get from the database that the system is parsing. Entities are also referred to as examples or data-tuples or data-records. Each entity is usually a tuple of attributes, e.g. eyes=blue, hair=blonde, complexion=fair,. classification=positive and is derived from the raw data set. The internal nodes called "concept nodes" are created when the lattice is formed during graph induction (also called knowledge base normalization). These are the nodes that relate attributes to each other and constitute the "knowledge" of the lattice. The concept nodes also have strengths(i.e. how many entities they cover), e.g. in figure 1, node *1 has a strength of 3 and node *2 has a strength of 2. Strengths also give an indication of the confidence of the relationships that concept nodes portray. The universal node that is connected to all attributes at the top and the NULL node that is connected at the bottom has been omitted for readability reasons. Lattice based knowledge base normalization has been proved as a useful technique for induction (Oosthuizen & McGregor 1988).

5.2 Mining for knowledge(concepts): From data to concept lattices

A concept lattice is constructed by creating a node for each data point at the bottom of the graph (e.g. nodes E1-E5 at the bottom of figure 1 represent the corresponding data records in Table 1) and a node for each attribute-value at the top. During this process internal nodes are created between the data points at the bottom and the attributes at the top (they are marked with *'s in figure 1). Each data point is then connected to its respective attributes whilst ensuring that the graph remains a lattice (see figure 1). It can be shown that a given set of entities give rise to a unique lattice. The exact manner in which the lattice is constructed is beyond the scope of this paper (algorithms can be found in (Oosthizen & McGregor 1988), (Carpineto & Romano 1991), (Rodin et. al. 1991), (Oosthuizen 1991)).

Each internal node denotes a pattern of attributes that occurred in more than one data point. Since all possible combinations of attributes could potentially occur in the data, the number of nodes in a lattice is equal to the size of the powerset of the number of attributes per entity, i.e. 2^n, where n is the number of attributes per entity. The actual number of nodes realizing in real world data sets is, however, only a fraction of this amount. This is also why the lattice is useful as a data analysis tool: each of the internal nodes in the lattice represents a regularity in the data. And to ensure that the graphs are not cluttered by accidental coincidences, statistically insignificant nodes are removed from the lattice.

Let us consider an example. For simplicity we consider a cleaned, integrated database in

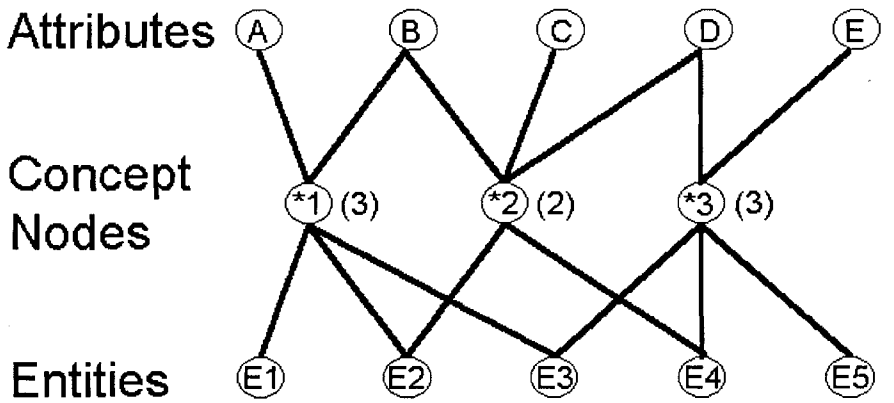


Figure 2: A small concept lattice

the form of a single file of variable length tuples of symbolic features(attribute values). Figure 3 shows a lattice generated from the data in table 1. It depicts underground sample information regarding rock samples collected for laboratory analysis.

The lattice in figure 3 appears rather irregular, but this is because we are representing an n-dimensional structure in two-dimensional space, and only the patterns of attributes that actually occurred in the data are represented.

Each concept in the lattice is described (intention) by the attributes that are included in its so-called "upper closure", i.e. all nodes that are transitively ancestral to ("above") the concept node in the lattice. All entities that pertain to this concept can be found in the downward closure of the node. The strength (confidence) of a concept is the number of entities pertaining to it, i.e. the number of entity nodes in its downward closure.

it Inference on subsets of attributes is possible as follows: To determine the implied set of attributes (we call it R), given an input set of attributes (we call it the set Q), the it meet of the set Q is determined (call it node M).. The upward closure of this node (M) is then taken. If the upward closure contains any attributes apart from those in Q, then these are implied by the ones in Q, i.e. Q -> R. For a more thorough discussion of this inference on a lattice see (Oosthuizen & McGregor 1988).

5.3 Exploring knowledge: A map of concepts

As opposed to some other KDD techniques where killer queries and/or a highly iterative approach is followed, lattices allow a graphical representation of the discovered knowledge space that can be navigated (browsed) by the user. A refinement of the induction algorithm has also been developed that employs a pruning regime to reduce the size of the lattice and facilitate focusing of the KDD process. We have developed a program, called GRAND (GRAph iNDuction) that implements the lattice induction algorithm and DATAMAP, that implements the graphical depiction and navigation of the induced lattices. A very thorough discussion on visualization of categorical data using lattices can be found in (Oosthuizen & Venter 1995). For the purpose of this paper we will give a brief description of how a user can navigate the lattice in DATAMAP.

it Visual display and navigation of the lattice

DATAMAP displays the lattice as a two-dimensional abstraction of the n-dimensional structure of the lattice, but unlike the toy example in figure 3, real lattices are too large to display on the screen in their entirety. Consequently only the relevant parts of the lattice are displayed. DATAMAP allows the user to firstly select an initial set of attributes from the recorded set of attributes in the database. DATAMAP then takes the user to the node that is the meet of the selected attributes. We call this the it focus. See the square node in figure 4. The user can then immediately see all the attributes of the focus (all

Entity no	Size	Color	Shape	Contains Heavy Metals	Structure
E1	Small	brown	regular	yes	hard
E2	large	brown	irregular	no	brittle
E3	large	yellow	regular	yes	hard
E4	small	black	regular	no	brittle
E5	large	black	regular	yes	brittle
E6	large	brown	regular	yes	hard
E7	large	black	irregular	no	brittle
E8	small	brown	irregular	no	brittle
E9	large	brown	irregular	yes	brittle

Table 1: Input tuples containing data for underground rock samples

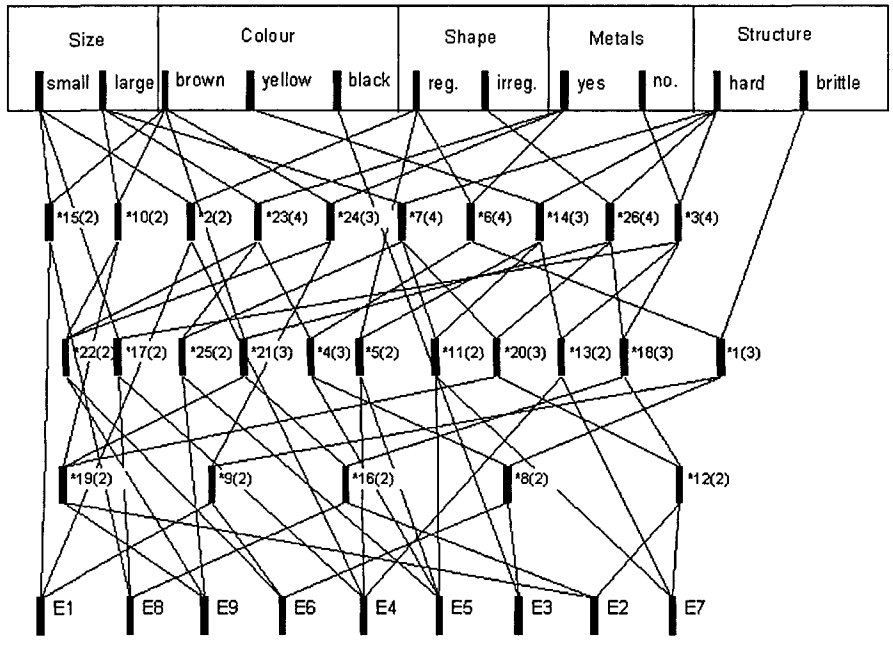


Figure 3: A lattice generated from data in Table 1

the attributes in the upper closure of the focus). All additional attributes in this list, that were not selected initially are the implied attributes as discussed above. The user can then decide to select another start set of attributes, i.e. explore dependencies of another set of attributes, or select another node of the lattice. If he selects another node in the lattice, that node becomes the new focus. Nodes above the focus are labeled by attribute names preceded by a '-'. This means that selecting one of these nodes will imply subtracting an attribute/s from the current set of attributes. Nodes below the focus are labeled by attribute names preceded by a '+'. The nodes in the lattice are thus labeled in terms of the difference with respect to their nearest parents (for nodes under the focus) or nearest children (for nodes above the focus). This means that selecting one of these nodes will imply adding an attribute/s from the current set of attributes. This helps the user to add/subtract features from the goal concept that he/she is searching for. Clicking on a node also moves the node(the new focus) to the center of the screen. If node *6 in figure 3 is the current focus, then the screen appears as shown in figure 4. If the user then selects node *9, the screen is updated as shown in figure 5.

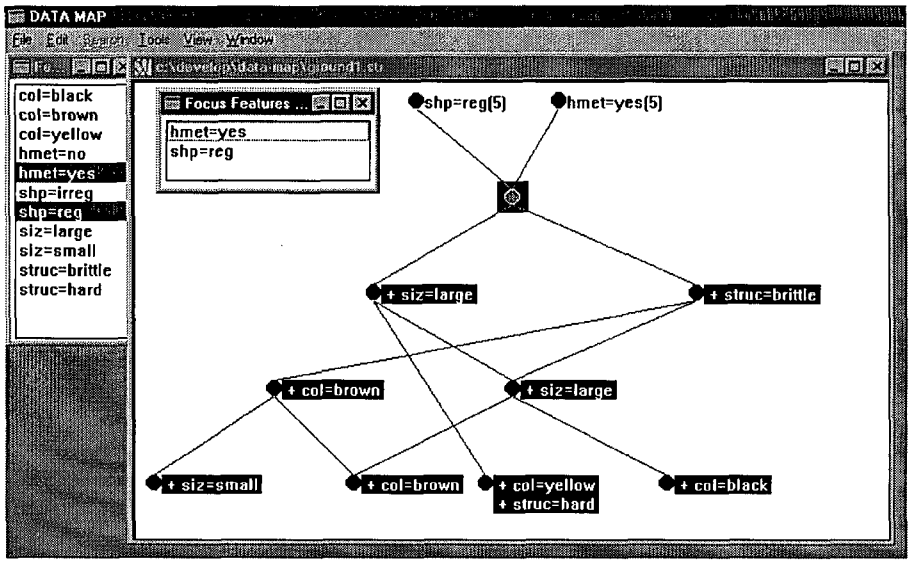


Figure 4: A lattice viewed in DATAMAP

6 The experiment

We conducted an experiment to test the feasibility of the lattice-based technique. The real world network that was used for the experiment is the Internet nodes forming the wide area network (WAN) between the South African universities (called the it UNINET). In order to test the success of the research, the following goals were set:

- 1. Develop an experimental system that will be able to facilitate the application of lattice based knowledge discovery on SNMP data collected from the UNINET. This system must be able to collect values of selected SNMP variables from selected agents, transform them into a format applicable to the lattice learning program and present the resulting dependencies to the user.

- 2. Use above mentioned system to do extensive experimentation with various combinations of SNMP variables and agents in order to prove that well known dependencies are induced, i.e. the application of the lattice-based technique, that should work in theory, is proven in practice.

- 3. Use the experimental system to investigate further interesting combinations of variables.

- 4. Conclude on the feasibility and applicability of the lattice based technique. If the technique does prove to live up to its promise then propose a model of implementation of the findings into a network management system.

- 5. Report on all possibly interesting issues arising from the research and point out which of these should be topics for further research.

6.1 Experimental procedure

The primary device in the UNINET for which SNMP variables can be measured is the router interface. In logical terms, an interface is a port at which a serial or ethernet transmission line can be connected. It can also be seen as the input/output port of the router through which all traffic flow. Each router in the UNINET has one or more serial interface/s and one or more ethernet interface/s. The serial interfaces connect routers to form the UNINET WAN circuit and each ethernet interface connects a local network of a university to the UNINET. The SNMP MIB defines a set of paramaters to measure all major traffic flow quantities on each interface of a router. We decided to select a set of SNMP parameters defined for interfaces and take samples of these parameter values over time. These samples were processed and entered into the lattice generation program

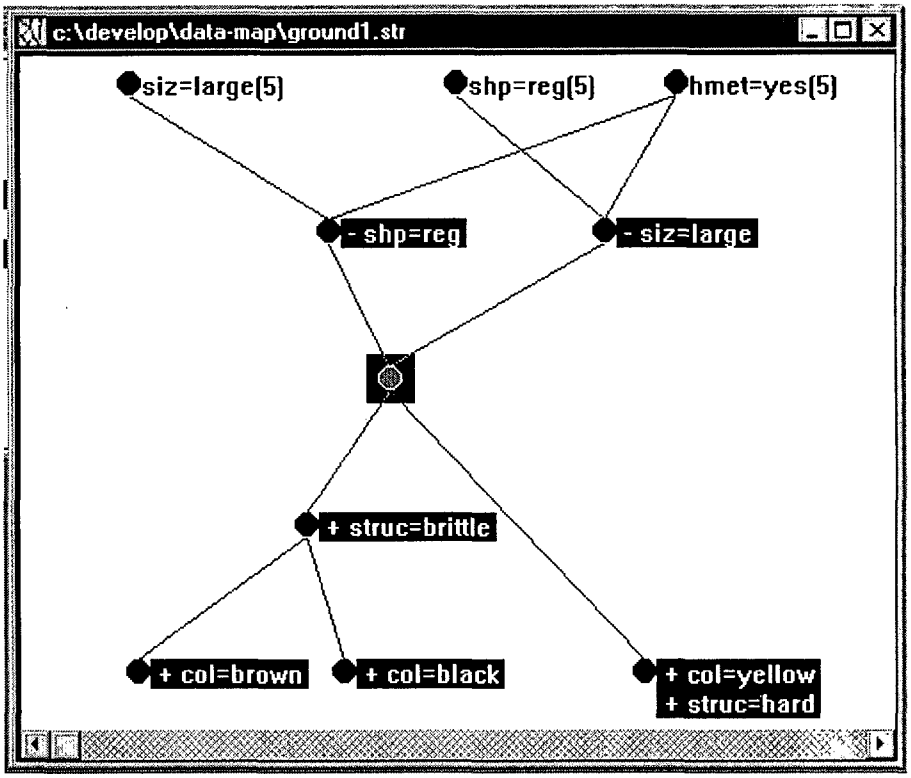


Figure 5: A lattice viewed in DATAMAP, after node *9 was selected

GRAND. The generated lattices were then presented and analysed using a lattice display and navigation program DATAMAP. Both of the programs were discussed above. The experiment was divided into two phases:

- The control experiment (Proof of concept). In this phase we tested the ability of the technique to automatically derive well known relationships between interfaces. During this phase the experimental tools were developed and tested.
- Discovering new parameter dependencies. In this phase we started to explore more combinations of SNMP variables in order to find new relationships between variables that may not have been known before.

6.2 The control experiment

To prove the feasibility of the lattice based KDD in NM concept, we sampled some vendor specific SNMP variables e.g. "locIfLoad" over several interfaces on different routers to determine whether well known load distribution dynamics could be determined by a latttice. We developed some Unix scripts to do the raw data collection into a textual database of SNMP information. AWK scripts were used to transform the raw data into tuples of symbolic attributes. The generated concept lattices indicate that high/medium/low load on selected serial interfaces were caused by corresponding high/medium/low input load on the ethernet interfaces. This result is consistent with the expected behaviour of the network.

6.3 Discovering new parameter dependencies

The control experiemnt was a very simple analysis problem and only proved once again that the concept lattice can learn, albeit on NM data. More elaborate experimentation was clearly needed to prove the suitability of the technique for NM specifically. This lead us to consider the problem of selecting appropriate SNMP variable combinations for experimentation. The first factor that we needed to keep in mind is the potentially explosive growth of lattices with respect to the number of attribues under investigation at a time. The real

challenge is to select parameters and interfaces in such a way that a canonical set of information elements with respect to a certain problem domain are selected. Using the load distribution example again, if we want to determine what factors influence load on a specific interface, then we should at least include load measurements from all neighbouring interfaces. Leaving one neighbouring interface out can result in an inaccurate picture. This problem is really a form of the it framing problem, or the problem of drawing the correct boundaries around the it closed world in which an experiment is executed. However, in order to determine new, previously unknown dependencies, some parameters that are theoretically not so closely related to the core paramater set should also be included in a specific experiment. We have started to define some areas of investigation, but the NM parameter selection for KDD is in itself a topic for further study. Consentrating on experimentation on interfaces, whe have decided to define two categories of parameter sets:

- Inter-interface experimentation. Here we keep the number of SNMP variables to a minimum and see how interface influence each other with respect to one or two variables. Sampling is done from a set of interfaces over time.

- Intra-interface experimentation. Here we try and investigate the influence of SNMP variables on each other on an interface. Sampling is done on several interfaces.

There are potentially numerous combinations of SNMP variables/interfaces to investigate together. The ideal tool to dynamically explore influences that different variables could have on each other, should allow the user to dynamically add/remove variables into/from a concept lattice. Currently however, the process is manual experimentation and we will have to keep to carefully selected parameter sets.

7 Discussion

7.1 Lattice-based KDD in general

KDD is a promising field of research and successful applications prove that industry is already cashing in on substantial return on investment. We believe the technique desecribed above could prove to be a valuable contribution to the field of KDD. In order to bring the current technique closer to the general KDD goal, we will have to relate aspects of our approach more closely to the previously mentioned KDD processes. We therefore firstly revisit each process from the lattice based KDD view point:

- it Data preparation, cleaning and warehousing We employ pruning techniques to eliminate weak concept nodes that could have been caused by noisy data.

- it Data-driven exploration Since we expose the underlying dependencies of the raw data graphically, the user has a rich exploration environment in the form of the DATAMAP program. This helps the user to be "led" by the system through the maze of possible "interesting" relationships to explore further.

- it Requirements analysis This process is currently beyond the scope of our aspirations.

- Search for interesting patterns(Data Mining) This process is well addressed as discussed in the major part of this paper. Lattice based machine learning forms the basis of the data mining phase.

- Presentation and navigation of patterns This is probably where our technique makes the biggest impact. The graphical depiction of the learned knowledge and navigational and explorational operations that our system facilitates as discussed in the paper allows not only a rich interactive environment during the exploration-and-mining cycle, but also presents knowledge as part of the unified view on the normalized knowledge base.

7.2 Lattice-based KDD in NM

With respect to the application of the lattice-based technique to NM, the preliminary results obtained from experimentation suggests that certain NM problems may be solved using the technique. This means that the application of the technique should be constrained to specific problem areas, where the notion of "what is inter-

esting" is well known - such as diagnosing certain error conditions or learning the dynamic state of certain predefined sets of network elements/variables that would add some value to the NM solution.

We believe that although we tried to achieve the specific goal of applying lattice-based KDD on NM, some general issues arise that are applicable to all dynamic environments such as computer networks. Some of these issues can be enumerated as follows:

- Most NM variables are time dependent. The way that the time dimension is handled should be addressed by either introducing it as another attribute or a higher level approach should be followed to correlate sets of dependencies with respect to time
- Pearl in (Pearl 1991) has indicated a distinction between data mining and data compression. He has indicated that summarization of data into more compact or other formats is adequate for concept formation, classification and prediction. However, in order to form causal models a technique that can do much stronger generalization is needed. In the NM context, the KDD process should relate the way that elements/services/applications/users under investigation influence each other directly or indirectly. This means that our technique needs to be able to have the power to discover stronger dependencies such as causal relationships. In practical terms our pruning/node constraining techniques need to be able to take causality into account.
- Since most NM variables are numeric and lattice-based KDD is in principal a symbolic learning mechanism, pre-processing of numerical values in order to categorize it into symbolic values need to be done. This area needs to be researched to greater detail.
- A more comprehensive analysis of specific NM aspects that allude to KDD should be done. The extent to which lattice-based KDD addresses these aspects should be compared to other KDD techniques. This would indicate if other techniques may be more suitable as KDD mechanisms in NM or if the lattice approach could be altered so that it competes favorably with the best KDD in NM contenders.

References

[1] Frawley W.J., Piatetsky-Shapiro G. & Matheus C.J. (1992) Knowledge Discovery in Databases: An Overview. *AI Magazine, Fall 1992.*

[2] Oosthuizen G.D. & McGregor D.R. (1988) Induction through Knowledge Base Normalization. *Proceedings of '88 European Conference on Artificial Intelligence.*

[3] Wille R. (1982) Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. *Ordered Sets*, Dordrecht-Boston, p. 445-470.

[4] Hoenigswald H.M. & Wiener L.F. (1987) Biological Metaphor and Cladistic Classification: An Interdisciplinary Perspective. *University of Pennsylvania Press*, Philadelphia.

[5] Sowa J.F. (1984) Conceptual Structures *Information Processing in Mind and Machine*, Addison-Wesley.

[6] Carpineto & Romano (1993) An order-theoretic approach to conceptual clustering. *Proceedings of the '93 International Machine Learning Conference*, Amherst, p. 33-40.

[7] Godin R., Missauoui R. & Hassan A. (1991) Learning Algorithms using a Galois Lattice Structure. *Proceedings of the '91 IEEE International Conference on Tools for AI*, San Jose, p. 22-29.

[8] Oosthuizen G.D. (1991) Lattice-Based Knowledge Discovery. *Proceedings of AAAI'91 KDD Workshop*, Anaheim, p. 221-235.

[9] Oosthuizen G.D. & Venter F.J. (1995) Using a Lattice for Visual Analysis of Categorical Data. *Perceptual Issues in Visualization*, Springer-Verlag, p. 142-155.

[10] Pearl J., Dechter R. & Verma T. (1991) Knowledge Discovery vs. Data Compression. *Proceedings of AAAI'91 KDD Workshop*, Anaheim, p. 191-194.

Organizational Science Approach to Knowledge Intensive Learning and Adaptation in a Multiagent System

Hiroshi Hatakama
Fujitsu Limited
2-15-16 Shin-yokahama, Kouhoku-ku, Yokohama 222, Japan
Phone: +81 45 476 4586, Fax: +81 45 476 4749
E-mail: hatakama@sysrap.cs.fujitsu.co.jp
AND
Takao Terano
Graduate School of Systems Management, University of Tsukuba
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112, Japan
Phone: +81 3 3942 6855, Fax: +81 3 3942 6829
E-mail: terano@gssm.otsuka.tsukuba.ac.jp

Keywords: multiagent model, organizational sciences, explanation-based learning (EBL)

Edited by: Se Woo Cheon
Received: April 22, 1997 **Revised:** May 16, 1997 **Accepted:** June 6, 1997

This paper proposes a knowledge intensive learning model in a multiagent system, based on the concepts described in organizational learning of management science literature. We examine the validity and feasibility of the model in order to apply it to distributed heterogeneous knowledge systems. We have implemented two typical variations of the model: the Specialists-Model and the Generalists-Model. Using the two variations, we carry out simulation studies on the dynamic behaviors of decision making and learning in the organization of agents. The experiments showed that the model had unique adaptability to the change of environments, by enabling agents to efficiently supplement knowledge each other.

1 Introduction

How do adaptive behaviors emerge in a multiagent system? How can the agents share knowledge in cooperative problem solving? To answer the questions, we must address learning mechanisms in a multiagent system.

In this paper, to uncover the above questions, we propose a knowledge intensive learning model, of which the knowledge intensive learning mechanism of a multiagent system consist; by which we will analyze emerging adaptive behaviors of multiagent application systems, e.g., distributed heterogeneous knowledge systems; and for which we incorporate the concepts of organizational learning in management sciences. Unlike usual approaches to multiagent learning via induction, we rather focus on (1) knowledge intensive adaptive learning by agents, whose knowledge is insufficiently shared, and (2) learning mechanisms in which agents try to continually solve similar problems. To develop the model, therefore, we integrate multiple knowledge intensive learning schemas: explanation-based learning (EBL) within an agent, learning of premises to apply the knowledge in cooperative problem solving, and learning about the reliability among agents using credit-assignment. To examine the validity and feasibility of the model, we will demonstrate the dynamic behaviors of the model via simulation studies.

This paper is organized as follows: In section 2, the background and motivation of the research are described. Following the discussion of section 2, a multiagent learning model is proposed in section 3. In section 4, we describe experimental results on two variations of the model: the Specialists-Model and the Generalists-Model. Based on the

experiments, in section 5, we discuss the two issues of distributed heterogeneous knowledge systems: the emergence properties of adaptive behaviors and the role of knowledge sharing among them. Finally, in section 6, summary and concluding remarks are given.

2 Background and Motivation of the Learning Model

In this section, we first discuss problems, which would occur in knowledge systems with shared ontology. Next, we make critical review of the artificial intelligence (AI) models for activities of agents. Then, we introduce the concepts of organizational science into multiagent learning models.

2.1 Problems in using shared ontologies

KSE (Knowledge Sharing Effort) provides a framework for heterogeneous knowledge-based systems to share knowledge (Neches et al. 1991). Based on the concepts of shared ontologies, agents or knowledge-based systems communicate each other using ACL (Agent Communicating Language). Also, CommonKADS model concerns not only expert knowledge, but also the various characteristics of how that knowledge is embedded and used in an organizational environment (Schreiber et al. 1994). KSE and CommonKADS seem promising. However, we believe that cooperation based on extracted ontologies is impractical when each agent has heterogeneous domain knowledge, because, in practice, it is very difficult to extract the shared ontologies from heterogeneous domain knowledge in the context of rapid increases of various distributed systems, for example, the WWW (World Wide Web) environments. Furthermore, when a system must cope with the knowledge over human-interface, it is impossible to coordinate ontologies to be shared.

We would like to investigate how words of the same meaning can be learned by agents in the context of their activities, rather than the assumption that shared ontologies can be extracted by some omniscient entity. Because similar words are sometimes used by same meaning in the context of activities by individuals or departments who have disparate knowledge, we study the framework in which agents can learn and unlearn such a relationship of words adaptively.

2.2 AI models for distributed activities

Recent researches on DAI (Distributed Artificial Intelligence) pay attention to organizational models in distributed problem solving and learning (Bond 1988, Aiba & Terano 1995, Terano & Oikawa 1996, Terano 1997). These DAI-based models are able to represent many features of real organizations, such as concepts of coherence, authorities, commitments, task decomposition and specialization. However, these models have not yet attained to describe the flexibility and practicability of real organizations.

Gasser (1993) investigates social aspects of knowledge and integrates different kinds of local knowledge based on DAI concepts. He suggests that social facts reside in collectivities not in individuals and that they ground knowledge in the practical behavior and context of a social interaction as pragmatic view of knowledge. In this paper, we will model Gasser's concepts in more concrete manners, and show the effectiveness of the model by simulation studies.

In Plural-Soar system (Carley 1992), each agent has learning capabilities of Soar system. Schwartz (1995) tries to apply double-loop learning concepts to multiagent learning. He indicates the effectiveness of meta-learning. Though these systems concern the problem of communication among agents, the problem of knowledge-sharing is not investigated.

Minsky (1985) discriminates between learning from failure and learning from success in his "The Society of Mind" theory. He points out that learning from success leads to relatively small improvement, while learning from failure leads to more productive thought even though it involves risk. Although granularity of his agents-model is finer than ours, we employ these learning mechanisms in our multiagent model. He says "most pairs of agents (in mind) can't communicate at all." We believe that most pairs of individuals equipped with different specialized knowledge in a real organization cannot communicate sufficiently either, because their ways of reasoning and terminologies are too specialized to communicate enough.

In the previous papers, we have extended the framework of EBL to distributed environments (DEBL) (Terano et al. 1994). In the previous model, an agent learns by EBL, and if an agent cannot generate explanations of a certain concept, then the agent distributes it to other agent as (sub-)goal concepts. The model we propose in this paper is an extension of DEBL, and learning among agents is more loosely-coupled, such a way that agents can work concurrently and adaptively.

2.3 Introducing the concepts of organizational science

While restructuring or reengineering methods are to force to improve organizational structures from outside the organization, the concept of organizational learning in organizational science emphasizes the activities to improve embedded learning mechanisms which intrinsically exist inside of the organization by instructing members the way of thinking (Levitt & March 1988, Senge 1990). In the proposed model, we will computationally represent such embedded learning mechanisms.

3 Description of the Multiagent Learning Model

3.1 Basic ideas on the model

In this model, each agent has a heterogeneous knowledge structure, and using the knowledge, reasons, makes decision and learns simultaneously from each viewpoint. Series of decision problems are repeatedly given to the multiagent system. On each of the problems, the system responses the decision after cooperative decision processes. Against each response of the problem, the value true or false is given by an oracle, or an informant, from outside the system, that is, the environment. True or false value respectively means that the decision is correct or not. Each agent recognizes the oracle, however, it cannot independently understand the explicit reason, that is, it cannot understand the whole causal relation how the oracle can be derived.

We focus on combination of different types of learning methods to improve adaptation-ability of multiagent systems. In the multiagent model, agents have three types of learning methods, learning within each agent (i.e. EBL-type learning), learning between agents using premises (i.e. learning complementary relation from past examples) and meta-learning (i.e. inter-agent credit-assignment learning), and these learning methods are integrated in order to solve problems effectively. In such situations, we investigate dynamic behaviors of the agents. Each agent can have any type of learning mechanism individually. Typical learning within an agent is EBL or cognitive learning, such as knowledge compilation in ACT* (Anderson 1983), in order to be more effective for the next time.

Simon (1976) suggests that decision in organization is composite. In "composite decision," decision premise plays an important role in organization. We pay attention here to the reasoning (interpretation) of others that can be found in meetings or informal conversations as premises of the decision. As "social facts" suggested by Gasser, these premises should be found in the context of agents' activities. In our model, the premises are found by searching complementary relationship between agents' reasoning for correct decision from the log of recent activities. We concentrate on premises influenced purely by intelligence and not forced by authorities, because we focus on the even-relationship among agents. We assume two-step decision by each agent and allow it to regard other agent's reasoning. We also adopt meta-learning such as credit-assignment learning, in order to select the most reliable agent at each time.

3.2 Dynamics of the model

Figure 1 shows the overall dynamics of actions of agents and meta-rules in the model. When they are given an example to solve, each agent first reasons by itself and it put its reasoning results on a blackboard to communicate. Then, it decides by itself using other's reasoning results, and according to specific meta-rules their decisions are unified into the whole. When each agent reasons in the first step, it utilizes the results of knowledge compilation learning within it. When it decides by itself in the second step, it utilizes both the reasoning of other agents which it relies on and the results of premises learning. We assume that agents can learn which reasoning result of other agents to rely on, from the case-based past

records from which they can induce how knowledge should be complemented in the context of activities.

Decision making strategies as a whole are given by meta-rules, such as selecting decision-makers among agents or weighing each decision using credit assignment, and simply majority decision. These meta-rules are similar to the intelligent influence rather than authorities in organization. After decision-making as a whole, feedback-learning is made depending on whether success or failure.

When agents succeed as a whole, every correct agent learns positively how to reason by itself and compiles its knowledge, and the meta-rules are positively feedbacked as it has done. This way of learning is similar to the environment of real organizations that every department is always demanded to improve efficiency. As long as an organization keeps succeeding as a whole, neither the flow of work between departments nor the way of the information exchange among departments can change easily, although improvement in each department may advance. When they fail, each agent reconsiders the premises about other agents, and the negative feedback is made to improve the meta-rules. This way of learning is case-based, because premises should be found in the context of activities. These learning mechanisms have the useful advantage of enabling to learn empirically by using others' reasoning results as premise of decision even if mutual knowledge structure is independent.

4 Experimental Results using Generalists-Model and Specialists-Model

We show the dynamic behavior of the model by making a preliminary experimentation in this section. We have implemented the model by using Prolog to handle simple problems and to simulate them on a computer. We develop two typical variations of our basic model, which we call the Specialists-Model and the Generalists-Model. The two variations are based on different styles of real-life organization. Each variation solves a simple cup concept learning problem organizationally in a different way. The activities demonstrate that agents continue to supplement each other's knowledge to solve problems given from environment to them.

Simple cup concept learning problem handled in this section is as follows:

- stable $\wedge$ liftable $\wedge$ open-vessel $\Rightarrow$ cup
- graspable $\wedge$ lightweight $\Rightarrow$ liftable
- has-handle $\Rightarrow$ graspable
- width-small $\wedge$ ceramic $\Rightarrow$ graspable
- width-small $\wedge$ styrofoam $\Rightarrow$ graspable
- has-bottom $\wedge$ flat-bottom $\Rightarrow$ stable

For simplicity, we do not use sub-concept of open-vessel. The last rule is eliminated in the Specialists-Model.

4.1 Specialists-Model

We assume all agents are specialists in this variation. Each is equipped with a specialized knowledge for a specific domain; these domains do not intersect each other. We assume that their knowledge is independent by specialization and specializes further as they cooperate.

For solving a simple cup concept learning problem organizationally by six agents, each agent knows different aspects of a cup and cooperates to distinguish a given object to be a cup or not. The six aspects are "has- handle," "width-small," "material (ceramics or styrofoam)," "lightweight," "open-vessel" and "stable." On the aspect of "material," the agent (Agent-C) has the mechanism of activation-value, and one of the materials is not recognized when its activation-value becomes lower than the fixed threshold value. We assume each agent can communicate with only two other adjacent agents.

Rules that six agents have are as follows:

Agent-A has-handle $\Rightarrow$ cup

Agent-B width-small $\Rightarrow$ cup

Agent-C styrofoam $\Rightarrow$ cup, ceramic $\Rightarrow$ cup

Agent-D lightweight $\Rightarrow$ cup

Agent-E open-vessel $\Rightarrow$ cup

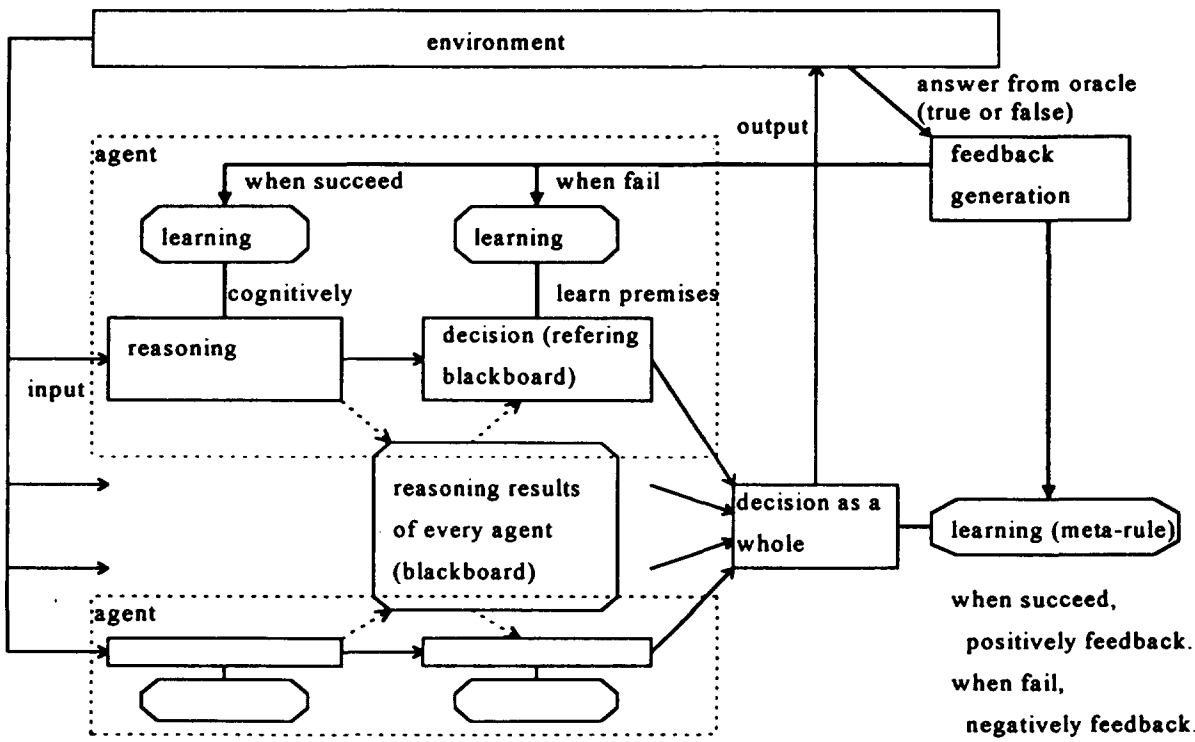


Figure 1: Multiagent model to represent the dynamics of organizational intellectual activities

Agent-F stable $\Rightarrow$ cup

Goal that not explicitly given to agents is as follows:

[$(A \vee (B \wedge C)) \wedge D \wedge E \wedge F \Rightarrow$ cup] (X means rules of Agent-x)

The situation is similar to intellectual activities in simple organization that consists of specialists or specialists-groups equipped with specialized knowledge. For example, six agents represent the planning department, the design department, the manufacturing department, the quality-assurance department, the distribution department and the sales department. They are related only to preceding and succeeding departments and their activities are linked circularly. The six specialized departments in the organization repeat a decentralized problem-solving and learning as each cooperates with the preceding and succeeding departments. We assume that their knowledge is not easily shared, because they are specialized too much. Each agent has narrow scope of cognition in the activities of an organization, like as "the prisoner of system" (Senge 1990). We would like to examine adaptablity and problems of the situation.

When each agent is given an example from environment, it reasons and puts reasoning results on a blackboard, on which information is shared among agents. Then each agent makes a decision considering the reasoning of the two other agents. Each agent can learn which reasoning result of other agents to rely on at each time. And when it fails, it tries to find out the complementary relationship between self and other agents, by inducing how they should rely on other agent's reasoning result, from past ten times experience records of the reasoning result (true or false) and the correctness (success or failure) of the agent itself and the both adjacent agents.

Next, the decision as a whole is made by decision-makers (a main-decision-maker and two sub-decision-makers), who are selected among agents using a credit-assignment method. Finally, each agent learns according to the result (success or failure). Because knowledge is distributed, the causal relation of the whole is not understood easily in the organization of the reality. As the environment changes, the relation of supplementing

knowledge also changes.

Credit values are assigned to each agent as follows:

- If it succeeds and they fail as a whole, add 6.
- For decision-maker agent, if it fails and they fail as a whole, subtract 2.
- For decision-maker agent, if it succeeds and they succeed as a whole, add 1. (Unless they continue succeeding over 5 times: This prevents them against inflexibility of meta-rules)
- For non-decision-maker agent, if it fails and they fail as a whole and it can not learn the premises from the adjacent agents, subtract 2.

The ways of operating credit values means following meta-strategy. An adaptive organization may try to adopt individual who has different idea that might be useful for the recent situation as decision-maker, when it fails.

Figure 2 shows the result of the simulation. In the graph, success or failure as a whole and credit-value of each agent are illustrated. The organization seldom arrives at the fittest condition. Rather, it shows adaptive behaviors. Sometimes, the organization becomes difficult to adapt to a new environment. We classify the condition of being difficult to change into three types:

Overconfidence of each agent: When an agent compiles its knowledge exceedingly, it fits the current environment too much to adapt itself to the new environment.

Trusting other's knowledge too much: When an agent trusts the judgment of another agent too much and keeps it as the premise of its own decision too long, the agent persists in relying on the other agent to an inappropriate degree and cannot change.

Inflexibility of meta-rules: When meta-rule fits the current environment too much, it is difficult for the organization to adapt itself to changes of environment.

The three types of ill condition should be avoided in developing multiagent system. Interestingly, the three types of problems correspond to the typical causes of management problems, organizational inflexibility, that are pointed out especially in Japanese organizations.

4.2 Generalists-Model

We assume all agents are generalists in this variation. They are all equipped with a general knowledge that can be contradictory to each other. Knowledge can be supplemented using a common word, even if the meaning of the word is not accurately the same.

For solving a simple cup concept learning problem organizationally by three generalist-agents, each agent knows general aspects of a cup and cooperates to distinguish the given object to be a cup or not. Three agents have three subgoals in reasoning to produce the result respectively, and communicate not only the reasoning result of the goal but also the reasoning results of subgoals (intermediate results). The shared subgoals are "liftable," "graspable" and "stable." Agents can share the result of subgoal when the subgoal has the same name.

They continue cooperating and solving the problems by learning, even if there remains a little contradiction between them. For example, if an agent fails and it finds that his intermediately reasoned result about "stable" is sometimes wrong, it utilizes other agents' reasoning about "stable" by operating AND or OR to its own reasoning the next time it decides. Decision as a whole is made by simple majority rule in this simulation, because the number of agents is only three. Rules that three agents have are as follows:

Agent-1 (As his favorite "cup" is beer-jug, his concept of a cup is a little bit strange.)

- stable $\wedge$ liftable $\wedge$ open-vessel $\Rightarrow$ cup
- has-handle $\wedge$ width-small $\Rightarrow$ liftable
- has-bottom $\wedge$ medium-weight $\Rightarrow$ stable

Agent-2 (As he always sips drinks without lifting a cup, he lacks sub-concept "liftable" of a cup.)

- stable $\wedge$ graspable $\wedge$ open-vessel $\Rightarrow$ cup
- has-handle $\Rightarrow$ graspable
- has-bottom $\wedge$ flat-bottom $\Rightarrow$ stable

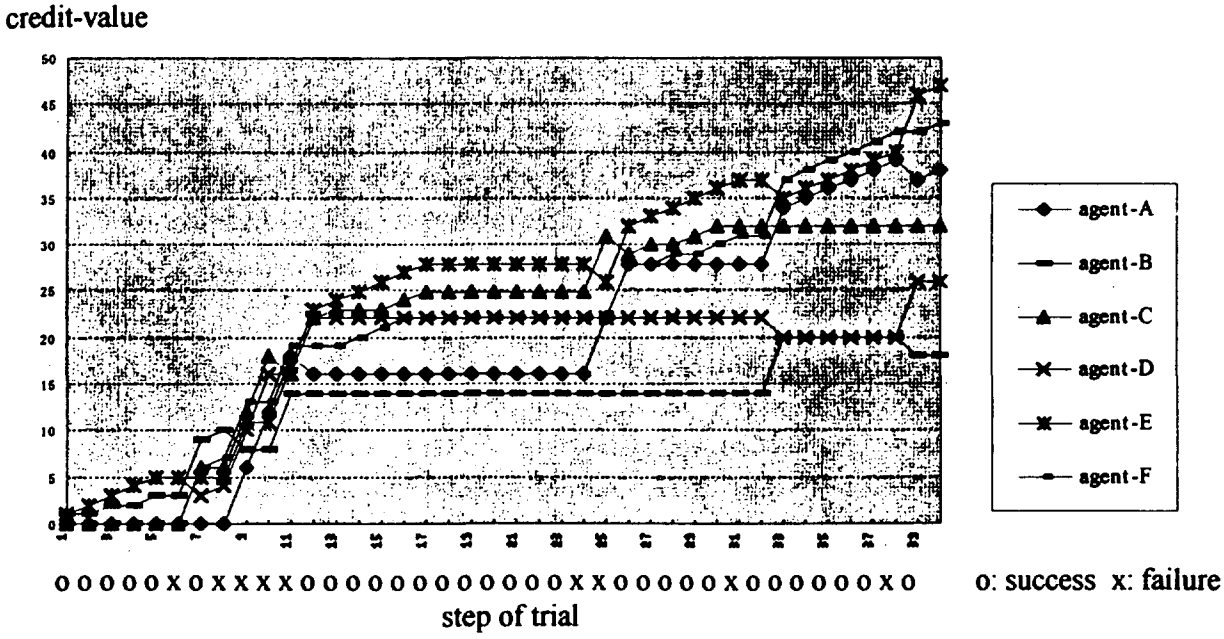


Figure 2: Result of the Specialists-Model simulation

Agent-3 (As he always uses a cup chained to a cock, he lacks sub-concept "stable" of a cup.)

- graspable $\wedge$ liftable $\wedge$ open-vessel $\Rightarrow$ cup
- lightweight $\Rightarrow$ liftable
- width-small $\wedge$ ceramic $\Rightarrow$ graspable

The situation is similar to intellectual activities in a simple organization that consists of generalists or generalists-groups. For example, when we reason whether the sales of a certain product shall be good or poor, it depends on business conditions. The method of judging whether business conditions are good or bad differs by individuals. One person judges from the economic indicator, another from sensing a customer's motivation to purchase. When a person reasons whether the sales of a certain product shall be good or poor, he can supplement his reasoning from other's reasoning about business conditions. The Generalists-Model represents such a situation. In the Generalists-Model, a subgoal (business conditions, in this example) can be shared among agents and agents cooperate to solve the entire problem (prospect of sales of a certain product, in this example).

The simulation is simpler than the Specialists-Model. In the simulation of the Generalists-Model, a single type of learning, that is, learning complementary relation from past examples, is implemented. Nevertheless, this simulation also shows adaptive behaviors. Figure 3 illustrates the organizational adaptation to the two groups of examples (group-A and group-B) that require different ways to supplement knowledge from the result of subgoal reasoned by others. When the group of examples changes, the organization fails once and adapts to the new group of examples (i.e. new environment).

5 Discussion

The model and these simulations suggest the important insight into the dynamics of multiagent learning. As for the accuracy of learning, the Generalists-Model can learn more accurately, because agents try to understand others' knowledge deeply. The Generalists-Model will be typically useful, when heterogeneous knowledge-based systems of common domain are to be integrated. On the other hand, the Specialists-Model will be useful, when knowledge-based systems of heterogeneous (specialized) domain are to be integrated.

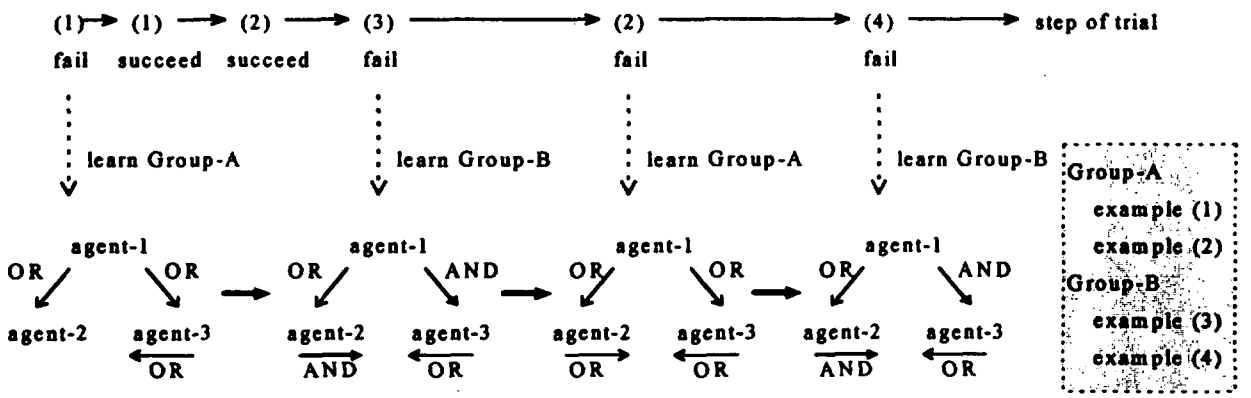


Figure 3: Result of the Generalists-Model simulation

5.1 Emergence of adaptive behaviors

In these simulations, adaptive learning as a whole is attempted by a variety of aspects, as agents reason and learn respectively. Moreover, since decision-maker is selected according to the maximum credit-values, the multiagent system can decide most appropriately to the external environment at each time.

Within an agent, knowledge is trustable, especially when success. Therefore, leaning using knowledge compilation is effective when success. The learning method within each agent can be any type of EBL-type learning, including Soar system (Laird, Newell & Rosenbloom 1987). On the other hand, when failure, each agent must consider to use other agents' knowledge. Because knowledge-sharing among agents is not very reliable, learning complementary relation from the context of the activities or from past examples is effective when failure. The result of the simulation explains the characteristic that if a multiagent system has an agent that has somewhat different knowledge from the other agents, then the system can adapt easily to the change of the environment.

Methods of reasoning and learning in these simulations are very simple. However, these methods can be refined as necessary to apply specific problems, and each agent system may be huge knowledge-based system. Therefore, the approach proposed in this paper can scale up to much larger and more complex problem domain.

5.2 Methods for knowledge sharing

Shared-ontologies approach is very accurate. However, it is very difficult to extract shared ontologies among agents' knowledge, and facilitators must be omniscient. The researchers in the methods of shared ontologies adopt knowledge engineering approach to focus on accurate mapping between different knowledge representation. On the other hand, we adopt organizational learning approach, or knowledge management approach (Hatakama & Terano 1996), to focus on how to maintain ongoing knowledge sharing pragmatically.

In this paper, we use accurately same vocabulary to be communicated between agents in the simulations. However, the proposing model will be able to be extended easily to more loose knowledge-sharing. By the Generalists-Model, we have simply investigated how common ontologies can be learned by agents. The model can be extended to sharing words that have close meaning when used in the context of activities by individuals or departments who have disparate knowledge. We have studied the framework in which agents can learn and unlearn such a relationship of vocabulary adaptively, based on multiagent model. Our model can also include human-beings whose knowledge is fairly epistemological.

6 Conclusion

This paper has proposed a knowledge intensive learning model in a multiagent system, and examines the validity and feasibility of the model in order to apply it to distributed heterogeneous knowledge systems. We have implemented two typical variations of the model: the Specialists-Model and the Generalists-Model. Using the two variations, we have carried out simulation of the dynamic behaviors of decision making and learning of multiagent system on a cup concept learning. The experiments showed that the model had unique adaptability to the change of environments, by enabling agents to efficiently supplement knowledge each other as they cooperatively makes decision and learns in the context of activities.

References

[1] Aiba H. & Terano T. (1996) Computational Model for Distributed Knowledge Systems with Learning Mechanisms. *Expert Systems with Applications*, 10, 3/4, p. 417-427.

[2] Anderson J. R. (1983) *The Architecture of Cognition.* Harvard University Press.

[3] Bond A. & Gasser L. Eds. (1988) *Reading in Distributed Artificial Intelligence.* Morgan Kaufmann.

[4] Carley K. et al. (1992) Plural-Soar: A Prolegomenon to Artificial Agent and Organizational Behavior. in Masuch M. & Wargelien M. Eds. *Artificial Intelligence in Organization and Management Theory*, p. 87-116.

[5] Gasser L. (1993) Social Knowledge and Social Action: Heterogeneity in Practice. *Proceedings of 13th Int. Joint. Conf. on AI*, p. 751-757.

[6] Hatakama H. & Terano T. (1996) A Multiagent Model of Organizational Intellectual Activities for Knowledge Management. in Schreinemakers J. F. Eds. *Knowledge Management: Organization, Competence and Methodology*, Wurzburg: Ergon Verlag.

[7] Laird J. E., Newell A. & Rosenbloom P. S. (1987) Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33, p. 1-64.

[8] Levitt B. & March J. G. (1988) Organizational Learning, *Annual Review of Sociology*, 14, p. 319-340.

[9] Masuch M. & Wargelien M. Eds. (1992) *Artificial Intelligence in Organization and Management Theory.* Elsevier Science Publishers B.V.

[10] Minsky M. (1985) *The Society of Mind.* New York: Simon and Schuster.

[11] Neches R. et al. (1991) Enabling Technology for Knowledge Sharing. *AI Magazine*, 12, 3, p. 36-56.

[12] Schreiber G. et al. (1994) CommonKADS: A Comprehensive Methodology for KBS Development. *IEEE Expert*, 9, 6, p. 28-37.

[13] Schwartz D. G. (1995) *Cooperative Heterogeneous Systems.* Kluwer Academic Publications.

[14] Senge P. M. (1990) *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York: Doubleday/ Currency.

[15] Simon H. A. (1976) *Administrative Behavior, 3rd ed.* New York: Free Press.

[16] Terano T. et al. (1994) A Machine Learning Model for Analyzing Performance of Organizational Behaviors of Agents. *Proc. of the Third Conference of the Association of Asian-Pacific Operational Research Societies (APORS)*, p. 164-171.

[17] Terano T. & Oikawa S. (1996) Genetic Algorithm-Based Feature Selection in Multiple Inductive Learning Agents. *Proceedings of 4th Int. Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery*, p. 347-352.

[18] Terano T. (1997) Towards Design and Analysis of Organizational Intelligence through Learning Multiagent Systems. *Proceedings of PACIS'97 (The Pacific Asia Conference on Information Systems)*, p. 827-834.

A Case Study in Non-monotonic Reasoning: an Alternative Analysis of the Yale Shooting Problem

Yong Sun
Department of Computer Science
The Queen's University of Belfast
Belfast BT7 1NN, Northern Ireland
Tel: +44-1232-2746565; Fax: +44-1232-683890
E-mail: Y.Sun@qub.ac.uk; http://www.cs.qub.ac.uk/~Y.Sun

Keywords: non-monotonic reasoning, Yale shooting problem

Edited by: Se Woo Cheon
Received: May 15, 1997 **Revised:** May 24, 1997 **Accepted:** June 6, 1997

Non-monotonic reasoning is often used in daily life, e.g. lacking evidence to the contrary of β we infer β. This kind of logic has its application in queries to knowledge bases, very large databases and deductive bases. This case study uses the "Yale Shooting" problem to present non-monotonic reasoning from a proof-theoretic standpoint. Basic knowledge of first order logic and many-sorted algebras is assumed.

1 Introduction

Non-monotonic reasoning (Davis 1980; McCarthy 1980; McDermott & Doyle 1980; Reiter 1980) extends the reasoning of first order logic in a way that captures the meanings of *"lacking evidence to the contrary of β, infer γ from α"*, or *"infer γ from α and the inability to infer $\neg\beta$"*, or *"from α and the inability to prove the negation of β, we infer γ"*. The last form is referred to as *default rules* by Reiter (1980), expressed as:

$$\frac{\alpha : \mathbf{M}\beta}{\gamma}$$

or more meaningfully as:

$$\frac{\Gamma \vdash \alpha; \Gamma \nvdash \neg\beta}{\Gamma \vdash \gamma}.$$

We will describe what non-monotonic reasoning is using a simple and often-quoted example. For instance, people often accept as a general truth the proposition

$$\mathbf{bird}(x) \Rightarrow \mathbf{fly}(x). \quad (1.1)$$

However, on reflection, they will admit that the proposition is not universally valid. For instance, a **penguin** is a **bird** but cannot **fly**. Thus, we need to modify (1.1) to

$$\mathbf{bird}(x) \textbf{ and } \neg\mathbf{penguin}(x) \Rightarrow \mathbf{fly}(x). \quad (1.2)$$

Additionally, an **ostrich** is a **bird** which cannot **fly** either. So, we need to modify (1.2) to

$$(\mathbf{bird}(x) \textbf{ and } \neg\mathbf{penguin}(x) \textbf{ and } \neg\mathbf{ostrich}(x)) \Rightarrow \mathbf{fly}(x). \quad (1.3)$$

This type of the reasoning process is unending. It has the following interesting characteristic. When we have the evidences in Γ, we may infer γ, written as $\Gamma \vdash \gamma$; but when we discover new evidences in Γ', we may not be able to infer γ, i.e. $\Gamma \cup \Gamma' \nvdash \gamma$. This pattern of reasoning is called *"non-monotonic"* reasoning.

Non-monotonic reasoning is quite widely used in answering queries in knowledge bases, very large databases and/or deductive bases. This paper demonstrates the sorts of problems associated with non-monotonic reasoning by means of a simple example, i.e. the Yale shooting problem (Hanks & McDermott 1985, 1986 & 1987). Unlike some reported research in which "frame axioms" are given as the source of the problems (Hanks & McDermott 1986 & 1987) we locate them elsewhere, and provide our own solutions, as a Chinese proverb says: *studying the old to discover the new*. This will become clearer in Section 5.

This paper is organized as follows. Section 2 introduces a Simplified Yale Shooting problem. Its relationship with the full version of Yale Shooting problem is given in Section 5 along with a summary of the results in this paper. The deduction trees for the Yale Shooting problem are given in Sectioon 3. An analysis of potential solutions is given in Section 4.

2 Simplified Yale shooting problem

The "Yale Shooting" problem is an example used to demonstrate the problems faced in the formalization of non-monotonic reasoning (Hanks & McDermott 1985, 1986 & 1987). The following is a brief description of the problem.

Yale Shooting Problem 1: *In a certain known situation* $\mathbf{s}_0$, *a person is* **alive**. *A gun is* **loaded** *whenever a* **load** *event happens. When the person is shot with a loaded gun, he/she becomes* **dead**; *furthermore, being shot with a loaded gun is* **abnormal** *with respect to staying* **alive** *for that person. In addition, "normal" facts persist across the occurrence of "normal" events.*

Let us introduce the problem formally as follows. Suppose that:

(2.a) $\mathbf{true}(f, s)$ means that (fact) f is *true* in situation s;

(2.b) **result** is a function with signature

$$\mathbf{result} : \mathit{event} \times \mathit{situation} \rightarrow \mathit{situation}$$

and $\mathbf{result}(e, s)$ is the *result* situation of event e occurring in situation s;

(2.c) $\mathbf{ab}(f, e, s)$ means that fact f is *abnormal* with respect to event e occurring in situation s;

(2.d) there are constant events **load** and **shoot**;

(2.e) **alive, dead** and **loaded** are constant facts;

(2.f) constant situations include

(2.f.i) the initial situation $\mathbf{s}_0$,

(2.f.ii) $\mathbf{s}_1 = \mathbf{result}(\mathbf{load}, \mathbf{s}_0)$, and

(2.f.iii) $\mathbf{s}_2 = \mathbf{result}(\mathbf{shoot}, \mathbf{s}_1)$

$$\mathbf{s}_0 \xrightarrow{\mathbf{load}} \mathbf{s}_1 \xrightarrow{\mathbf{shoot}} \mathbf{s}_2.$$

Axioms *YSh* and inference rules $\mathcal{D}$ of Yale Shooting 2: *The set YSh of axioms for the Yale Shooting problem contains the following:*

(2.1) $\mathbf{true}(\mathbf{alive}, \mathbf{s}_0)$

where $\mathbf{s}_0$ *is the initial situation - intuitively, axiom (2.1) states that the fact* **alive** *is* true *in the initial situation* $\mathbf{s}_0$*;*

(2.2) $\forall s.\mathbf{true}(\mathbf{loaded}, \mathbf{result}(\mathbf{load}, s))$,

intuitively, axiom (2.2) says that the fact **loaded** *is* true *in the situation* s' *as a result of the event* **load** *occurring in the situation* s *where* $s' = \mathbf{result}(\mathbf{load}, s)$*;*

(2.3) $\forall s.\mathbf{true}(\mathbf{loaded}, s) \Rightarrow \mathbf{true}(\mathbf{dead}, \mathbf{result}(\mathbf{shoot}, s))$,

intuitively, axiom (2.3) says that if the fact **loaded** *is* true *in the situation* s *then the fact* **dead** *is* true *in the situation* $s' = \mathbf{result}(\mathbf{shoot}, s)$*;*

(2.4) $\forall f.\forall e.\forall s.[\mathbf{true}(f, s) : \mathbf{M}\neg\mathbf{ab}(f, e, s)] \Rightarrow \mathbf{true}(f, \mathbf{result}(e, s))$,

intuitively, axiom (2.4) says that if the fact f *is* true *in the situation* s *and the negation of* $\neg\mathbf{ab}(f, e, s)$ *is not derivable (i.e.* $\mathbf{ab}(f, e, s)$ *is not derivable), then the fact* f *is preserved in the situation* s' *by the event* e *occurring in the situation* s *where* $s' = \mathbf{result}(e, s)$*;*

The inference rules $\mathcal{D}$ for YSh contain the inference rules of first order logic except for the weakening (wkn) rule, i.e.

$$(wkn) \quad \frac{\Gamma \vdash \gamma}{\Gamma \cup \Gamma' \vdash \gamma},$$

and has an extra default modus ponens rule

$$(deMP) \quad \frac{\alpha, \alpha : \mathbf{M}\beta \Rightarrow \gamma}{\gamma}$$

provided that $\neg\beta$ *is not derivable.*

Note that the exclusion of the weakening rule in $\mathcal{D}$ means that such a logic is *non-monotonic*. The interesting part of *YSh* is axiom (2.4), which is referred to as a *"frame axiom"* or *"default axiom"* in the literature. Assume that YSh^- is the same

as *YSh* with the exception that the axiom (2.4) is deleted. In this case we do not need to include the (deMP) rule, and $[\mathcal{D} - (deMP)]$ and $\mathcal{D}$ are equivalent. Thus, $[YSh^{-} \cup \mathcal{D}]$ is an "ordinary" deduction system.

3 Contradiction?

Now, we give the deduction trees for the Yale Shooting problem and establish that:

(3.a) **true(dead**, s_2) is derivable in $[YSh \cup \mathcal{D}]$,

(3.b) **true(alive**, s_2) is derivable in $[YSh \cup \mathcal{D}]$,

(3.c) **true(dead**, s_2) is derivable in $[YSh^{-} \cup \mathcal{D}]$ (i.e. $YSh^{-} \cup [\mathcal{D} - (deMP)]$) but **true(alive**, s_2) is not.

Thus, from (3.a) and (3.b) we seem to arrive at a *"contradiction"* in $[YSh \cup \mathcal{D}]$. ¿From (3.c) we conclude that this "contradiction" is a particular problem associated with *non-monotonic* reasoning.

(3.i)

$$\frac{\begin{array}{ll}\mathbf{true(loaded, s_1)}, & (2.2)\\ \mathbf{true(loaded, s_1)} \Rightarrow \mathbf{true(dead, s_2)} & (2.3)\end{array}}{\mathbf{true(dead, s_2)}}\ (\text{modus ponens})$$

(3.ii)

$$\frac{\begin{array}{ll}\mathbf{true(alive, s_0)}, & (2.1)\\ \mathbf{true(alive, s_0)} : \mathbf{M\neg ab(alive, load, s_0)} \Rightarrow \mathbf{true(alive, s_1)} & (2.4)\end{array}}{\begin{array}{l}\mathbf{true(alive, s_1)},\\ \mathbf{true(alive, s_1)} : \mathbf{M\neg ab(alive, shoot, s_1)} \Rightarrow \mathbf{true(alive, s_2)}\end{array}}\ \begin{array}{l}(\text{deMP})\\ (2.4)\end{array}$$

$$\frac{}{\mathbf{true(alive, s_2)}}\ (\text{deMP})$$

provided that **ab(alive, load, s_0)** and **ab(alive, shoot, s_1)** are not derivable in $[YSh \cup \mathcal{D}]$.

(3.iii) Looking at the derivation tree for **true(dead**, s_2), we observe that axiom (2.4) is not used. Thus, the same derivation holds in $[YSh^{-} \cup \mathcal{D}]$. To derive **true(alive**, s_2) we know that the use of axiom (2.1) is the only way to have the fact **alive** in **true(alive**, s_2) appearing in a derivation tree of $[YSh^{-} \cup \mathcal{D}]$.

4 Potential solutions

In (3.i) and (3.ii) (in Section 3) we appear to have a contradiction, i.e. we only expect one of them to hold but not both. In other words, we expect that the following hold:

(4.A) $\forall s.$**true(dead**, s) **or true(alive**, s), and

(4.B)$\forall s.\neg$(**true(dead**, s) **and true(alive**, s)).

Let us carefully examine the deduction trees (3.i) and (3.ii) above. The key is the meaning of **alive** and **dead**. The problem arises because we use *our intuitive interpretation* of the trees rather than the one provided by *YSh*. Indeed, there is no connection between the fact **alive** and the fact **dead** in *YSh*. So, the *"contradiction"* is not a contradiction in *YSh*; rather, it is an *illusion* arising from our intuition.

To put this another way, *YSh* is an *under-specification* of our intuitive interpretation. That is, not enough information of the intuitive interpretation of the Yale shooting problem is captured by *YSh*. Thus, no solution is called for (i.e. there is no contradiction) unless we want the deduction

system to capture more fully this intuitive meaning.

There are a number of ways to capture more on this matter (i.e. to augment the system $[YSh \cup \mathcal{D}]$). Suppose that we introduce (4.A) and (4.B) into YSh, then we will have a *true* contradiction in $[YSh + (4.A) + (4.B)] \cup \mathcal{D}$, since by (3.i) and (3.ii) with the (and-introduction) rule (from propositional/predicate logic) we would have that **true(dead**, s_2) **and true(alive**, s_2) is derivable in $YSh \cup \mathcal{D}$, which contradicts (4.B). Note that, from (4.A) and (4.B), we can have:

$$(4.AB) \quad \forall s.\neg \mathbf{true}(\mathbf{dead}, s) \Longleftrightarrow \mathbf{true}(\mathbf{alive}, s).$$

This contradiction arises from the fact that the abnormal predicate **ab** is not explicitly related to the constant facts **dead** and **alive** in $[YSh + (4.A) + (4.B)]$.

Now, let us pursue further the quest to capture the intuitive meaning. There are two "obvious" axioms connecting **ab** with **dead** and **alive**. They are given by (4.C) and (4.D) below. We can only choose one (since they are inconsistent).

$$(4.C) \quad \forall s.\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, s) \Longleftrightarrow \mathbf{true}(\mathbf{dead}, \mathbf{result}(\mathbf{shoot}, s)),$$

i.e. the fact that **dead** holds in situation s' (a result of an event **shoot** occurring in a situation s) is equivalent to the fact **alive** being *abnormal* if the event **shoot** occurs in s where $s' = \mathbf{result}(\mathbf{shoot}, s)$.

$$(4.D) \quad \forall s.\neg\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, s) \Longleftrightarrow \mathbf{true}(\mathbf{dead}, \mathbf{result}(\mathbf{shoot}, s)),$$

i.e. the fact **alive** is not *abnormal* if an event **shoot** occurs in a situation s; this is equivalent to the fact **dead** being *true* in the situation s' as a result of the event **shoot** occurring in the situation s where $s' = \mathbf{result}(\mathbf{shoot}, s)$.

If we introduce (4.D) into $[YSh + (4.A) + (4.B)]$, then we still have an *inconsistent* deduction system. The reason is as follows: By (4.A), (4.B) and (4.D) we would have that $(4.D')$ is derivable in $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$ where $(4.D')$ is

$$\forall s.\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, s) \Longleftrightarrow \mathbf{true}(\mathbf{alive}, \mathbf{result}(\mathbf{shoot}, s)).$$

By (3.i) and (4.D), we have that $\neg\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, s_1)$ is derivable in $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$. Therefore, $\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, \mathbf{s}_1)$ is not derivable in $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$. Also, $\mathbf{ab}(\mathbf{alive}, \mathbf{load}, \mathbf{s}_0)$ is not derivable in $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$. So, derivation (3.ii) is valid. Hence, $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$ is *not* consistent. However, our intuition tells us that the source of this inconsistency lies in (4.D) and not elsewhere.

On the other hand, if the axiom (4.C) is chosen for inclusion in $[YSh + (4.A) + (4.B)]$ rather than the axiom (4.D), then we would have that (3.i) but not (3.ii) is derivable in $[YSh + (4.A) + (4.B) + (4.C)] \cup \mathcal{D}$. So, the deduction system $[YSh + (4.A) + (4.B) + (4.C)] \cup \mathcal{D}$ is *consistent.* The reason for this is simply that the condition $\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, \mathbf{s}_1)$ is not derivable, which is no longer valid in $[YSh + (4.A) + (4.B) + (4.C)] \cup \mathcal{D}$.

¿From the above analysis, we conclude that a proper specification for a deduction system is crucial.

5 Summary and Remarks

In (Hanks & McDermott 1985, 1986 & 1987), Hanks and McDermott observed that the *"frame axiom"* (2.4) creates the problem for them. They developed their theory to solve the problem along that line. In this paper, we observe differently so that we provide an alternative analysis of the Yale shooting problem as a case study in non-monotonic reasoning. In summary, we have discussed five deduction systems and we draw the following conclusions.

5.1. $YSh^{-} \cup \mathcal{D}$ [or $(YSh - (2.4)) \cup \mathcal{D}$] is a consistent but ordinary system;

5.2. $YSh \cup \mathcal{D}$ is consistent but underspecified;

5.3. $[YSh + (4.A) + (4.B)] \cup \mathcal{D}$ is inconsistent;

5.4. $[YSh + (4.A) + (4.B) + (4.C)] \cup \mathcal{D}$ is consistent; and

5.5. $[YSh + (4.A) + (4.B) + (4.D)] \cup \mathcal{D}$ is inconsistent.

Because of non-monotonicity, we can have 5.4 from 5.3. However, if we are not careful, we would still have an inconsistent theory such as 5.5 from 5.3.

The differences among these five are very interesting. However, the means of generating the five systems automatically is not clear, since it seems to rely on human creativity. We believe that the analysis given in this paper is a step in the direction of understanding the nature of this creativity.

We should point out that the original Yale Shooting $[YSh^+ \cup \mathcal{D}]$ in (Hanks & McDermott 1985, 1986 & 1987) is somewhat more complicated than YSh in this paper.

The signature of YSh^+ has an additional constant event **wait** and an additional constant situation $\mathbf{s}_3$ where $\mathbf{s}_1 = \mathbf{result}(\mathbf{load}, \mathbf{s}_0)$, $\mathbf{s}_2 = \mathbf{result}(\mathbf{wait}, \mathbf{s}_1)$ and $\mathbf{s}_3 = \mathbf{result}(\mathbf{shoot}, \mathbf{s}_2)$; i.e.

$$\mathbf{s}_0 \xrightarrow{\mathbf{load}} \mathbf{s}_1 \xrightarrow{\mathbf{wait}} \mathbf{s}_2 \xrightarrow{\mathbf{shoot}} \mathbf{s}_3.$$

YSh^+ is the same as YSh with the following exceptions:

(5.a) The axiom (2.3) is replaced by

(5.3^+)
$\forall s.\mathbf{true}(\mathbf{loaded}, s) \Rightarrow$
$\mathbf{ab}(\mathbf{alive}, \mathbf{shoot}, s)$ **and** $\mathbf{true}(\mathbf{dead}, s')$

where $s' = \mathbf{result}(\mathbf{shoot}, s)$,

(5.b) The axiom (2.4) is replaced by

(5.4^+)
$\forall f.\forall e.\forall s.\mathbf{true}(f, s)$ **and** $\neg\mathbf{ab}(f, e, s) \Rightarrow$
$\mathbf{true}(f, \mathbf{result}(e, s))$,

and

(5.c) An extra Normal Default axiom (NDe) is added

$$(NDe) \quad [:\mathbf{M}\neg\mathbf{ab}(f, e, s)] \Rightarrow \neg\mathbf{ab}(f, e, s).$$

So, (5.3^+) implies (2.3); and $[(5.4^+) + (NDe) \cup (\Rightarrow MP)]$ implies (2.4), where $(\Rightarrow MP)$ is

$$(\Rightarrow MP) \quad \frac{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}$$

and α **and** $[:\mathbf{M}\beta] \Rightarrow \gamma$ is taken to be $[\alpha : \mathbf{M}\beta] \Rightarrow \gamma$.

However, the analysis given in Section 3 and Section 4 will be essentially the same for the original $[YSh^+ \cup \mathcal{D}]$. The interested reader is invited to conduct a similar analysis for $[YSh^+ \cup \mathcal{D}]$.

By the way, Apt and Bezem give their solution to the original Yale Shooting problem by acyclic programs in (Apt & Bezem 1991).

Lastly, all results in this paper are presented from a proof-theoretic point of view. It would be interesting to see the results presented from a set-theoretic point of view.

This paper is an enhanced version of (Sun & Fang 1997) and a revised version of (Sun 1997).

6 Acknowledgements

The author would like to thank M. Fang for sharing his insight into the non-monotonic reasoning, M. Clint, Se Woo Cheon and A. Stewart for their constructive comments on the early versions of this paper, and N. D. N. Measor for his encouragement.

References

[1] Krzysztof R. Apt and Marc Bezem. (1991) Acyclic Programs. *New Generation Computing*, 9, p.335-363.

[2] Martin Davis. (1980) The mathematics of non-monotonic reasoning. *Artificial Intelligence*, 13, p.73-80.

[3] Steven Hanks and Drew McDermott. (1985) Temporal reasoning and default logics. Computer Science Research Report No.430, Yale Univeristy, USA.

[4] Steven Hanks and Drew McDermott. (1986) Default Reasoning, non-monotonic logics, and the frame problem. in *the proceedings of AAAI-86*, American Association for Artificial Intelligence, Philadelphia, USA, p.328-333.

[5] Steven Hanks and Drew McDermott. (1987) Non-monotonic logics, and temporal projection. *Artificial Intelligence*, 33, p.379-412.

[6] John McCarthy. (1980) Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13, p.27-39.

[7] Drew V. McDermott and Jon Doyle. (1980) Non-monotonic Logic I. *Artificial Intelligence*, 13, p.41-72.

[8] Raymond Reiter. (1980) A logic for default reasoning. *Artificial Intelligence*, 13, p.81-132.

[9] Yong Sun. The Yale Shooting Problem: A Case Study in Non-Monotonic Reasoning. in the *proceedings of the Joint Pacific Asian Conference on Expert Systems and Singapore International Conference on Intelligent Systems*, eds. Dan Patterson, et al., Nanyang Technological University Press, pp.415-422, Singapore, 24-27 February 1997.

[10] Yong Sun and Ming Fang. (1997) The Yale Shooting Problem: A Case Study in Non-Monotonic Reasoning. to appear in the *Journal of Peking University* (Natural Science Section, Bimonthly), i.e. Acta Scientiarum Naturalium Universitatis Pekinensis, vol. 33, Peking University Press, Beijing, China.

The Weighting Issue in Fuzzy Logic

Xudong Luo and Chengqi Zhang
Department of Mathematics, Statistics and Computing Science
The University of New England, Armidale, NSW 2351, Australia
Phone: +61 67 73 {3574, 2350}, Fax: +61 67 73 3312
E-mail: {xluo, chengqi}@neumann.une.edu.au

Jingqiu Cai
Department of Computer Science, Xiamen University, Fujian 361005, China

Keywords: fuzzy logic, relative weighted models, uncertainty, weight, triangular norms, triangular conorms

Edited by: Se Woo Cheon

Received: November 12, 1996 **Revised:** March 26, 1997 **Accepted:** June 6, 1997

The objective of this paper is to establish a class of models for solving the weight problem in fuzzy logic. First, some constraints that weighted fuzzy logic should satisfy are given based on Triangular norms and conorms. Then, based on these constraints, a class of models (referred to as relative weighted models) are established for handling weights in fuzzy logic. These models are novel in three aspects: (1) they include non-weighted models as their special cases, (2) the weighted conjunction and weighted disjunction can be distinguished from each other, and (3) the information from all sub-propositions can be sufficiently considered. In addition, this paper proposes principles for selecting proper weighted models.

1 Introduction

When a proposition is made up of multiple sub-propositions, the different sub-propositions are usually assumed to carry equal importance in traditional knowledge-based systems. However, this is not true when experts are interviewed [4, 8, 9, 12], or the decision process is analyzed [6, 10, 14]. In fact, different sub-propositions may have different (even quite different) influences on a composite proposition. Such a case is often encountered in practical applications. In order to take account of different effects, a weight can be assigned to each sub-proposition. The more a sub-proposition effects the composite proposition, the larger is the weight assigned to the sub-proposition. The problem here is how to calculate the truth value of the composite proposition using the truth value of every sub-proposition and its corresponding weight. In this paper, we discuss this issue in the context of fuzzy logic.

Some researchers [5, 14] have tried to exploit *the weighted averaging model* to cope with the weight problem in fuzzy logic. The weighted averaging model can be described as follows: Suppose that there are n sub-propositions. Let A_i denote a sub-proposition of the composite proposition A, $T(A_i) \in [0,1]$ denote the fuzzy truth value of A_i, and w_i denote the weight of A_i, which satisfies $\sum_{i=1}^{n} w_i = 1$. If a composite proposition in the form of weighted conjunction is

$$A = A_1(w_1) \wedge \cdots \wedge A_n(w_n) \qquad (1)$$

then the truth value $T(A)$ of A is given by

$$T(A) = \sum_{i=1}^{n} w_i \times T(A_i). \qquad (2)$$

However, this weighted averaging model bears two main limitations which we will address below.

1) *Degeneration problem.* Intuitively, a non-weighted conjunction should be a special case of a weighted conjunction. This is the same for a non-weighted disjunction and weighted disjunction. In other words, when every sub-proposition

is regarded equivalently, that is, $w_1 = \cdots = w_n$, a weighted fuzzy logic should degenerate into a non-weighted fuzzy logic. For a non-weighted conjunction $A = A_1 \wedge \cdots \wedge A_n$, its fuzzy truth value $T(A)$ is given by [15]:

$$T(A) = \min\{T(A_1), \cdots, T(A_n)\} \tag{3}$$

On the other hand, in the weighted averaging model, when $w_1 = \cdots = w_n = \frac{1}{n}$, formula (2) becomes:

$$T(A) = \frac{\sum_{i=1}^{n} T(A_i)}{n} \tag{4}$$

Obviously, in most cases, formula (4) would not be equivalent to formula (3). In other words, using the weighted averaging model to deal with the weight problem in fuzzy logic is not compatible with the non-weighted case. We argue that such a use is inconsistent with fuzzy logic as well as with classical logic.

2) *Differential problem.* If we have a weighted disjunction

$$B = A_1(w_1) \vee \cdots \vee A_n(w_n) \tag{5}$$

then by DeMorgan's law, we should have

$$B = \neg((\neg A_1)(w_1) \wedge \cdots \wedge (\neg A_n)(w_n))$$

Thus, if we use the weighted averaging model, i.e. formula (2), we can obtain

$$T(B) = 1 - \sum_{i=1}^{n} w_i \times (1 - T(A_i)) = \sum_{i=1}^{n} w_i \times T(A_i) \tag{6}$$

A comparison between formulas (2) and (6) shows that there is no difference between weighted conjunction and weighted disjunction. Clearly, this is against the principles of fuzzy logic and classical logic. This demonstrates again that the weighted averaging model is not appropriate to handle weights in fuzzy logic.

Therefore, other models should be developed for processing weights in fuzzy logic. These new models should have the following characteristics:

1. in the situation where all weights are equal to each other, the model should degenerate into a non-weighted model;
2. the model should distinguish a weighted conjunction from a weighted disjunction; and
3. the fuzzy truth value of the composite proposition should comprehensively take the fuzzy truth value and weight of every sub-proposition into consideration.

The rest of this paper is organized as follows. Section 2 recaptures the basic notions of Triangular norms and Triangular conorms. In Section 3, we present the constraints on models for handling weights in fuzzy logic. Section 4 presents a class of models for handling weights in fuzzy logic, and discusses the selection problem of these models. Some examples are provided in Section 5 to illustrate the flexibility of our model. Finally, conclusions are outlined in Section 6.

2 T-norms and T-conorms

Before further discussion it is useful to recall briefly the notions of Triangular norms (T-norms) and Triangular conorms (T-conorms) in fuzzy logic. A detailed description of these can be found elsewhere [2].

In fuzzy logic without weights there are many operators dealing with conjunction and disjunction. For example:

1. Zadeh operators $(\wedge, \vee)$:

$$a \wedge b = \min\{a, b\},$$
$$a \vee b = \max\{a, b\};$$

2. Probability operators $(\bullet, \overset{\wedge}{+})$:

$$a \bullet b = a \times b,$$
$$a \overset{\wedge}{+} b = a + b - a \times b;$$

3. Einstein operators $(\overset{\bullet}{E}, \overset{+}{E})$:

$$a \overset{\bullet}{E} b = \frac{ab}{1 + (1-a)(1-b)}$$
$$a \overset{+}{E} b = \frac{a+b}{1+ab};$$

4. Boundary operators $(\odot, \oplus)$:

$$a \odot b = \max\{0, a + b - 1\},$$
$$a \oplus b = \min\{1, a + b\}.$$

Among the above operators, the pair of operators $(\wedge, \vee)$ take only minimum and maximum, so they lose too much information about the other sub-propositions. Instead, to some extent, other kinds of operators can consider information comprehensively from all sub-propositions.

In fact, the above operators are some typical pairs of T-norms and their dual T-conorms. Dubois and Prade [3] have shown that T-norms and T-conorms are the most general families of binary functions, which respectively satisfy the requirements of the conjunction and disjunction operators. The axiomatic definition of T-norms and T-conorms is as follows [3]:

Definition 1 *If the operator* $\circ : [0,1] \times [0,1] \longrightarrow [0,1]$ *satisfies the following conditions:*

1) commutativity: $a \circ b = b \circ a$;

2) associativity: $(a \circ b) \circ c = a \circ (b \circ c)$;

3) monotonicity: if $a \leq b$ *and* $c \leq d$ *then* $a \circ c \leq b \circ d$;

4) boundary: $a \circ 1 = a$,

where $a, b, c, d \in [0,1]$, *then* $\circ$ *is said to be a T-norm on* $[0,1]$, *denoted as* $\triangle$. *If* $\circ$ *satisfies 1), 2), 3) and*

4)' boundary: $a \circ 0 = a$,

then $\circ$ *is said to be a T-conorm on* $[0,1]$, *denoted as* $\triangledown$.

Although defined as two-place functions, the T-norms and T-conorms can be used respectively to process more than two sub-propositions in a composite proposition. Due to the associativity of the T-norms, it is possible to define recursively $\triangle(x_1, \cdots, x_n, x_{n+1})$ and $\triangledown(x_1, \cdots, x_n, x_{n+1})$, for $x_1, \cdots, x_{n+1} \in [0,1]$, as

$$\triangle(x_1, \ldots, x_n, x_{n+1}) = \triangle(\triangle(x_1, \ldots, x_n), x_{n+1})$$
$$\triangledown(x_1, \cdots, x_n, x_{n+1}) = \triangledown(\triangledown(x_1, \cdots, x_n), x_{n+1})$$

For the negation operation $N(x) = 1 - x$, T-norm $\triangle$ and T-conorm $\triangledown$ are duals in the sense of the following DeMorgan's Law:

$$\triangle(a,b) = N(\triangledown(N(a), N(b)))$$
$$\triangledown(a,b) = N(\triangle(N(a), N(b)))$$

Therefore, T-norms and T-conorms should be given in the form of pairs which satisfy DeMorgan's Law.

When $(\triangle, \triangledown)$ is $(\wedge, \vee)$, $(\bullet, \overset{\wedge}{+})$, $(\underset{E}{\bullet}, \underset{E}{+})$ and $(\odot, \oplus)$ respectively, let the corresponding conjunction and disjunction be denoted as $F_\wedge$ and $G_\vee$, $F_\bullet$ and $G_{\overset{\wedge}{+}}$, $F_{\underset{E}{\bullet}}$ and $G_{\underset{E}{+}}$, $F_\odot$ and $G_\oplus$, respectively. Dubois and Prade [2] gave the following order:

$$F_\odot \leq F_{\underset{E}{\bullet}} \leq F_\bullet \leq F_\wedge \leq G_\vee \leq G_{\overset{\wedge}{+}} \leq G_{\underset{E}{+}} \leq G_\oplus \tag{7}$$

3 Constraints on Weighted Fuzzy Logic

The constraints on a weighted model in fuzzy logic will be identified in this section. The proposed constraints provide a unifying framework for handling weights in fuzzy logic based on T-norms and T-conorms. In other words, these constraints can be used to guide the construction of the weighted model.

Consider the weighted conjunction in formula (1) and the weighted disjunction in formula (5). Obviously, the fuzzy truth values of A and B should be determined by functions which combine the fuzzy truth value with the corresponding weight, of each sub-proposition. Formally, we have

$$T(A) = f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n))$$
$$T(B) = f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n))$$

where $f_\wedge$ and $f_\vee$ are functions from $[0,1]^n$ to $[0,1]$, and g is a function from $[0,1]^2$ to $[0,1]$. For the negation of a weighted sub-proposition A_i with weight w_i, its truth value should also be determined by a function from $[0,1]^2$ to $[0,1]$:

$$T(\neg A_1) = f_\neg(T(A_1), w_1)$$

Now two questions arise:

1) What are the constraints on $f_\wedge$, $f_\vee$, $f_\neg$ and g?

2) How do we construct $f_\wedge$, $f_\vee$, $f_\neg$ and g?

In the following, we will answer the first question. The second question will be answered in the next section.

Constraint 1 *When* $w_1 = \cdots = w_n = \frac{1}{n}$,

$$\begin{aligned} & f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \\ = \; & T(A_1) \triangle \cdots \triangle T(A_n) \end{aligned}$$

$$\begin{aligned} & f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \\ = \; & T(A_1) \triangledown \cdots \triangledown T(A_n) \end{aligned}$$

This constraint reveals that a weighted model degenerates into a non-weighted model in fuzzy logic when each sub-proposition takes an equal weight, i.e. carries equal importance. In other words, their corresponding boundary conditions should fulfill the *AND* and *OR* operations in fuzzy logic without weights.

Constraint 2 *There are* $T(A_1), w_1, \cdots, T(A_n)$, w_n *in* $[0, 1]$ *such that*

$$\begin{aligned} & f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \\ \neq \; & f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \end{aligned}$$

This constraint reveals that the function $f_\wedge$ is not equivalent to the function $f_\vee$ since there is a difference between conjunction and disjunction.

In Section 1, we show that the weighted averaging model does not satisfy Constraints 1 and 2.

Constraint 3 *Let* $w_i = \max\{w_1, \cdots, w_n\}$, *then*

$$\begin{aligned} & T(A_i) = 0 \\ \Rightarrow \; & f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 0 \end{aligned}$$

$$\begin{aligned} & T(A_i) = 1 \\ \Rightarrow \; & f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) = 1 \end{aligned}$$

This constraint means that, in a weighted conjunction, if the most important sub-proposition is false, the conjunction proposition should be false; in a weighted disjunction, if the most important sub-proposition is true, the disjunction proposition should be true. This constraint is a direct extension of the corresponding properties in a non-weighted model in fuzzy logic.

It is easy to show that the weighted averaging model does not satisfy Constraint 3.

Constraint 4

$$\begin{aligned} & f_\wedge(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \\ = \; & 1 - f_\vee(f_\neg(T(A_1), w_1), \cdots, f_\neg(T(A_n), w_n)) \end{aligned}$$

$$\begin{aligned} & f_\vee(g(T(A_1), w_1), \cdots, g(T(A_n), w_n)) \\ = \; & 1 - f_\wedge(f_\neg(T(A_1), w_1), \cdots, f_\neg(T(A_n), w_n)) \end{aligned}$$

This constraint states that the weighted conjunction and weighted disjunction should satisfy DeMorgan's law.

The weighted averaging model can satisfy Constraint 4.

Constraint 5 $f_\neg(T(A_i), w_i) + g(T(A_i), w_i) = 1$

In the above constraint, $f_\neg(T(A_i), w_i)$ can be viewed as the truth value of $\neg A_i$, and $g(T(A_i), w_i)$ can be viewed as the truth value of A_i. So this constraint is an extension of the corresponding property in fuzzy logic as well as in classical logic.

4 Relative Weighted Models

The constraints mentioned in the previous section have shed some light on the design of a rational model for handling the weights in fuzzy logic. In this section, we will suggest a class of models which can be used to solve the weight problem in fuzzy logic in a sound and efficient way. Our models are based on T-norms and T-conorms, and satisfy the constraints listed in Section 3, thereby sidestepping the problems in the weighted averaging model.

We define the relative weighted model as follows:

Definition 2 *Let* $(\triangle, \triangledown)$ *be a pair of T-norms and T-conorms which satisfy DeMorgan's Law. And let* $w = \max\{w_1, w_2, \cdots, w_n\}$.

1. *The truth value,* $T(A)$, *of weighted conjunction (1) is given by*

$$T(A) = (\frac{w_1}{w} \times T(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T(A_n)) \tag{8}$$

2. *The truth value,* $T(B)$, *of weighted disjunction (5) is given by*

$$T(B) = (\frac{w_1}{w} \times T(A_1)) \triangledown \cdots \triangledown (\frac{w_n}{w} \times T(A_n)) \tag{9}$$

3. *The truth value, $T(\neg A_i)$, of the negation of weighted sub-proposition A_i is given by*

$$T(\neg A_i) = 1 - \frac{w_i}{w} \times T(A_i) \qquad (10)$$

The idea behind this definition is that after weighting w_i on A_i, its truth value is updated to $\frac{w_i}{w} \times T(A_i)$ from $T(A_i)$.

Theorem: This relative weighted model satisfies Constraints 1–5 in Section 3.

Proof. 1) Notice that when $w_1 = w_2 = \cdots = w_n = w = \frac{1}{n}$, $\frac{w_i}{w} = 1$, so formulas (8) and (9) satisfy Constraint 1.

2) Let $n = 2$, $T(A_1) = 0.4$, $w_1 = 0.8$, $T(A_2) = 0.7$ and $w_2 = 0.2$. Thus,

$$\begin{aligned} & w \\ = \ & \max\{w_1, w_2\} \\ = \ & \max\{0.8, 0.2\} \\ = \ & 0.8 \end{aligned}$$

$$\begin{aligned} & T(A) \\ = \ & (\frac{w_1}{w} \times T(A_1)) \triangle (\frac{w_2}{w} \times T(A_2)) \\ = \ & (\frac{0.8}{0.8} \times 0.4) \triangle (\frac{0.2}{0.8} \times 0.7) \\ = \ & 0.4 \triangle 0.175 \end{aligned}$$

$$\begin{aligned} & T(B) \\ = \ & (\frac{w_1}{w} \times T(A_1)) \triangledown (\frac{w_2}{w} \times T(A_2)) \\ = \ & (\frac{0.8}{0.8} \times 0.4) \triangledown (\frac{0.2}{0.8} \times 0.7) \\ = \ & 0.4 \triangledown 0.175 \end{aligned}$$

Clearly, $T(A) \neq T(B)$. Therefore, formulas (8) and (9) satisfy Constraint 2.

3) Let $w_i = \max\{w_1, w_2, \cdots, w_n\}$. If $T(A_i) = 0$, then

$$\begin{aligned} & T(A) \\ = \ & (\frac{w_1}{w} \times T(A_1)) \triangle \cdots \triangle (\frac{w_n}{w} \times T(A_n)) \\ = \ & (\frac{w_1}{w_i} \times T(A_1)) \triangle \cdots \triangle (\frac{w_{i-1}}{w_i} \times T(A_{i-1})) \\ & \triangle (\frac{w_i}{w_i} \times T(A_i)) \triangle (\frac{w_{i+1}}{w} \times T(A_{i+1})) \\ & \cdots \triangle (\frac{w_n}{w} \times T(A_n)) \\ = \ & (\frac{w_1}{w_i} \times T(A_1)) \triangle \cdots \triangle (\frac{w_{i-1}}{w_i} \times T(A_{i-1})) \\ & \triangle (\frac{w_i}{w_i} \times 0)) \triangle (\frac{w_{i+1}}{w} \times T(A_{i+1})) \\ & \cdots \triangle (\frac{w_n}{w} \times T(A_n)) \\ = \ & 0 \end{aligned}$$

When $T(A_i) = 1$, as long as in the above proof we replace $\triangle$ and 0 by $\triangledown$ and 1, respectively, we can get

$$\begin{aligned} & T(B) \\ = \ & (\frac{w_1}{w} \times T(A_1)) \triangledown \cdots \triangledown (\frac{w_n}{w} \times T(A_n)) \\ = \ & 1 \end{aligned}$$

So formulas (8) and (9) satisfy Constraint 3.

4) Notice that $(\triangle, \triangledown)$ is a pair of T-norms and T-conorms which satisfy DeMorgan's Law. So formulas (8), (9) and (10) satisfy Constraint 4.

5) Clearly the model satisfies Constraint 5. □

So, our model overcomes the problems associated with using the weighted averaging model for handling weights in fuzzy logic.

Note that in the above mentioned model, $\triangle$ and $\triangledown$ are any pair of T-norms and T-conorms which satisfy DeMorgan's law. Then where is the difference among models with different T-norms and T-conorms?

For this problem, we recall inequality (7). This inequality indicates that, when using $F_\wedge$ and $G_\vee$ the difference between the weighted conjunction and the weighted disjunction is minimal. The next smallest difference occurs when using $F_\bullet$ and $G_{\underset{+}{\wedge}}$, the third smallest occurs when using $F_{\underset{E}{\bullet}}$ and $G_{\underset{E}{+}}$, and the maximum difference occurs when using $F_\odot$ and $G_\oplus$. Accordingly, in practice, we can select the appropriate one according to the degree to which we need to distinguish the weighted conjunction from the weighted disjunction.

The weighted averaging model and the relative weighted model based on Zadeh operators $(\wedge, \vee)$ are both suggested based on practical experience. For this reason, although they have some limitations, they should not be abandoned. Additionally, this paper gives other alternatives of relative weighted models for processing weights in fuzzy logic. The problem here is how to choose an appropriate model from all of these models for a particular application. In general, it should obey the following principles:

1. If there is no need to distinguish the weighted conjunction from the weighted disjunction, we can select *the weighted averaging model.*

2. If there is a need to take the point of optimistic conjunction and a pessimistic disjunction view, we use *the relative weighted model based on Zadeh operators.*

3. If there is a need to consider comprehensively information from all sub-propositions, we can employ *other relative weighted models.* Furthermore, we can select an appropriate pair of T-norms and T-conorms from probability operators, Einstein operators and boundary operators according to the degree to which we need to distinguish weighted conjunction from the weighted disjunction.

In uncertain reasoning, T-conorms are also used to aggregate the degree of certainty of the (same) conclusion derived from multiple rules [1]. If these rules with the same conclusion require to be weighted in order to represent different influences on the same conclusion from different sources, we can make use of formula (9) to calculate the uncertainty of the same conclusion from these different weighted rules.

5 Examples

For examining the effect of different models, this section calculates the values of $T(A_1 \wedge A_2)$ and $T(A_1 \vee A_2)$ by using different models when $T(A_1)$ and $T(A_2)$ are assumed respectively to take on some typical values: 0, 0.4, 0.5, 0.8 and 1.

Example 1 Let $T(A_1) = 0.8, T(A_2) = 0.4, w_1 = 0.7, w_2 = 0.3$.

1) By using the weighted averaging model, we have

$$\begin{aligned} & T(A_1 \wedge A_2) \\ = \; & T(A_1 \vee A_2) \\ = \; & w_1 \times T(A_1) + w_2 \times T(A_2) \\ = \; & 0.7 \times 0.8 + 0.3 \times 0.4 \\ = \; & 0.68 \end{aligned}$$

2) By using the relative weighted model based on Zadeh operators, we have

$$\begin{aligned} & w \\ = \; & \max\{w_1, w_2\} \\ = \; & \max\{0.7, 0.3\} \\ = \; & 0.7 \end{aligned}$$

$$\begin{aligned} & T(A_1 \wedge A_2) \\ = \; & \min\{\frac{w_1}{w} \times T(A_1), \frac{w_2}{w} \times T(A_2)\} \\ = \; & \min\{\frac{0.7}{0.7} \times 0.8, \frac{0.3}{0.7} \times 0.4\} \\ = \; & 0.17 \end{aligned}$$

$$\begin{aligned} & T(A_1 \vee A_2) \\ = \; & \max\{\frac{w_1}{w} \times T(A_1), \frac{w_2}{w} \times T(A_2)\} \\ = \; & \max\{\frac{0.7}{0.7} \times 0.8, \frac{0.3}{0.7} \times 0.4\} \\ = \; & 0.80 \end{aligned}$$

3) By using the relative weighted model based on probability operators, we have

$$\begin{aligned} & T(A_1 \wedge A_2) \\ = \; & \frac{w_1}{w} \times T(A_1) \times \frac{w_2}{w} \times T(A_2) \\ = \; & \frac{0.7}{0.7} \times 0.8 \times \frac{0.3}{0.7} \times 0.4 \\ = \; & 0.14 \end{aligned}$$

$$\begin{aligned} & T(A_1 \vee A_2) \\ = \; & \frac{w_1}{w} \times T(A_1) + \frac{w_2}{w} \times T(A_2) \\ & -\frac{w_1}{w} \times T(A_1) \times \frac{w_2}{w} \times T(A_2) \\ = \; & \frac{0.7}{0.7} \times 0.8 + \frac{0.3}{0.7} \times 0.4 \\ & -\frac{0.7}{0.7} \times 0.8 \times \frac{0.3}{0.7} \times 0.4 \\ = \; & 0.83 \end{aligned}$$

4) By using the relative weighted model based on Einstein operators,

$$\begin{aligned} & T(A_1 \wedge A_2) \\ = \; & \frac{\frac{w_1}{w} \times T(A_1) \times \frac{w_2}{w} \times T(A_2)}{1 + (1 - \frac{w_1}{w} \times T(A_1))(1 - \frac{w_2}{w} \times T(A_2))} \\ = \; & \frac{\frac{0.7}{0.7} \times 0.8 \times \frac{0.3}{0.7} \times 0.4}{1 + (1 - \frac{0.7}{0.7} \times 0.8) \times (1 - \frac{0.3}{0.7} \times 0.4)} \\ = \; & 0.118 \end{aligned}$$

$$\begin{aligned} & T(A_1 \vee A_2) \\ = \; & \frac{\frac{w_1}{w} \times T(A_1) + \frac{w_2}{w} \times T(A_2)}{1 + \frac{w_1}{w} \times T(A_1) \times \frac{w_2}{w} \times T(A_2)} \end{aligned}$$

$$= \frac{\frac{0.7}{0.7} \times 0.8 + \frac{0.3}{0.7} \times 0.4}{1 + \frac{0.7}{0.7} \times 0.8 \times \frac{0.3}{0.7} \times 0.4}$$
$$= 0.85$$

5) By using the relative weighted model based on boundary operators,

$$\begin{aligned} & T(A_1 \wedge A_2) \\ = & \max\{0, \frac{w_1}{w} \times T(A_1) + \frac{w_2}{w} \times T(A_2) - 1\} \\ = & \max\{0, \frac{0.7}{0.7} \times 0.8 + \frac{0.3}{0.7} \times 0.4 - 1\} \\ = & 0 \end{aligned}$$

$$\begin{aligned} & T(A_1 \vee A_2) \\ = & \min\{1, \frac{w_1}{w} \times T(A_1) + \frac{w_2}{w} \times T(A_2)\} \\ = & \min\{1, \frac{0.7}{0.7} \times 0.8 + \frac{0.3}{0.7} \times 0.4\} \\ = & 0.97 \end{aligned}$$

In summary, we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.68	0.68
Zadeh	0.17	0.80
Probability	0.14	0.83
Einstein	0.118	0.85
Boundary	0	0.97

Obviously, these data in the above table satisfy inequality (7).

Example 2 Let $T(A_1) = 0.8, T(A_2) = 0.4, w_1 = 0.3, w_2 = 0.7$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.52	0.52
Zadeh	0.34	0.40
Probability	0.137	0.61
Einstein	0.089	0.65
Boundary	0	0.74

Example 3 Let $T(A_1) = 0.8, T(A_2) = 0.4, w_1 = 0.5, w_2 = 0.5$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.60	0.60
Zadeh	0.4	0.80
Probability	0.32	0.88
Einstein	0.286	0.91
Boundary	0.2	1

From the data in the above table, we can see that, when weights are equal to each other, the relative weighted models degenerate into the corresponding models without weights.

Example 4 Let $T(A_1) = 1, T(A_2) = 0, w_1 = 0.7, w_2 = 0.3$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.70	0.70
Zadeh	0	1
Probability	0	1
Einstein	0	1
Boundary	0	1

From the data in the above table, we can see that, for relative weighted models, if the most important sub-proposition is true, the weighted disjunction proposition is also true.

Example 5 Let $T(A_1) = 1, T(A_2) = 0, w_1 = 0.3, w_2 = 0.7$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.30	0.30
Zadeh	0	0.43
Probability	0	0.43
Einstein	0	0.43
Boundary	0	0.43

From the data in the above table, we can see that, for relative weighted models, if the most important sub-proposition is false, the weighted conjunction proposition is also false.

Example 6 Let $T(A_1) = 1, T(A_2) = 0, w_1 = 0.5, w_2 = 0.5$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.5	0.5
Zadeh	0	1
Probability	0	1
Einstein	0	1
Boundary	0	1

From Example 6, we can discover that there is a potential problem associated with the weighted averaging model. In this example, the weights are equal. So, according to classical logic, we should have $T(A_1 \wedge A_2) = 0$ and $T(A_1 \vee A_2) = 1$. But by using the weighted averaging model, we fail to obtain such results. This is a serious contradiction to classical logic. On the contrary, our relative weight models are consistent with classical logic at this point.

Example 7 Let $T(A_1) = 0.5, T(A_2) = 0.5, w_1 = 0.7, w_2 = 0.3$, then we have

	$T(A_1 \wedge A_2)$	$T(A_1 \vee A_2)$
W-Averaging	0.50	0.50
Zadeh	0.214	0.50
Probability	0.107	0.61
Einstein	0.077	0.65
Boundary	0	0.71

The results of these examples are encouraging.

6 Conclusion

The relative weighted models presented in this paper are approaches capable of coping with weights in fuzzy logic in a sound and efficient manner. These models have been applied successfully in an expert system. We think that these models can be used for real world problems in areas such as: expert systems, fuzzy logic controllers and information retrieval.

Acknowledgement

This research is supported by the large grant from the Australian Research Council (A49530850).

References

[1] Bonisssone P. P. & Decker K. S. (1986) Selection Uncertainty Calculi and Granularity: An Experiment Trading-off Precision and Complexity. *Uncertainty in Artificial Intelligence.* North-Holland, p. 217-247.

[2] Dubois D. & Prade H. (1982) A Class of Fuzzy Measures based on Triangular Norms. *International Journal of General Systems*, 8, 1.

[3] Dubois D. & Prade H. (1984) Criteria Aggregation and Ranking of Alternatives in the Framework of Fuzzy Set Theory. *TIMS/Studies in the Management Science*, 20, Elsevier Science Publishers, p. 209-240.

[4] Greco G. & Rocha A. F. (1987) The Fuzzy Logic of a Text Understanding. *Fuzzy Sets and Systems*, 3, p. 347-360.

[5] He X. (1990) Knowledge Processing and Expert Systems. National Defence Industry Publisher.

[6] Kacprzick J. (1985) Zadeh's Commonsense Knowledge. *Approximate Reasoning in Expert Systems*, Elsevier Science Publishers.

[7] Luo X. & Zhang C. (1996) A Unified Algebraic Structure for Uncertain Reasonings. *PRICAI'96: Topics in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, 1114, Springer, p. 459-470.

[8] Rocha A. F., Theoto M., Rizzo I. & Laginha M. P. R. (1989) *Proc. 3rd IFSA Congress*, Seattle, p. 480-483.

[9] Rocha A. F., Laginha M. P. R., Machado R., Sigulen D., & Ancao M. (1990) Declarative and Procedural Knowledge: Two Complimentary Tools for Expertise. *Approximate Reasoning Tools for Artificial Intelligence*, Verlag Tuv Rheinland, p.229–253.

[10] Sanchez E. (1989) Importance in Knowledge Systems. *Information Systems*, 14, 4, p. 455-464.

[11] Shi Z. & Luo X. (1992) NLPM: An Approach for Nonmonotonic Reasoning based on Linguistic Probability. *Automated Reasoning*, North-Holland, p. 107-121.

[12] Theoto M. T. & Rocha A. F. (1990) Fuzzy Belief and Text Understanding. *Proc. 3rd IFSA Congress*, Seattle, p. 552-554.

[13] Yager R. R. (1980) On a General Class of Fuzzy Connectives. *Fuzzy Sets and Systems*, 4, p. 235-242.

[14] Yager R. R. (1988) On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making. *IEEE Trans. on Systems, Man, and Cybernetics*, 18, p. 183-190.

[15] Zadeh L. A. (1985) The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. *Approximate Reasoning in Expert Systems*, Elsevier Science Publishers.

Principle-based Parsing and Logic Programming

Matthew W. Crocker
Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh, UK EH8 9LW
E-mail: mwc@cogsci.ed.ac.uk

Keywords: Natural language processing, computational linguistics, deductive parsing

Edited by: Rudi Murn
Received: August 22, 1996 **Revised:** April 11, 1997 **Accepted:** May 5, 1997

While deductive parsing techniques are well-understood for traditional rule-based and lexicalist grammars, they are rather more elusive for current principle-based grammars. In this paper, we argue that a major source of difficulty arises from a fundamental difference in the way such grammars should be axiomatised. While rule-based grammars typically consist of a set of sufficient *'structure-generating' axioms, principle-based grammars are more naturally expressed as a set of* necessary *'structure-licensing' conditions. On this basis we propose a methodology for implementing deductive parsers which is more suitable for this class of 'licensing grammars'. We then argue that current principle-based grammatical theories can be most naturally implemented by decomposing them into representationally homogeneous subsystems, which are axiomatised as licensing 'sub-grammars' – each contributing its own aspect of a global syntactic deduction system. Finally we consider the new range of options this approach offers for developing flexible and possibly distributed control regimes.*

1 Introduction

Syntactic constraints form an important source of knowledge in any natural language processing (NLP) system, be it intended for a practical application requiring deep interpretation, or as theoretical or cognitive model for current theories.[1] However, modern 'principle-based' linguistic theories are complex, abstract, and formally underspecified, making them difficult to incorporate within current NLP systems. To begin exploiting these linguistic developments in computational systems, we require a rigorous framework in which we can formally represent such grammars and then construct systems which use these grammar respresentations transparently. We argue that the logic programming paradigm provides such a framework.

[1]This is the case regardless of whether or not a syntactic representation is explicitly constructed. That is, such constraints may be used to directly map from an input string to, say, some meaning representation language.

However, while phrase structure grammars are highly amenable to axiomatisation in horn-clause logic, and lexicalist frameworks such as categorial grammar have well-defined logical interpretations, this paper considers how more heterogeneous, principle-based linguistic theories might benefit similarly from the logic programming paradigm. We demonstrate that while deductive parsing techniques can be fruitfully applied to principle-based or 'licensing' grammars, several interesting differences emerge. First, we show that the correspondence between a syntactic analysis and a proof must be reconsidered, and the deductive techniques revised. Secondly, we argue that such grammatical theories may be most effectively axiomatised as several sub-grammars in which particular constraints are defined over the simple representational types to which they apply. We then consider the motivations and implications of such axiomatisations for efficient implementation of principle-based parsing systems.

The relationship between logic programming and language processing dates back to the origins of Prolog. Indeed, Definite Clause Grammars (DCGs), now a component of virtually all Prolog implementations, provide a notation explicitly for writing phrase structure grammars. Once these grammar rules are compiled into standard Prolog[2] they receive a procedural interpretation, becoming a top-down, left-to-right, recursive-descent parser. That is to say, by representing the rules of grammar as axioms in Prolog's horn-clause logic, we can use Prolog's theorem proving engine as a parser.

This natural embedding of phrase structure grammars in Prolog suggests that we should be able to cast natural language parsing into the broader logic programming paradigm. That is to say, distinction between *logic* and *control* within logic programming may be naturally inherited by the parsing domain. This constitutes the so-called *Parsing as Deduction* (PAD) hypothesis, wherein a parser is defined as a logical specification of a grammar, in conjunction with some deductive inference engine which realises a particular parsing algorithm for that grammar (Pereira and Warren, 1983) (Pereira and Shieber, 1987). So just as there is a direct correspondence between a grammar and its logical representation as horn clauses, so is there a similar correspondence between the logical inferencing (or theorem proving) strategy and the parsing algorithm. A further, and important, correspondence is that for such logic grammars, the 'proof' that a sentence is indeed a theorem of the grammatical axioms is precisely the parse tree for that sentence.

If we adopt the PAD methodology, the task of developing a parser can be simply broken down into (a) the provision of a logical specification of our grammar, and (b) the construction of a suitable theorem prover. The latter of course, is dependent upon the richness of the logic we have used, and may conceivably take advantage of specific properties of our grammar axiomatisation.[3] While such deductive parsing techniques are well-understood for traditional rule-based, phrase structure grammars, they are rather more elusive for the current wave of principle-based or constraint-based grammars. In this paper, we argue that a major source of difficulty arises from a fundamental difference in the way such grammars should be axiomatised. While rule-based grammars typically consist of a set of *sufficient* 'structure-generating' axioms, principle-based grammars are more naturally expressed as a set of *necessary* 'structure-licensing' conditions which in essence rule-out ill-formed structures, rather than generating well-formed ones. On the basis of this observation we propose a new methodology for implementing deductive parsers for this class of licensing grammars.

We begin below with a brief discussion of principle-based parsing, highlighting the tendency to adapt traditional rule-oriented parsing technology. In particular, we stress that the standard notions of parsing are inherently biased towards the homogeneous nature of rule-based grammars, and are inappropriate given current abstract licensing grammars which consist of a small set of interacting principles/constraints – typified by the principles and parameters paradigm, e.g. *Government-Binding Theory* (Chomsky, 1981) (Lasnik and Uriagereka, 1988). We then cast the parsing problem in deductive terms, to provide a more formal foundation for our discussion. In particular, we observe that construction-oriented, rule-based grammars (such as CFGs, etc.) typically consist of structure specific rules, where one rule is sufficient to license or indeed generate a particular instance of syntactic structure. In the context of licensing grammars, however, particular instances of syntactic structure are often required to meet a number of more abstract conditions which interact.

We will then examine some previous approaches to implementing deductive, principle-based parsers, and the key problems they face. In particular, we consider the relationship between a syntactic analysis and the deductive proof. While the relationship is direct for phrase structure grammars, this is not the case for the current principle-based accounts in which principles in-

[2]The translation from DCG form into Prolog is a trivial one, which requires the simple addition of string-handling difference-lists to the original rules. For a thorough exposition of DCG implementation see (Pereira and Warren, 1980), and for other logic programming formalisms see (Abramson and Dahl, 1989).

[3]Consider, for example, that the inference procedure used by Prolog to parse DCGs restricts one not only to grammars that may be specified as horn-clauses, but also to grammars that contain no left-recursive rules.

teract combinatorially to license annotated structural representations. In contrast with some existing deductive, principle-based techniques, we advocate an approach which *decomposes* syntactic analysis into several uniform representation types. This reduces the logical and deductive complexity of the component systems, permitting the use of simple and well-understood techniques. Within this approach, a complete syntactic analysis consists of a 'tuple' of proofs; each corresponding to a particular representational aspect of the overall analysis. We demonstrate the application of this technique by constructing a grammar fragment which distinguishes the recovery of local and hierarchical constituent structure (or *phrase structure*), from the determination of long distance relationships, called *chains*.

1.1 Parsing with Principles

The computational linguistics community has recently exhibited increasing interest in the development of systems based on the principles and parameters model of current linguistic theory.[4] The adoption of this paradigm remains rather tentative, however, for several reasons. First, principle-based syntactic theories such as GB theory are typically complex, unstable, and – by computational standards – informal. Thus there is no readily available formal specification of what a GB grammar is (but cf. (Stabler,1992)). Secondly, there has been a reluctance to abandon the traditional parsing technology – centered around phrase structure grammars (or, equivalents) – which is rendered largely inadequate by the modular, heterogeneous and abstract nature of principle-based theories. The recent trend towards statistical parsing techniques, has also shifted emphasis away from the explicit formalisation of linguistic knowledge, to its 'approximation' (and acquisition) by data-intensive, stochastic techniques (though Fordham and Crocker (1997) consider how the principle-based and stochastic techniques might be combined).

There are, however, a number of arguments in favour of pursuing principle-based systems. The most obvious is that it allows the exploitation of 'state of the art' syntactic theorising. In this way we might also contribute to the formalisation of syntactic theory as it develops (see (Stabler, 1992) in particular). From an 'engineering' perspective, there are other potential advantages; principle-based systems are much more compact and may well be easier to maintain than construction-based systems. While the interaction of principles is typically more complex, the principles themselves are relatively simple and prohibit the 'yet another special rule' approach which is rife in the development of construction-based systems. Furthermore, given the typically large numbers of rules involved in the latter, there is no explicit notion of an underlying theory which can be used to justify particular rules, while principles must remain consistent with the overall theory of grammar. A further appeal is that principle-based systems may be designed to apply cross-linguistically, sharing the fundamental grammar and parsing machinery, while construction based systems are inherently language specific.

1.2 Parsing as Deduction

In an effort to construct more faithful and transparent realisations of principle-based systems, there has been recent interest in so-called 'deductive' parsing methods. As we sketched above, this approach explicitly separates the axiomatisation of grammatical principles – a purely declarative specification – from the procedures which use these axioms to 'prove' derivations of syntactic analyses. In particular it has been shown that meta-interpreters or program transformations can be used to affect the manner in which a logic grammar is parsed (Pereira and Warren, 1983), while leaving the grammar unaltered (or logically equivalent). That is, for a given logic grammar we can construct a variety of parsers such as LL, LR, and Earley, among others. One problem, however, is that not all parsing strategies are suited to all grammars. A recursive descent parser, for example, may not terminate for a grammar with left recursive rules, while bottom-up parsers are typically unsuitable for grammars with empty productions (Pereira and Shieber (1987) provide a thorough exposition of these issues).

One attempt to extend the PAD hypothesis be-

[4] See Berwick et al (1991) for a good introduction, and a collection of papers on various systems and approaches. Additional systems are presented by Crocker (1991b) and Merlo (1995).

yond its application to simple phrase structure logic grammars is presented by Johnson (1989). In particular, Johnson has developed a prototype parser for a fragment of a GB grammar. The system consists of a declarative specification of the GB model, which incorporates the various principles of grammar and multiple levels of representation. His top-level axiomatisation is as follows:

```
(1) parse(String,LF) :-
                xBar(infl2,DS),
                theta(infl2,0,DS),
                moveAlpha(DS,[],SS,[]),
                caseFilter(info2,0,SS),
                phonology(String/[],SS),
                lfMovements(SS,LF).
```

This specification transparently encodes the standard T-model of transformational syntax: xBar and theta instantiate well-formed D-structure (DS) representations, moveAlpha then transforms these into candidate S-structures (SS), which are in turn mapped onto phonetic (in this case a String) form and logical form (LF). When trying to compute this relation using Prolog's default control mechanism, however, this system has obvious problems. The result will be a naïve generate and test parser, since D-structure and S-structure will first be generated by the grammar and only then matched to the String by the phonology predicate. Furthermore, if xBar were to contain left recursive axioms (Johnson's does not, but this would be required for a more complete axiomatisation) that predicate alone might never halt.

It is at this point that the declarative interpretation of logic programs becomes of practical use. Crucially, Johnson illustrates how the fold/unfold transformation, when (manually) applied to various components of the grammar, can be used to render more efficient implementations derived from the sort of axiomatisation given above. This approach effectively reaxiomatises the grammar into a system more amenable to Prolog's control and inference regimes.[5] Roughly speaking, this can be viewed as a step in the direction of *partially evaluating* a principle-based system into a set of phrase structure rules (although Johnson does not advocate such a move); moving from an abstract specification to a more concrete or 'compiled-out' form (for discussion of this in a non-deductive context see (Merlo, 1995)).

While the transformation approach may be a practical solution for constructing efficient parsers, it loses the appeal of a system which directly exploits a compact, modular system of principles *on-line*. Also, every time a change is made to the underlying grammar, it will need to be re-transformed, and the nature of these transformations may need to be revised for the new grammar. As an alternative, Johnson also demonstrates how goal *freezing*, an alternative Prolog control strategy, can be used to increase efficiency by effectively coroutining the recovery of the various levels of representation, allowing all principles to be applied as soon as possible, at all levels of representation. While attractive, this approach is not without its difficulties; the success of coroutining relies not only on the careful encoding of the representations, but efficiency and indeed termination properties will depend on precisely how and when the various principles apply to successfully constrain what is essentially an *informed* generate and test procedure.

In sum, the deductive approach to parsing is theoretically attractive, but unsurprisingly inherits a number of problems with automated deduction in general. Real automated theorem provers are, at least in the general case, incomplete. That is, they cannot *a priori* be guaranteed to return all (or any) solutions to a given request. One instance of this is the left-recursion example cited above; a perfectly legitimate grammar rule may cause the Prolog inference engine to pursue an infinite path and never halt. While it is possible to solve or at least detect some of these problems, especially for parsing algorithms and grammars formalisms which are well understood,[6] it is certainly not possible in the general case. We can therefore imagine that a true, deductive implementation of GB would present a problem. Unlike traditional, homogeneous phrase structure grammars, GB makes use of abstract, modular principles, each of which may be relevant to only a particular type or level of representation. This modular, heterogeneous organisation makes the task of

[5] See (Johnson, 1991) for further discussion of such techniques.

[6] As another example, many bottom-up algorithms cannot be guaranteed to succeed if there are empty productions in the grammar.

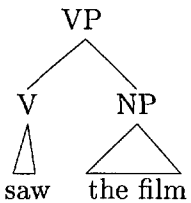
deriving some single, specialised interpreter with adequate coverage and efficiency a very difficult one. For further illumination of these issues, the reader is referred to (Stabler, 1992), although that work is not specifically directed at parsing concerns.

2 Rules *versus* Principles

The traditional characterisation of language in terms of construction-based systems (i.e. systems which use individual rules to describe units of structure, as exemplified by a context-free grammar) has significantly influenced our views about parsing. In particular, there is a tendency for parsers to use the rules of grammar to 'generate' instances of structure. This is possible because of the homogeneous nature of construction-based grammars, where a single rule is sufficient to license a particular unit of syntactic structure.

This property of 'rule-to-structure' correspondence does not obtain in principle-based grammars, however, where particular units of structure must satisfy a collection of relevant principles. Furthermore, any given principles might only be concerned with a particular aspect of that structure. Consider, for example, the following VP structure:

(2)



In a typical phrase structure grammar, this structure (excluding the subtrees of V and NP, and henceforth written as $[_{V^2}\ V^0\ N^2]$) is licensed by the single rule; VP $\rightarrow$ V NP. In a principle-based grammar, the structure is only well-formed if it satisfies a number of principles; $\overline{X}$-theory licenses the basic structural configuration (i.e. the complement NP as sister to the V^{min} projection, and dominated by a higher V projection, namely VP), the θ-criterion is satisfied since the NP both requires a thematic (θ-)role and occupies a θ-marked position, and finally, the NP must satisfy the Case filter (which it does, due to the transitive verb). Given that the principles are each concerned with only a particular aspect of the syntactic structure, none are particularly appropriate for generating the structure. In sum, principles are more naturally viewed as constraints on syntactic structures, rather than generators of them.

The approach adopted in most existing principle-based parsers is to treat the rules of $\overline{X}$-theory as structure generators, and then apply the principles as constraints on these structures (e.g. see (Crocker, 1991b) and also the discussion of other systems by Berwick (1991)). This technique has a number of potential disadvantages, however. In the first place, $\overline{X}$-theory is a principle of D-structure in most current instantiations of the theory, and hence is not sufficient for generating the set of possible S-structure configurations.[7] Secondly, there has been an increase in support for the abolition of $\overline{X}$ 'rules' *per se*, favouring feature-based constraints derived from more fundamental properties of lexical items, e.g. (Speas, 1990).

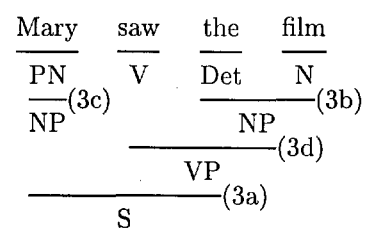
The fundamental difference between rule- and principle-based systems – i.e. rules as 'generators' versus principles as 'constraints' – is made more precise when cast in terms of deductive parsing. To begin, consider the following (horn clause) axiomatisation of a simple context-free grammar:

(3)

(a)	S	$\leftarrow$	NP $\wedge$ VP
(b)	NP	$\leftarrow$	Det $\wedge$ N
(c)	NP	$\leftarrow$	PN
(d)	VP	$\leftarrow$	V $\wedge$ NP
(e)	PN	$\leftarrow$	"Mary"
(f)	Det	$\leftarrow$	"the"
(g)	N	$\leftarrow$	"film"
(h)	V	$\leftarrow$	"saw"

Now we can illustrate the derivation or 'proof' of a sentence S for the string *Mary saw the film*, as follows:

(4)



Borrowing a notation widely used in the Categorial Grammar literature (as introduced by Ades

[7] That is, surface structures may be composed of both the configurations licensed by $\overline{X}$-theory and those which result from transformations, such as adjunction.

and Steedman (1982)), this derivation illustrates a complete proof of the theorem: S ← *Mary saw the film*, derived from the axioms given in (3).[8] Furthermore, the derivation transparently represents the (inverted) phrase structure for the sentence, where derivation steps are translated into branches connecting the consequence (below the derivation line) to it premises (above the derivation line) – this follows from the fact that the rules/axioms directly characterise the well-formed units of structure, as discussed above.

Note, it is possible to construct our derivation such that we record the proof/constituent structure as we go along. This is accomplished by writing the derived consequence X as [$_X$ Prem1 ...PremN], where each premise in turn is the structure of the sub-proof:

(5)

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Mary}}{\text{PN}}\quad\dfrac{\text{saw}}{\text{V}}}{[_{NP}\ \text{PN}]}(3c)\quad\dfrac{\dfrac{\text{the}}{\text{Det}}\quad\dfrac{\text{film}}{\text{N}}}{[_{NP}\ \text{Det N}]}(3b)}{[_{VP}\ \text{V}\ [_{NP}\ \text{Det N}]]}(3d)}{[_S\ [_{NP}\ \text{PN}]\ [_{VP}\ \text{V}\ [_{NP}\ \text{Det N}]]]}(3a)$$

Thus the final derivation is in fact a bracketed list encoding the structure of the original derivation in (4).[9] Crucially, while it is possible to carry the sub-proof as we construct the derivation (as in (5)), this structure is not used or referenced by the axioms of grammar (indeed, (4) does not build such a proof record). In early transformational grammar, however, it was possible write such structure sensitive rules, as in the case of the passive[10]:

(6) [$_S$ NP^1 [$_{VP}$ V NP^2]] ← [$_S$ NP^2 [$_{VP}$ V [$_{PP}$ by NP^1]]]

As we discussed above, current conceptions of the phrase-structure component within the principles and parameters approach are heading away from such a rule-oriented characterisation. This is exemplified by the *Project Alpha* proposal of Speas, where the projection of structure from the lexicon is free, and only subsequently constrained by the syntax (Speas, 1990). This move is essentially the final step in eliminating the rule component in favour of a pure 'licensing' grammar. In sum, the view is one where syntax simply constrains (or, licenses) virtually arbitrary structures, rather than generating them.

Our approach is to characterize the well-formed structures of a language in two stages. First, we define a set of possible structures against which the grammatical principles can apply. Secondly, we specify the grammatical principles which pick out those members of the set of possible structures which are, in fact, well-formed. What are the possible structures? Here, we begin with the view that they are simply binary branching trees. We will further assume that the principles that apply to them are strictly *local* and are defined exclusively with respect to local structure units. That is, our principles will not apply to whole trees but only to branches of trees, e.g. [$_{V^2}$ V^0 N^2]. This restriction permits the close interleaving of the two stages outlined above; as each new binary branch is proposed during the construction of a tree, we can immediately verify the local well-formedness of that branch (Crocker (1991a, 1996) provide more detailed discussion and illustration of this point).

It might be objected that a complete axiomatisation incorporating movement would entail the use of *tree transducing* axioms, permitting the description and constraint of Move-α.[11] This would eliminate the strictly local characterisation of the principles.[12] In recent work, however, Crocker (1992,1996) develops a 'representational' reformulation of the transformational model which de-

[8] In fact, the reader may have noticed that linear order is implicit in this system. That is, NP ← Det ∧ N, implies that Det is adjacent to, and precedes, N. Furthermore, any theorem (i.e. derived result) or premise (i.e. lexical item) may only be used once in the course of the derivation. This may be defined naturally within the general framework of a linear logic. For our purely illustrative purposes, however, a thorough and formal exposition of the assumed logical framework would take us too far afield.

[9] See (Stabler, 1987) for a discussion relating the proof procedure assumed in (4) with an equivalent one which records the proof tree, as in (5).

[10] Note, this transformational rule is a simplified reformulation of the standard passive rule in TG, where the premise is the 'structural description', and the consequence is the resulting 'structural change'.

[11] See (Stabler, 1992) for an elaborate exposition of this approach.

[12] Note, the approach presented here would still be applicable, but the axiomatisation would be rather more complicated, and the resulting computational complexity would be increased.

composes syntactic analysis into several representation types – including phrase structure, chains, and coindexation – allowing one to maintain the strictly local characterisation of principles with respect to their relevant representation type. This approach entails the use of multiple inference engines: one dedicated to the construction of each representation.

For expository purposes, we will first consider a single, phrase structure representation. This allows us to illustrate the general licensing approach for a simple grammar. We then consider the treatment of non-local dependencies, which is realised by introducing the representation of *chain structure*. We first extend the phrase structure system to allow empty categories, and a broader range of structures. We then show how the techniques developed for proving phrase structure 'theorems' for a given string can be similarly exploited to prove chain structure theorems for a given phrase structure. A complete, well-formed analysis is obtained precisely when we can prove a phrase structure which covers the string, and a chain structure which covers the phrase structure. Finally, we consider the issue of control in the context of our multiple deductive systems.

2.1 Licensing and Phrase Structure

Given our commitment to *local binary branches*, our procedure is the following: First, we define the set of possible binary branches. Secondly, we apply the principles of grammar to the branches thereby defining the set of proper branches, and thirdly, we define the set of well-formed binary branching trees from of the set of proper branches. This can be formalised straightforwardly as follows:

(7) I. Two sets of nodes:
(a) The set T of all terminals.
(b) The set NT of all non-terminals.

II. The set B of branches:
(a) if $X, Y, Z \epsilon NT$, then $[_X\ Y\ Z\ ] \epsilon B$
(b) if $X \epsilon NT$, $Y \epsilon T$, then $[_X\ Y\ ] \epsilon B$
(c) there is nothing else ϵB

III. The set PB of proper branches:
(a) $\alpha \epsilon PB$ iff $\alpha \epsilon B$, and
(b) α meets all necessary conditions in (8)

IV. The set Tr of well-formed trees:
(a) if $\alpha = [_X\ Y\ ] \epsilon PB$, then $\alpha \epsilon Tr$
(b) if $A_t, B_t \epsilon Tr$ and $[_X\ A\ B\ ] \epsilon PB$, then $X_t = [_X\ A_t\ B_t\ ] \epsilon Tr$
(Z_t denotes a tree rooted at the NT node Z)

To complete the definition, we now give an example of a rather simple set of principles, which state the conditions which are necessary for a branch ($b \in B$) to be a proper branch ($b \in PB$).[13] In the following definitions, X, Y, Z are variables over NT, Word is a variable over T and N, D, $V \in NT$.

(8) $\overline{X}$-theory:
(a) $[_{X^i}\ Y^j\ Z^k\ ] \rightarrow$
X = Z, i=j=2, k $\leq$ 1, Y is-spec-of X
or,
X = Y, j=0, k=2, 1 $\leq$ i $\leq$ 2.
(b) $[_{X^n}$ Word$] \rightarrow$ cat(Word,X), 2 $\geq$ n $\geq$ 0.

Case Filter:
(c) $[_Z\ X\ N^2\ ] \rightarrow$ case-assigner(X).

θ-Criterion:
(d) $[_Z\ X^0\ Y^2\ ] \rightarrow$ θ-marks(X^0,Y^2).

Lexicon/Parameters:
(e) cat('the',D).
(f) cat('film',N).
(g) cat('saw',V).
(h) case-assigner(V).
(i) θ-marks(V^0,N^2).
(j) D is-spec-of N.

In these rules, the left-hand-side shows a structure which pattern-matches against structures in B. The right-hand-side states a necessary condition for the structure to be a proper branch. In contrast, our earlier CFG rules stated *sufficient* conditions for structures to be well-formed.

Given this set of axioms, consider the following proof of VP $\rightarrow$ *saw the film*:[14]

[13]The given definitions are only intended as simple approximations for the purposes of exposition, and are not to be construed as axiomatisations of the actual grammatical principles.

[14]We consider only a VP constituent for the moment, since a full sentence would require an axiomatisation of movement, which is not included in the current fragment. Precisely such a grammar is developed in the next section.

(9)

$$\dfrac{\dfrac{\text{saw}}{[_{V^0}\ \text{saw}]^{\{8b,g\}}} \quad \dfrac{\dfrac{\text{the}}{[_{D^2}\ \text{the}]^{\{8b,e\}}} \quad \dfrac{\text{film}}{[_{N^1}\ \text{film}]^{\{8b,f\}}}}{[_{N^2}\ \text{D}^2\ \text{N}^1]^{\{8a,j\}}}\text{(7IIa)}}{[_{V^2}\ \text{V}^0\ \text{N}^2]^{\{8a,c,d,h,i\}}}\text{(7IIa)}$$

In such a proof, simply drawing a line under one or more structures and writing a new structure underneath the line corresponds to constructing a branch using clause (7IIa) where X and Y are instantiated on the basis of the root node of each structure above the line (or by clause (7IIb), for the lexical cases). The superscript rule numbers next to each branch indicate which constraints are satisfied to show the branch is a proper branch (in accordance with (7III)). By virtue of (7IV) we can then construct the well-formed tree:

(10) $[_{V^2}\ [_{V^0}\ \text{saw}]\ [_{N^2}\ [_{D^2}\ \text{the}]\ [_{N^1}\ \text{film}]]]$

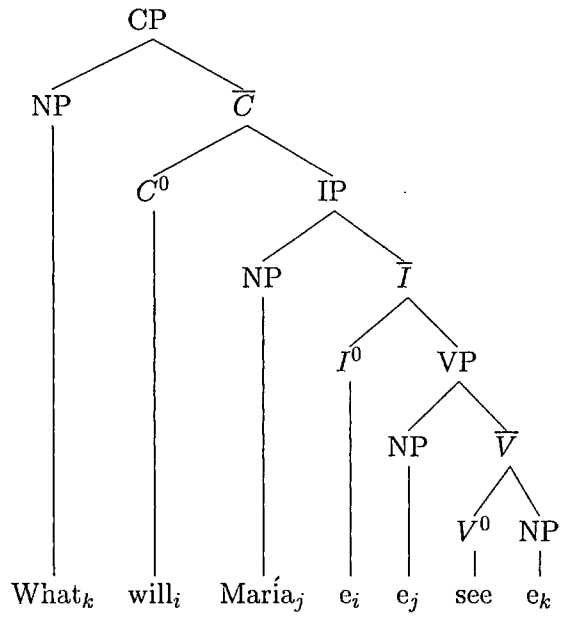
From this example, we can see that the axiomatisation and derivations for a principle-based – or, more properly, licensing – grammar are rather more complex than for a traditional 'construction-based' CFG. Most importantly, the derivation of a particular unit of structure is not supported by an individual (sufficient) axiom characterising precisely that construction, but must rather be consistent with the complete set of (necessary) axioms – each of which might only constrain some aspect of that unit of structure.

To summarise, the difference between rule-based and licensing grammars can be conceptually characterised as follows: Assuming that the task of a grammar is to define the set of well-formed syntactic analyses of a language, the rule-based grammar provides a set of sufficient conditions for the construction of particular units of structure (e.g. to construct an NP it is sufficient to have a determiner and a noun, NP ← Det N). In a licensing grammar however – and assuming that principle-based grammars are licensing grammars – the axioms do not 'generate' well-formed structures. Rather the axioms are necessary conditions which, given some structure, determine whether or not it is a candidate member of the set of well-formed formulae (wffs) and only once all the relevant axioms are satisfied can we be sure it is indeed a wff.

2.2 Licensing and Chains

It should now be clear that the licensing grammar introduced above is extremely limited in its potential coverage by the fact that only local (proper branch) dependencies can be expressed.[15] Analyses within current principle-based accounts often posit numerous such dependencies, even for superficially simple utterances. Consider the sentence *What will María see?*, which might reasonably be assigned an analysis as follows:

(11)



In comparison with the simple grammar fragment introduced above, such an analysis requires several additions. We need to introduce relevant constraints for the clausal, functional categories C and I including their valid head, complement and specifier selections. We need also permit NPs (and also I) to be empty, dominating *traces* which replace the lexical element which has moved out of that position. None of these additions is particularly onerous, and are achieved by extending the phrase structure constraints given in (8) as follows:

(12) $\overline{X}$-theory:
(a)$[_{X^i}\ \text{Y}^j\ \text{Z}^k\ ] \rightarrow$
X = Z, i=j=2, k $\leq$ 1, Y is-spec-of X

[15] It is of course possible to implement longer distance relations using feature passing techniques, such as GPSG's slash category. This will, however, compromise the efficiency of the system by introducing a greater degree of backtracking, i.e. proper branches will be licensed contingent upon features successfully being passed between mother and daughters.

or,
X = Y, j=0, k=2, $1 \leq i \leq 2$.

(b) $[X^n$ Word$] \rightarrow$ cat(Word,X), $2 \geq n \geq 0$.

Case Filter:

(c) $[_Z$ V N^2 $] \rightarrow$ case-assigner(V)
or,
$[_Z$ N^2 I $] \rightarrow$ case-assigner(I).

θ-Criterion:

(d) $[_Z$ X^0 Y^2 $] \rightarrow$ θ-marks(X,Y^2)
or,
$[_Z$ Y^2 V^1 $] \rightarrow$ θ-marks(V,Y^2).

Lexicon/Parameters:

(e) cat('What',N).
(f) cat('Maria',N).
(g) cat('see',V).
(h) cat('will',I).
(i) cat('will',C).
(j) cat(e,N).
(k) cat(e,I).
(l) case-assigner(V^0).
(m) case-assigner(I^1).
(n) θ-marks(V,N).
(o) θ-marks(I,V).
(p) θ-marks(C,I).
(q) N is-spec-of V.
(r) N is-spec-of I.
(s) N is-spec-of C.

The main additions to the system are: the introduction of N and I traces (12j,k),[16] the addition of the functional categories I and C and their relevant properties (12h,i,m,o,p,r,s),[17] and the introduction of NP specifiers of V (12q).[18] To simplify the system slightly, we have also removed determiners in favour of a simple proper noun (12f) and Wh-pronoun (12e). This revised grammar axiomatisation is now sufficient to prove the appropriate structure for CP $\rightarrow$ *What will María see*, shown in (14).

As before, we annotate each derivation with reference to the axioms required to license the corresponding branch of structure. Again by virtue of (7IV) we can construct the resulting well-formed tree,shown by the following bracketed list:

(13) $[_{C^2}$ $[_{N^2}$ What $]$ $[_{C^1}$ $[_{C^0}$ will $]$ $[_{I^2}$ $[_{N^2}$ María $]$ $[_{I^1}$ $[_{I^0}$ $e_1]$ $[_{V^2}$ $[_{N^2}$ $e_2]$ $[_{V^1}$ $[_{V^0}$ see $]$ $[_{N^2}$ $e_3]]]$ $]]]]$

The are, however, two important points about the revised system. Firstly, while the above is the correct parse tree, it is an incomplete syntactic analysis. While the system successfully recovers the phrase structure in (11), the coindexations (e.g. between *what* and the trace in the verbs object position) are not represented (the subscripts on traces are purely for later identification, and not recovered by the system). Furthermore, the phrase structure axioms now substantially overgenerate. In particular, we could generate the above tree with or without traces in any of the positions, parse it as a CP or IP, and all such permutations. Clearly these two points are related, in that only that trees for which there are well-formed coindexations among moved constituents and traces, should receive a parse.

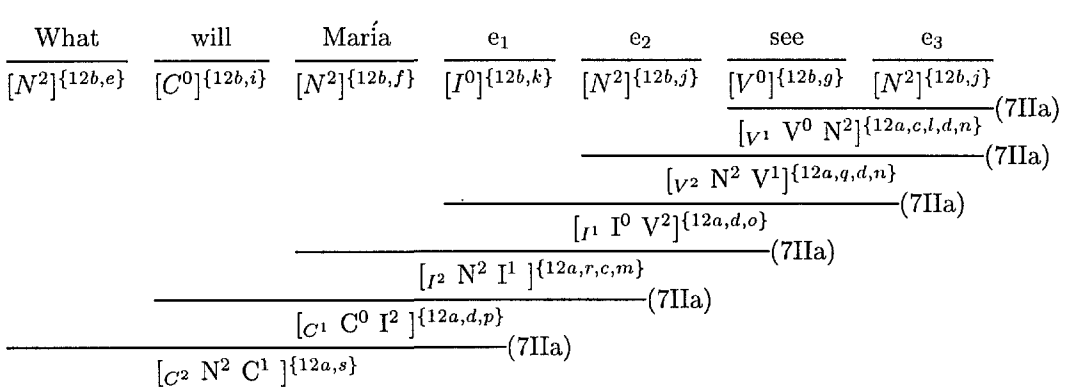
In this section we will argue that the correct solution to this problem is not to augment the phrase structure component described above. Such a move would complicate the simple representational schema for phrase structure, and weaken the notion of locality. That is, it would no longer be possible to define the necessary axioms in terms of a proper branch. Rather, we introduce a new type of licensing grammar, exclusively for the purpose of recovering a representation of long distance relationships among constituents. We will call this representation a *chain*, consisting of a list of constituents which enter into a well-formed long distance relation, and thus capture the coindexations shown in (11). Indeed, just as we have argued for the view of phrase structure as an abbreviated proof that a given string is a valid theorem w.r.t. the axioms over proper branches, we will now propose that the recovery of a chain structure constitutes a similarly abbreviated proof that a given phrase structure is a valid theorem, w.r.t the axioms over the *proper links* of chains. The result is a two-stage deductive parsing system which distinguishes the recovery of phrase structure from the recover of chains. The advantage of such an articulated system is that each component — phrase structure and chains — is kept simple, with axioms defined as strictly local, necessary constraints licensing branches and links, respectively.

[16] We assume the parsing engine treats e as the empty string appropriately. The prototype parser is a top-down one, so empty categories do not cause a problem. We return to this issue at the end of this section.

[17] We specify the head of I, *will*, also as head of C, since it may move to this position. This a simply a substitute for a more sophisticated treatment of head-movement.

[18] Our examples in this section will assume the VP-internal subject hypothesis that subjects are base-generated and hence θ-marked in the [Spec,VP] position (i.e. the specifier of VP), and move to the [Spec,IP] position to receive Case. Further, we assume that any Case/θ-marked NP can move to [Spec,CP] to form a Wh-question.

(14)



We begin as we did for our phrase structure system, by providing an axiomatisation for well-formed chains. In contrast with the binary branching tree representation of phrase structure, chains can be encoded as lists which contain one lexical antecedent as their head, followed by an arbitrary number of traces, and ending with *nil*. Just as the well-formedness of trees was recursively defined in terms of well-formed (proper) branches, so can well-formed chains be defined in terms of proper links. The details of this formulation are as follows:

(15) I. Two sets of nodes:
(a) The set L of all lexical nodes.
(b) The set NL of all non-lexical (empty) nodes.

II. The set L_{ink} of links:
(a) if $X \epsilon L$, $Y \epsilon NL \cup nil$ then $[X\ Y] \epsilon L_{ink}$
(b) if $X \epsilon NL$, $Y \epsilon NL \cup nil$ then $[X\ Y] \epsilon L_{ink}$
(c) there is nothing else ϵL_{ink}

III. The set PL of proper links:
(a) $\alpha \epsilon PL$ iff $\alpha \epsilon L_{ink}$, and
(b) α meets all necessary conditions in (16)

IV. The set Ch of well-formed chains:
(a) if $\alpha = [X\ Y] \epsilon PL$, then $\alpha \epsilon Ch$
(b) if $Y_c \epsilon Ch$ and $[X\ Y] \epsilon PL$, then $X_c = [X\ Y_c] \epsilon Ch$
(Z_c denotes a chain headed by the node Z)

The formulation above defines a chain to be a list, headed by a lexical node,[19] followed by a list of arbitrary many NL nodes (including zero), and ending in *nil*. This allows for the singleton chain [L nil] to represent constituents which have not moved. Our specification of grammatical well-formedness constraints determining the proper links is given below.[20] Where nodes are prefixed (e.g. by 'lex' (lexical), 'a-pos' (A-postion), and 'a-bar' ($\overline{A}$-position)), we assume these attributes are accessible by whatever mechanism matches postulated links with the constraints. We discuss later how such attributes might be assigned in the first place.

(16) θ-Criterion:
(a) [N^2 nil] $\rightarrow$ theta-marked(N^2).

Case Theory:
(b) [a-bar:N^2_a a-pos:N^2_b] $\rightarrow$ case-marked(N^2_b)
or,
[lex:N^2 X] $\rightarrow$ case-marked(N^2).

A-to-$\overline{A}$ Constraint:
(c) [N^2_a N^2_b] $\rightarrow$ not(a-pos:N^2_a and a-bar:N^2_b).

Head Movement:
(d) [C^0 I^0] $\rightarrow$ true.

Category Constraint:
(e) [X Y] $\rightarrow$ X = Y = N *or* X, Y ϵ {C,I}.

Level Constraint:
(f) [X^i Y^j] $\rightarrow$ i=j=0 *or* i=j=2.

To briefly summarise these constraints, (16a) states that the last NP in a chain (the one linked to *nil*) must be θ-marked.[21] Case Theory in (16b)

[19] Since (15II) does not permit the links of the form: [NL L].

[20] Again, we stress that this formulation is simplified for expository purposes, and is not intended to represent an actual axiomatisation of current syntactic theory.

[21] We will not give definitions for the theta-marked/1 and case-marked/1 predicates here. We assume that they simply check that the relevant features are instantiated on the

states that the NP in the highest A-position (either [Spec,IP], [Spec,VP] or [Comp,VP]) must be Case marked – the two possible link configurations are either the NP linked to and $\overline{A}$-node or the head of the chain. The A-to-$\overline{A}$ Constraint in (16c) encodes the fact that there is no movement from an $\overline{A}$-position ([Spec,CP]) to an A-position. No constraints on Head movement are required for the current fragment, so this is represented by the trivially satisfied constraint in (16d). And finally, the Category constraint requires that any two linked nodes be either the same category or functional categories (16e), while the Level Constraint requires both nodes to have the same level, either 0 or 2 (16f). As with $\overline{X}$-theory for phrase structure, these last two constraints apply to all links (except those with *nil*).

Chain structure is concerned only with a particular subset of constituents in the phrase structure, namely lexical and empty NPs and the heads of the function categories (C and I). We will therefore assume here, the existence of an *interface* procedure which traverses a particular phrase structure to find these nodes which will constitute the premises of our chain structure derivation. This procedure might be reasonably used to annotate the nodes with the relevant 'lex', 'a-pos', and 'a-bar' information mentioned above (or this could simply be done by the parser). The result of this procedure will recovery of the following set of nodes (shown below with the (non-)lexical material they dominate):

(17)

What	will	María	e_1	e_2	e_3
N^2	C^0	N^2	I^0	N^2	N^2

The task of our chain deduction system is to show that this set constitutes a theorem, or rather *theorems*, since more that one chain will be formed. The derivations of each chain (i.e. one for each lexical node), are shown below, along with the its complete chain representation as constructed by (15IV):

nodes, and the counterpart Case Filter and θ-criterion in the phrase structure component do the relevant instantiations (as well as licensing).

(18)

$$\dfrac{\text{What}\quad \dfrac{e_3 \quad \dfrac{}{nil}}{[N^2\ nil]^{\{16a,c\}}}(15\text{IIb})}{[N^2\ N^2]^{\{16b,c,e,f\}}}(15\text{IIa})$$

$\Longrightarrow$ [What, e_3, nil]

(19)

$$\dfrac{\text{will}\quad \dfrac{e_1 \quad \dfrac{}{nil}}{[I^0\ nil]}(15\text{IIb})}{[C^0\ I^0]^{\{16d,e,f\}}}(15\text{IIa})$$

$\Longrightarrow$ [will, e_i, nil]

(20)

$$\dfrac{\text{María}\quad \dfrac{e_2 \quad \dfrac{}{nil}}{[N^2\ nil]^{\{16a,c\}}}(15\text{IIb})}{[N^2\ N^2]^{\{16b,c,e,f\}}}(15\text{IIa})$$

$\Longrightarrow$ [María, e_2, nil]

By successfully deriving a proof of chain structure (i.e. the above set of well-formed chains) for the phrase structure derivation in (14) we have (i) proven that the phrase structure is a theorem of the chain grammar, and (ii) recovered the long distance relationships (i.e. in the representation of chains) which corresponds to the coindexations shown in (11). As a result, we have a complete proof that the string *What will María see* is well-formed, and our 2-tuple of proofs, phrase structure and chain structure, provide the abbreviated proofs which are the complete syntactic analysis. As we saw for the deduction of phrase structure, the links of chains do not correspond to some single grammatical axiom, but rather meet a variety of independent necessary constraints. It should also be apparent that alternative possible phrase structure derivations, such as those with omitted/extra traces would not be provable as theorems of the chain grammar, and thus rejected.

We have presented a highly simplified grammatical axiomatisation of both phrase structure and chain systems, with the aim of demonstrating how a set of necessary licensing constraints

can be exploited in a syntactic parsing framework, and how distinct representations may be decomposed, logically, while contributing to a single, complete linguistic analysis. The proposed techniques have, however been exploited in a system of more substantial linguistic coverage. In particular, the techniques outlined here underlie the basis of a larger project on the construction of a bilingual (English-German) system (Crocker, 1992, 1996). The coverage of that systems includes the following construction types (for both languages):

- Subcategorization: complex subcategorization patterns (intransitive, transitive, ditransitive, and sentence complements).
- Case-marking: morphological and abstract case assignment and exceptional case marking phenomena.
- Thematic-role assignment: as determined by subcategorization and case.
- Head movement: subject-aux inversion, V-2 raising in German, and V-to-I raising for inflection.
- Recursive sentences: relative clauses and sentential complements.
- Cyclic movement: long-distance movement from within embedded clauses.

The full system further decomposes the thematic representation from the phrase structure representation, primarily for reasons of cognitive plausibility which we are not concerned with here. Indeed it should be noted that the primary function of the system was as a model of human syntactic parsing, and thus the coverage of the grammar was determined by those constructions which were of linguistic and psychological interest, rather than a goal of wide, practical coverage. The current implementation of the system therefore does not require the coverage of some other principle-based parsers (e.g. (Merlo, 1995) and (Fong, 1991)). One significant omission in the system presented here is the apparent lack of any technique for capturing the so-called *command* relations typically required for constraining long-distance dependencies, e.g. Subjacency. Roughly speaking, command relations make reference to dominance and precedence configurations in the phrase marker. Such information will apparently be unavailable to our chain system which simply receives a set of relevant nodes found in the phrase structure by the interface relation. There are, however, indexing techniques for labelling the nodes of a parse tree, such that various command relations can be determined without re-traversing the tree (Latecki, 1991) (for further discussion see (Merlo, 1993)). If we assume that the parser or interfacing procedure performs such an indexing on phrase structure, then it is possible to compute the command relations necessary for defining constraints such as Subjacency. It is then straightforward to specify such constraints as strictly local conditions on possible proper links.

2.3 Issues of Control

In the exposition to this point, we have focussed on the logical specification of a licensing grammar, and the decomposition of the system on the basis of the representation type being licensed: phrase structure and chains. To construct theorem provers for our licensing grammar axiomatisations, we first require a mechanism that can generate arbitrary formulae (i.e. trees or chains as defined in (7) and (15)) so that we can then apply the axioms of grammar to determine which trees are grammatical. When cast in these terms, the solution to developing an efficient deductive parser for a principle-based, licensing grammar might seem rather elusive. One crucial aspect of this axiomatisation, however, is precisely the locality property mentioned above. By defining the principles of grammar as conditions on proper branches or proper links, we can recursively define the set of well-formed syntactic trees and chains. That is, we need not define the entire set of trees and chains, and then subsequently restrict this set. Rather, the set of possible syntactic structures is defined exclusively in terms of locally well-formed branches and chain links. Given this characterisation of principle-based grammars we can construct structure generating theorem provers which interleave the process of structure building and licensing by the grammatical principles, on a branch-by-branch (or link-by-link) basis.

Thus far, we have treated these as distinct systems, with the latter presumably invoked after the

former. That is, the top level of the system can be specified in Prolog as follows:

```
(21) parse(String,PS,CS) :-
                phrase-structure(String,PS),
                interface(PS,List),
                chain-structure(List,CS).
```

Clearly, however, such a strategy will be wildly inefficient due to its generate and test nature, although unlike Johnson's (1989) system at least the string will be used to inform the construction of phrase structure (PS). As with Johnson's system, however, we have employed coroutining to achieve much improved performance where the chain structure is recovered in tandem with phrase structure. Such a technique permits us to detect invalid phrase structures (i.e. those which are not theorems of the chain system) as early as possible. For example, assuming no rightward movement of constituents, if a trace is postulated by the parser, but cannot be licensed in a chain, then the parser can immediately backtrack and pursue an alternative path. That is, the parser will not sustain the postulation of traces which don't have a potential antecedent in the current, partially constructed chain structure.

The individual theorem provers also provide natural loci for implementing particular controls strategies without requiring modification of the principles themselves. That is, we can adjust the way candidate structures are proposed by the interpreters, but the constraints which apply to these structures remain unchanged. While any suitable parsing algorithm is in theory possible (modulo the relevant constraints imposed by recursion and empty productions), incremental parsing algorithms are necessary for optimal coroutining. That is, if the chain interpreter traverses the phrase structure tree as it is built, then stack based algorithms which delay attachments (e.g. shift-reduce), will delay the invocation of the chain interpreter. In the present system, our prototype uses a simple recursive descent algorithm for postulating trees, but related work has advocated combining top-down and bottom-up techniques (Crocker, 1994) (Stabler, 1994). Since chains are essentially a one-dimensional construct, we simply construct them linearly, from left to right.

It should be apparent that such a coroutining approach will perform as well as a traditional phrase structure grammar using a gap-threading technique, since both procedures will make use of antecedent information during parsing, to constrain the parser's search. Let us now, however, speculate on other potential control regimes. The present approach of decomposing the parsing task into distinct deductive tasks points to the possibility of using other control strategies such as parallel techniques. Since coroutining effectively simulates synchronous parallel computation, it should be relatively straightforward to implement the parallel counterpart. Indeed this should also be possible for the coroutining model proposed by Johnson (1989). That is, while traditional phrase structure grammars permit parallel computation of multiple, distinct syntactic analyses in parallel, the modular licensing model also permits parallel computation *within* a single syntactic analysis. That is, a principle-based system may be distributed, with separate processes computing different representational levels or types.

Interestingly, however, the approach developed here also presents the possibility of asynchronous parallelisation of the chain system. Once the relevant nodes in the phrase structure tree are identified by the interface procedure, they can simply be handed to an asynchronous chain process. That is, the derivation of chain structure, assuming the indexing technique mentioned above, becomes independent of the actual phrase structure parser. The implementation of such systems remains a matter for future investigation.

3 Conclusion

In this paper we have advanced two principal arguments. Firstly we highlighted a fundamental difference between rule-based and principle-based grammars with regards to deductive parsing. Rule-based grammars may be more accurately termed 'construction-based' grammars, in that an individual rule is *sufficient* to license a particular unit of structure. This one-to-one, rule-to-structure correspondence means that rules may be efficiently used to *propose* candidate structures during parsing. Principle-based grammars, however, fall into a category we have termed 'licensing-grammars' (following

Speas (1990)), whose axiomatisation possesses rather different characteristics. In the case of licensing grammars, principles are typically *necessary* conditions on particular structural configurations, and indeed numerous conditions may apply to a particular unit of structure. Furthermore, none of these principles is particularly suited to the task of proposing candidate structures, since a given constraint is (typically) only defined with respect to some isolated aspect of the structure (i.e. some subset of features). Since the individual principles are not *sufficient* to license some structure, such an axiomatisation also entails the existence of axioms which generate the space of possible structure, such as binary-branching trees, for example.

The second contribution is to suggest that the various informational dependencies involved in current principle-based analyses be decomposed into simple, homogeneous representation types. In so doing, we permit the generalised invocation of the above methodology; that is, just as we define phrase structure as a set of structure licensing conditions over the branches of a binary-branching tree, so can we define chain structure and a set of structure licensing conditions over the links in a chain (or, list). While it is an empirical issue as to whether or not the locality constraints imposed by such a formalisation are sufficiently powerful, they are certainly in the spirit of current proposals in syntactic theory (see Manzini (1992) as an example).

In the context of the decomposed, modular architecture we have constructed, we argue that the generate and test nature of the system can be overcome using coroutining techniques similar to Johnson (1989). It has also been our experience, however, that by separating the grammatical principles along representational lines it is much easier to monitor the informational dependencies among various constraints, a matter of crucial importance to the performance in systems which use the *goal freezing* mechanism which underlies coroutining. The 'decomposition' approach also means that theorem proving strategies can potentially be adapted to individual representation types, in a manner that would be virtually impossible for systems such as Johnson's, and finally, we have suggested that our approach is potentially much more amenable to distribution and parallelisation of the parsing task. Whether or not this turns out to be valuable remains a topic for future research.

Acknowledgments

The author would like to thank Ian Lewin for valuable comments on this work. This paper was written while the author was supported by ESRC research fellowship #H52427000394.

References

[1] Abramson, H. and Dahl, V. (1988). *Logic Grammars*. Symbolic Computation Series. Springer Verlag.

[2] Ades, A. and Steedman, M. (1982). On the Order of Words. *Linguistics and Philosophy*, **4**, 517–558.

[3] Berwick, R. C. (1991). Principle-Based Parsing. In P. Sells, S. Sheiber, and T. Wasow, editors, *Foundational Issues in Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

[4] Berwick, R. C., Abney, S. P., and Tenny, C., editors (1991). *Principle-Based Parsing: Computation and Psycholinguistics*. Studies in Linguistics and Philosophy. Kluwer Academic Publishers, Dordrecht.

[5] Chomsky, N. (1981). *Lectures on Government and Binding*. Foris Publications, Dordrecht.

[6] Crocker, M. W. (1991a). Multiple Interpreters in a Principle-Based Model of Sentence Processing. In *Proceedings of the 5th Conference of the European ACL*, Berlin, Germany.

[7] Crocker, M. W. (1991b). *A Principle-based System for Syntactic Analysis. Canadian Journal of Linguistics*, **3**.

[8] Crocker, M. W. (1992). *A Logical Model of Competence and Performance in the Human Sentence Processor*. Ph.D. thesis, Dept. of Artificial Intelligence, University of Edinburgh, Edinburgh, U.K.

[9] Crocker, M. W. (1994). On the Nature of the Principle-Based Sentence Processor. In

C. Clifton Jr., L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, chapter 11, pages 245–266. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[10] Crocker, M. W. (1996). *Computational Psycholinguistics: An Interdisciplinary Approach to the Study of Language*. Studies in Theoretical Psycholinguistics 20. Kluwer Academic Publishers.

[11] Fong, S. (1991). *Computational Properties of Principle-Based Grammatical Theories*. Ph.D. thesis, MIT, Cambridge, Massachusetts.

[12] Fordham, A. J. and Crocker, M. W. (1997). A Stochastic Government-Binding Parser. In D. B. Jones and H. Somers, editors, *New Methods in Language Processing*, UCL Press, London, UK.

[13] Johnson, M. (1989). Use of Knowledge of Language. *Journal of Psycholinguistic Research*, **18**(1).

[14] Johnson, M. (1991). Program Transformation Techniques for Deductive Parsing. In C. Brown and G. Koch, editors, *Natural Language Understanding and Logic Programming, III*. Elsevier Science Publishers (North-Holland).

[15] Lasnik, H. and Uriagereka, J. (1988). *A Course in GB Syntax: Lectures on Binding and Empty Categories*. Current Studies in Linguistics. MIT Press, Cambridge, Massachusetts.

[16] Latecki, L. (1991). An Indexing Technique for Implementing Command Relations. In *Proceedings of the 5th Conference of the European ACL*, Berlin, Germany.

[17] Manzini, R. (1992). *Locality: A Theory and Some of Its Empirical Consequences*. Linguistic Inquiry Monograph Nineteen. MIT Press, Cambridge, Massachusetts.

[18] Merlo, P. (1993). For an Incremental Computation of Intra-sentential Coreference. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Chambery, France.

[19] Merlo, P. (1995). Modularity and Information Content Classes in Principle-based Parsing. *Computational Linguistics*, **21**(4).

[20] Pereira, F. and Shieber, S. (1987). *Prolog and Natural-Language Analysis*. CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, California.

[21] Pereira, F. and Warren, D. (1980). Definite Clause Grammars for Language Analysis. *Artificial Intelligence*, **13**, 231–278.

[22] Pereira, F. and Warren, D. (1983). Parsing as Deduction. In *Proceedings of Twenty-First Conference of the ACL*, Cambridge, Massachusetts.

[23] Speas, M. (1990). *Phrase Structure in Natural Language*. Studies in Natural Language and Linguistic Theory. Kluwer, Dordrecht.

[24] Stabler, E. P. (1987). Logic Formulations of Government-Binding Principles for Automatic Theorem Provers. Cognitive Science Memo 30, University of Western Ontario, London, Ontario.

[25] Stabler, E. P. (1992). *The Logical Approach to Syntax: Foundations, Specifications, and Implementations of Theories of Government and Binding*. MIT Press, Cambridge, Massachusetts.

[26] Stabler, E. P. (1994). Syntactic Preferences in Parsing for Incremental Interpretation. Unpublished manuscript, UCLA.

On the Impact of Communication Overhead on the Average-Case Scalability of Random Parallel Programs on Multicomputers

Keqin Li
Department of Mathematics and Computer Science
State University of New York, New Paltz, New York 12561-2499, USA
Phone: (914) 257-3534, Fax: (914) 257-3571
E-mail: li@mcs.newpaltz.edu
AND
Yi Pan
Department of Computer Science
University of Dayton, Dayton, Ohio 45469-2160, USA
Phone: (937) 229-3807, Fax: (937) 229-4000
E-mail: pan@cps.udayton.edu

Keywords: bus connected system, communication cost, completely connected system, efficiency, execution time, performance analysis, random parallel program, scalability, speedup

Edited by: Marcin Paprzycki

Received: November 29, 1996 **Revised:** February 28, 1997 **Accepted:** March 20, 1997

A parallel system is a parallel algorithm together with a parallel machine which supports its implementation. Scalability of parallel computing systems describes a property which exhibits performance linearly proportional to the number of processors employed. This paper is devoted to the investigation of average-case scalability of random parallel algorithms executing on multicomputer systems. Such probabilistic analysis is important in revealing the average-case performance of parallel processing, and performance of randomized parallel computations. We study the effect of communication overhead on speedup, efficiency, and scalability of such parallel systems. In particular, we characterize the communication overhead such that the expected efficiency can be kept at certain constant level, and that the number of tasks grows at the rate $\Theta(P \log P)$, where P is the number of processors. We also show that due to load imbalance, $\Theta(P \log P)$ scalability of the number of tasks is the best possible in our probabilistic model of parallel computations. Task granularity and isoefficiency functions are also discussed.

1 Introduction

A parallel system is a parallel algorithm together with a parallel machine which supports its implementation. [1] Scalability of parallel computing systems has been an important research issue in recent years. Though no consensus on the definition of scalability has been reached (Gustafson 1988, Karp & Flatt 1990, Kumar & Gupta 1991, Sarukkai *et al.* 1995, Singh *et al.* 1993, Sun & Rover 1994, Sun & Ni 1993, Zorbas *et al.* 1989), essentially, scalability describes a property which exhibits performance linearly proportional to the number of processors employed. According to the definition in (Kumar & Rao 1987), the scalability of a parallel system is a measure of its capability to increase speedup in proportion to the number of processors, or, its capability to keep a constant efficiency as the number of processors increases. Scalability analysis has many applications in understanding parallel computations. For example, one can predict the performance of a parallel system with a large number of processors from the known performance with few processors.

[1] A preliminary and shorter version of the paper was presented on 1996 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications. Similar parts are reprinted with permission from C.S.R.E.A. Press.

Scalability of many parallel algorithm and machine combinations have been studied extensively. A rich literature exists regarding scalability analysis of parallel algorithms for sorting, Fast Fourier Transform, graph problems, solving partial differential equations, and so on, running on various machines including shared memory multiprocessor, distributed memory multicomputers with buses, rings, meshes, and hypercubes as interconnection networks. The recently published textbook (Kumar *et al.* 1994) is a good source of references in this area.

The present paper is devoted to the investigation of average-case scalability of random parallel algorithms executing on bus connected multicomputer systems and completely connected multicomputers. Such probabilistic analysis is important in revealing the average-case performance of parallel processing, and performance of randomized parallel computations. We study the effect of communication overhead on speedup, efficiency, and scalability of such parallel systems. In particular, we characterize the communication overhead such that the expected efficiency can be kept at certain constant level, and that the number of tasks N grows at the rate $\Theta(P \log P)$, [2] where P is the number of processors. We also show that due to load imbalance, $N = \Theta(P \log P)$ scalability of the number of tasks is the best possible in our probabilistic model of parallel computations. Task granularity and isoefficiency functions are also discussed.

2 Background Information

The performance analysis of a parallel algorithm A implemented on a parallel computer system M is based on the execution time. Let T^* be the execution time of the best sequential algorithm which solves the same problem as A does. T^* is also referred to as the work W (i.e., the amount of basic operations) to solve a problem. Let T_P denote the execution time of A when there are P processors in M. Then the speedup is defined as

$$\text{speedup} = \frac{T^*}{T_P},$$

and the efficiency is

$$\text{efficiency} = \frac{T^*}{T_P P}.$$

The parallel processing time T_P can be divided into two components, namely, computation time (i.e., the work W) and overhead of parallel processing T_o (i.e., communication time, load imbalance, and extra computation), that is,

$$T_P = \frac{W + T_o}{P}.$$

Hence, we can rewrite the efficiency as

$$\text{efficiency} = \left(1 + \frac{T_o}{W}\right)^{-1}.$$

Typically, if P is increased while W is kept constant, the efficiency decreases because T_o increases with P. On the other hand, the efficiency increases with W if P is fixed. Thus, it is possible to maintain efficiency at a fixed level K (or, speedup linearly proportional to P) if W is allowed to increase with P, and this is essentially what a scalable parallel system means. The function

$$W = \left(\frac{K}{1-K}\right) T_o,$$

called the isoefficiency function (Kumar & Rao 1987), determines the growth rate of W required to keep a constant efficiency K as P increases. Small (large) rates imply high (poor) scalability. From another point of view, if we keep the communication cost under certain bound, i.e., $T_o = \psi W$, where

$$\psi = \frac{1-K}{K},$$

then the efficiency K can be maintained, where $0 < K < 1$, and $\psi > 0$.

As pointed out in (Kumar *et al.* 1994), there are essentially three sources of overhead in parallel computations, namely, communication time, load imbalance, and extra computation.

- **Interprocessor Communication.** Any nontrivial parallel system requires communication among processors. The time to transfer information among processors is usually the most significant source of parallel processing overhead.

[2] All logarithms in this paper are base $e = 2.71828....$

- **Load Imbalance.** In many parallel applications, it is difficult to assign exactly the same amount of work load to all processors. When different processors have different work loads, some processors will be idle while others are working. Such idle time contributes to the overhead function T_o.
- **Extra Computation.** It is common that an efficient parallel algorithm is not a straightforward parallelization of the best sequential algorithm for a problem. The amount of work W' performed by a parallel computation is usually larger than the amount of work W done by the best sequential algorithm. The difference $W' - W$ should be considered as part of the overhead function, since it expresses the amount of extra work to be done so that a problem can be solved in parallel.

In this paper, we are mainly interested in the effects of interprocessor communication overhead and load imbalance on the performance of parallel computations. We will assume that there is no extra computation.

3 A Probabilistic Model of Parallel Computation

A multicomputer system M has P processors $M_1, M_2, ..., M_P$. It is assumed that the P processors are homogeneous. There is no global shared memory, and processors can communicate with each other by sending and receiving messages via an interconnection network, which could be a common bus, or a static network. Communications through a network cause significant time delay, which is the major source of overhead of parallel computations on multicomputer systems.

A parallel algorithm (or, program) A executing on a multicomputer system can be represented as a task interaction graph, (or, a static process-based model with both computation and communication costs (Norman & Thanisch 1993)), i.e., a triplet $A = (G, t, c)$. $G = (V, E)$ is an undirected task graph, where $V = \{v_1, v_2, ..., v_N\}$ is a set of tasks (or, processes), and E is a set of edges. An edge $(v_i, v_j) \in E$ means that tasks v_i and v_j need to communicate during the execution of the program A. Task running times (i.e., times for computation only, excluding communication overhead) are given by

$$t : V \to (0, +\infty),$$

such that $t(v_i)$ is the running time of task v_i. Intertask communications are specified by

$$c : E \to (0, +\infty),$$

i.e., $c(v_i, v_j)$ is the amount of information (input data, partial results, etc.) exchanged between tasks v_i and v_j. Note that $c(v_i, v_j) = c(v_j, v_i)$ for all $i \neq j$. For a parallel program to be executed much faster on a multicomputer system than on a sequential machine, intertask communication times should be much less than running times.

We are going to study the scalability of random parallel programs implemented on multicomputer systems. We make the following assumptions:

1. $G = (V, E)$ is a random graph of model A as defined in (Palmer 1985). The sample space consists of all labeled graphs G with N vertices. If G has q edges, the probability of G is given by

   $$\Pr(G) = p_N^q (1 - p_N)^{\binom{N}{2} - q},$$

   where p_N is the probability of an edge. In other words, all the $\binom{N}{2}$ edges occur independently with probability p_N. Usually p_N is a function of N. Obviously, all graphical properties of G are determined by p_N.

2. Task running times $t(v_1), t(v_2), ..., t(v_N)$ are independent and identically distributed (i.i.d.) random variables, for which, we at least know their mean R and variance σ_r^2. It is assumed that the running times have a finite coefficient of variation β_r, i.e., $\sigma_r = \beta_r R$. This assumption is valid for virtually all analytically tractable distributions.

3. Intertask communications $c(v_i, v_j)$, $i \neq j$, under the condition that $(v_i, v_j) \in E$, are i.i.d. random variables, for which, at least their mean C and variance σ_c^2 are available. Again, we assume that C and σ_c^2 are associated as $\sigma_c = \beta_c C$.

4. The random variables $t(v_i)$, $1 \leq i \leq N$, and $c(v_j, v_k)$, $j \neq k$, are uniformly bounded, that is, there exists Z such that $\Pr\{t(v_i) < Z\} = 1$ for all i, and $\Pr\{c(v_j, v_k) < Z\} = 1$ for all $j \neq k$. This is certainly reasonable since no task needs a significant amount of time to finish, and parallel computation cannot afford extraordinary amount of communication.

Thus, our model which was established in (Li & Pan 1996), is a septuple $(P, N, p_N, R, \sigma_r^2, C, \sigma_c^2)$. Our computational model is similar to the model proposed by Indurkhya *et al.* in studying randomly generated distributed programs (Indurkhya *et al.* 1986).

One concern about a random graph G is its connectivity, since a task interaction graph G is usually connected. There is a threshold of p_N for random graph connectivity, namely, $\varrho(\log N/N)$. Graph G has a high probability to be connected if and only if $\varrho > 1$. In particular, we have the following claim (Palmer 1985).

Proposition 1. Let G be a random graph with N vertices, and

$$p_N = \varrho\left(\frac{\log N}{N}\right),$$

where $\varrho > 1$. Then,

$$\Pr\{G \text{ is connected}\} = 1 - O\left(\frac{1}{N^{\varrho-1}}\right).$$

The edge probability p_N given in Proposition 1 implies that $\deg(v)$, the degree of a vertex v, is $\Theta(\log N)$ on the average. Actually, the degrees of all the vertices are very close to this average degree. More specifically, we have the following result from (Palmer 1985).

Proposition 2. Assume that $\epsilon > 0$ is any small quantity. Let G be a random graph with N vertices, and

$$p_N = \varrho\left(\frac{\log N}{N}\right),$$

where $\varrho = \sqrt{3}/\epsilon$. Then,

$$\Pr\{(1-\epsilon)\varrho \log N < \deg(v) < (1+\epsilon)\varrho \log N\} \geq 1 - \frac{O(1)}{\sqrt{\varrho \log N}},$$

for every vertex v.

In this paper, we assume that $p_N = \Omega\left(\frac{\log N}{N}\right)$.

Since the paper is essentially concerned about the effects of communication cost and load imbalance on the scalability, we will ignore the effect of extra computations, which is one of the three factors of parallel processing overhead. Using the above computational model, we have $\mathbf{E}(T^*) = NR$, since the sequential computation does not involve any communication. We are going to represent the mean execution time $\mathbf{E}(T_P)$ as

$$\mathbf{E}(T_P) = \left(\frac{NR}{P}\right)(1 + \phi(N, P, \beta_r, \beta_c, ...)),$$

where ϕ is an expression involving various parameters. Thus, the expected efficiency is

$$\mathbf{E}(\text{efficiency}) = \frac{\mathbf{E}(T^*)}{\mathbf{E}(T_P)P} = \frac{1}{1 + \phi(N, P, \beta_r, \beta_c, ...)}.$$

Basically we describe how N should grow with P so that $\phi(N, P, \beta_r, \beta_c, ...)$ is fixed at a constant ψ. From another point of view, we will give an upper bound for C, the most important parameter in our computational model, such that the scalability function is $N = \Theta(P \log P)$. Note that we scale N against the number of processors P, where N, the number of tasks, is considered as the size of a computation. Also, our scalability analysis is based on mean execution time and mean efficiency. Strictly speaking, we are analyzing the average-case scalability of random parallel programs, as reflected in the title of this paper. However, we still use the term "scalability" for convenience.

4 Computation and Communication times

A task assignment is a partition of V into P disjoint subsets $(V_1, V_2, ..., V_P)$, so that tasks in V_i are assigned to processor M_i, where $1 \leq i \leq P$. Thus, the computation time $T_{\text{comp}}(i)$ of processor M_i is

$$T_{\text{comp}}(i) = \sum_{v_j \in V_i} t(v_j).$$

We consider load balanced parallel computing, that is,

$$|V_1| = |V_2| = \cdots = |V_P| = \frac{N}{P},$$

which is relatively tractable. (The reader is referred to (Monakhov 1991, Norman & Thanisch 1993) and references therein for more discussion and information on mapping parallel tasks to parallel processors.) In other words, the N tasks are evenly distributed among the P processors, such that each processor is assigned N/P tasks, where we assume that P divides N without loss of generality. Proposition 3 (Nicol 1989) essentially states that a perfectly balanced load distribution minimizes the expected maximum computation time of the N processors, i.e., $\mathbf{E}(\max_{1\le i\le P}(T_{\text{comp}}(i)))$.

Proposition 3. Let $Y(n)$ denote the sum of n i.i.d. nonnegative random variables. If $n_i - n_j \ge 2$, then

$$\begin{aligned}&\mathbf{E}(\max(Y(n_1),...,Y(n_i),...,Y(n_j),...,Y(n_P)))\\ &\ge \mathbf{E}(\max(Y(n_1),...,Y(n_i-1),...,\\ &\qquad Y(n_j+1),...,Y(n_P))).\end{aligned}$$

Note that each $T_{\text{comp}}(i)$ is a summation of N/P i.i.d. random variables with mean R and variance σ_r^2. To analyze the random variable $T_{\text{comp}}(i)$, we quote the Central Limit Theorem (Feller 1968), which we state below.

Proposition 4. Let $X_1, X_2, ...,$ be a sequence of independent random variables, where X_i has mean μ_i and variance σ_i^2. If the X_i's are uniformly bounded, and $\sum_{i=1}^{\infty}\sigma_i^2 = \infty$, then, as $n \to \infty$, the summation $X_1 + X_2 + \cdots + X_n$ approaches a normal random variable with mean $\mu_1+\mu_2+\cdots+\mu_n$ and variance $\sigma_1^2+\sigma_2^2+\cdots+\sigma_n^2$.

By Proposition 4, we obtain

Proposition 5. The $T_{\text{comp}}(i)$'s are i.i.d. normal random variables with mean

$$\mathbf{E}(T_{\text{comp}}(i)) = \left(\frac{N}{P}\right)R,$$

and variance

$$\text{Var}(T_{\text{comp}}(i)) = \left(\frac{N}{P}\right)\sigma_r^2,$$

as $N/P \to \infty$.

Message routing between two different processors through an interconnection network incurs substantially more time delay than communication between two processes running on the same processor via a shared memory. For this reason, we assume that the communication time for $c(v_j, v_k)$ is negligible when two tasks v_j and v_k are mapped to the same processor. However, if they are assigned to different processor, then both v_j and v_k spend the same amount of communication time which is proportional to $c(v_j, v_k)$, and depends on the topology of an interconnection network. For example, if the P processors are completely connected, then the communication time $T_{\text{comm}}(i)$ of processor M_i is

$$T_{\text{comm}}(i) = \sum_{\substack{v_j\in V_i, v_k\notin V_i\\ (v_j,v_k)\in E}} c(v_j, v_k),$$

where, for convenience, we assume that the unit of message is normalized such that the time to transmit a message of size c is also c.

Notice that $c(v_i, v_j)$ is defined as the amount of communication between v_i and v_j under the condition that the edge (v_i, v_j) exists. For an arbitrary pair of tasks v_i and v_j, the amount of communication is a random variable $c^*(v_i, v_j)$ which is different from $c(v_i, v_j)$. The following proposition gives its characterizations.

Proposition 6. The amount of communication $c^*(v_i, v_j)$ between arbitrary pair of tasks v_i and v_j, no matter whether $(v_i, v_j) \in E$, is a random variable with expectation $p_N C$ and variance $p_N\sigma_c^2 + (p_N - p_N^2)C^2$.

Proof. Assume that $c(v_i, v_j)$ is a discrete random variable, such that $u_n = \Pr\{c(v_i, v_j) = n\}$, where $n = 1, 2, 3, ...$ Then

$$C = \sum_{n\ge 1} nu_n.$$

The random variable $c^*(v_i, v_j)$ takes values $0, 1, 2, 3, ...,$ where $u_0^* = \Pr\{c^*(v_i, v_j) = 0\} = 1 - p_N$, and $u_n^* = \Pr\{c^*(v_i, v_j) = n\} = p_N u_n$, $n = 1, 2, 3, ...$ Therefore,

$$\begin{aligned}&\mathbf{E}(c^*(v_i, v_j))\\ &= \sum_{n\ge 0} nu_n^*\\ &= \sum_{n\ge 1} np_N u_n\\ &= p_N \sum_{n\ge 1} nu_n\\ &= p_N C,\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{Var}(c^*(v_i,v_j))\\
&=\sum_{n\geq 0} u_n^*(n-p_NC)^2\\
&=(1-p_N)(p_NC)^2\\
&\quad+\sum_{n\geq 1} p_Nu_n(n^2-2p_NCn+(p_NC)^2)\\
&=(1-p_N)(p_NC)^2+p_N\left(\sum_{n\geq 1}u_nn^2\right)\\
&\quad-2p_N^2C\left(\sum_{n\geq 1}u_nn\right)+p_N^3C^2\left(\sum_{n\geq 1}u_n\right)\\
&=(1-p_N)(p_NC)^2+p_N(\sigma_c^2+C^2)\\
&\quad-2p_N^2C^2+p_N^3C^2\\
&=p_N(\sigma_c^2+C^2)-p_N^2C^2.
\end{aligned}$$

In the above derivation, we notice that $\sum_{n\geq 1}u_nn^2$, i.e., the expectation of the second moment of $c(v_i,v_j)$, is $\sigma_c^2+C^2$. ■

Since $T_{\mathrm{comm}}(i)$ is a summation of $\frac{N}{P}\left(N-\frac{N}{P}\right)$ random variables with expectation p_NC and variance $p_N\sigma_c^2+(p_N-p_N^2)C^2$, by Proposition 4, we have

Proposition 7. The $T_{\mathrm{comm}}(i)$'s are normal random variables with mean

$$\mathbf{E}(T_{\mathrm{comm}}(i))=\frac{N^2}{P}\left(1-\frac{1}{P}\right)p_NC,$$

and variance

$$\begin{aligned}
&\mathrm{Var}(T_{\mathrm{comm}}(i))\\
&=\frac{N^2}{P}\left(1-\frac{1}{P}\right)\left(p_N\beta_c^2+p_N-p_N^2\right)C^2,
\end{aligned}$$

as $N/P\to\infty$.

Remark 1. It should be noticed that processor communication times $T_{\mathrm{comm}}(1)$, $T_{\mathrm{comm}}(2)$, ..., $T_{\mathrm{comm}}(P)$, are not truly independent. Consider $T_{\mathrm{comm}}(i)$ and $T_{\mathrm{comm}}(j)$, where $1\leq i\neq j\leq P$. It is clear that $T_{\mathrm{comm}}(i)=X_1+Y$, and $T_{\mathrm{comm}}(i)=X_2+Y$, where X_1 is the communication overhead among M_i and all M_k, where $k\neq j$, X_2 is the communication overhead among M_j and all M_k, where $k\neq i$, and Y is the communication overhead between M_i and M_j. Also, X_1, X_2, and Y are independent of each other; however, $T_{\mathrm{comm}}(i)$ and $T_{\mathrm{comm}}(j)$ are slightly correlated. In particular, the correlation between $T_{\mathrm{comm}}(i)$ and $T_{\mathrm{comm}}(j)$ is

$$\rho(X_1+Y,X_2+Y)=\frac{\mathrm{Cov}(X_1+Y,X_2+Y)}{\sqrt{\mathrm{Var}(X_1+Y)}\sqrt{\mathrm{Var}(X_2+Y)}}.$$

Since

$$\begin{aligned}
&\mathrm{Cov}(X_1+Y,X_2+Y)=\mathrm{Cov}(Y,Y)=\mathrm{Var}(Y),\\
&\mathrm{Var}(Y)=(N/P)^2(p_N(\sigma_c^2+C^2)-p_N^2C^2),\\
&\mathrm{Var}(X_1+Y)=\mathrm{Var}(X_2+Y)\\
&=(N/P)(N-N/P)(p_N(\sigma_c^2+C^2)-p_N^2C^2),
\end{aligned}$$

we have

$$\rho(X_1+Y,X_2+Y)=\frac{1}{P-1}.$$

For large multicomputer systems, this correlation is small (see Appendix 1 for more justification). In this paper, the $T_{\mathrm{comm}}(i)$'s are treated as independent random variables.

5 Completely Connected Systems

In a completely connected system, there is a communication link between each pair of processors. Thus, the execution time of processor M_i is

$$T(i)=T_{\mathrm{comp}}(i)+T_{\mathrm{comm}}(i),$$

and the overall execution time is

$$T_P=\max\,(T(1),\,T(2),\,\ldots,\,T(P)).$$

In a balanced task assignment, each $T(i)$ is the summation of

$$\frac{N}{P}+\frac{N}{P}\left(N-\frac{N}{P}\right)$$

random variables, with the first N/P described in Proposition 5, and the last $\frac{N}{P}\left(N-\frac{N}{P}\right)$ described in Proposition 7. By Proposition 4, $T(i)$ is roughly a normal random variable with mean

$$\left(\frac{N}{P}\right)R+\left(\frac{N^2}{P}\right)\left(1-\frac{1}{P}\right)p_NC,$$

and variance

$$\left(\frac{N}{P}\right)\sigma_r^2+\left(\frac{N^2}{P}\right)\left(1-\frac{1}{P}\right)(p_N\beta_c^2+p_N-p_N^2)C^2.$$

To study $\mathbf{E}(T_P)$, we need the the following result from order statistics (Ang & Tang 1984, Arnold *et al.* 1992).

Proposition 8. Let $X_1, X_2, ..., X_n$ be i.i.d. normal random variables with mean μ and variance σ^2. Then,

$$\mathbf{E}(\max(X_1, X_2, ..., X_n)) = \mu + B_n\sigma,$$

with

$$\begin{aligned} B_n &\approx \sqrt{2\log n} - \frac{\log\log n + \log 4\pi - b\gamma}{2\sqrt{2\log n}} \\ &= \Theta(\sqrt{\log n}\,), \end{aligned}$$

where $\gamma = 0.5772156649...$ is Euler's constant, and b is a small constant.

Remark 2. There is no closed form expression for B_n. However, by choosing b appropriately, the B_n given in Proposition 8 is very accurate. Appendix 2 provides more detailed information.

By Proposition 8, the expected parallel execution time is approximately

$$\begin{aligned} \mathbf{E}(T_P) \approx \left(\frac{N}{P}\right)R + \left(\frac{N^2}{P}\right)\left(1-\frac{1}{P}\right)p_NC \\ +B_P\left[\left(\frac{N}{P}\right)\sigma_r^2 + \left(\frac{N^2}{P}\right)\left(1-\frac{1}{P}\right)\right. \\ \left.(p_N\beta_c^2 + p_N - p_N^2)C^2\right]^{1/2}. \end{aligned}$$

Since

$$\frac{P-1}{P^2}$$

is a decreasing function for $P \geq 2$, so is the mean execution time $\mathbf{E}(T_P)$, i.e., more processors require less processing time.

For scalability, we assume that

$$C = \alpha\left(\frac{R}{p_NN}\right).$$

Thus, we have

$$\begin{aligned} \mathbf{E}(T_P) \approx \frac{NR}{P}\left[1+\alpha\left(1-\frac{1}{P}\right)+B_P\left(\beta_r^2\left(\frac{P}{N}\right)\right.\right. \\ \left.\left.+\alpha^2\left(\frac{\beta_c^2+1-p_N}{p_N}\right)\left(\frac{P-1}{N^2}\right)\right)^{1/2}\right]. \end{aligned}$$

As $P \to \infty$, to maintain certain efficiency level K, approximately we need

$$\alpha + \beta_r B_P\sqrt{\frac{P}{N}} = \psi,$$

which implies that

$$N = \left(\frac{\beta_r}{\psi-\alpha}\right)^2 B_P^2P = \Theta(P\log P).$$

Note that there is an upper bound on the mean efficiency, which is

$$\frac{1}{1+\alpha}.$$

To remove this bound we should have communication cost $C = o\left(\frac{R}{p_NN}\right)$. However, the scalability function is still $N = \Theta(P\log P)$. In particular, one can verify that if

$$C = \alpha\left(\frac{R}{p_NNw_N}\right),$$

where w_N is any slowly growing infinity, then,

$$N = \left(\frac{\beta_r}{\psi}\right)^2 B_P^2P.$$

The above discussion also shows that if

$$C = \alpha\left(\frac{Rw_N}{p_NN}\right),$$

where w_N is any infinity, then the parallel system is not scalable, i.e., it is impossible to maintain a constant efficiency when P grows.

The characterization of C implies that $R = \Theta(p_Nw_NNC)$. If p_N is known, e.g., $p_N = \Theta\left(\frac{\log N}{N}\right)$ as given in Propositions 1 and 2, then the growth rate of R could be found, which is

$$R = \Theta((\log P + \log\log P)w_PC),$$

where, we use w_P instead of $w_{P\log P}$, since any infinity of N or P suffices. Furthermore, we can obtain the growth rate of expected work $W = \Theta(NR)$, which relates to the (expected) isoefficiency functions, i.e.,

$$W = \Theta(P(\log P)^2w_PC).$$

6 Bus Connected Systems

In a bus connected system, the P processors send and receive messages over a common shared bus. At any moment, only one processor can send message to another processor. Thus, the total communication overhead T_{comm} is the summation

$$T_{\text{comm}} = \frac{1}{2}\sum_{i=1}^{P} T_{\text{comm}}(i),$$

and by Proposition 7,

$$\begin{aligned}\mathbf{E}(T_{\text{comm}}) &= \frac{1}{2}\sum_{i=1}^{P}\mathbf{E}(T_{\text{comm}}(i)) \\ &= \frac{1}{2}p_N C N^2\left(1-\frac{1}{P}\right).\end{aligned}$$

$\mathbf{E}(T_{\text{comm}})$ given above is also the mean communication overhead incurred by each individual processor.

The computation time T_{comp} is

$$T_{\text{comp}} = \max\,(T_{\text{comp}}(1),\ T_{\text{comp}}(2),\ \ldots,\ T_{\text{comp}}(P)).$$

By Proposition 5, the $T_{\text{comp}}(i)$'s are approximately i.i.d. normal random variables. Using Proposition 8, we have

$$\mathbf{E}(T_{\text{comp}}) = \left(\frac{N}{P}\right)R + B_P\sqrt{\frac{N}{P}}\,\sigma_r.$$

Thus, the expected overall execution time is

$$\begin{aligned}&\mathbf{E}(T_P) \\ &= \mathbf{E}(T_{\text{comp}}) + \mathbf{E}(T_{\text{comm}}) \\ &= \left(\frac{N}{P}\right)R + B_P\sqrt{\frac{N}{P}}\,\sigma_r + \frac{1}{2}p_N C N^2\left(1-\frac{1}{P}\right).\end{aligned}$$

Remark 3. In a bus connected system, a perfectly balanced load distribution actually maximizes the total communication overhead

$$\begin{aligned}&\mathbf{E}(T_{\text{comm}}) \\ &= \frac{1}{2}\sum_{i=1}^{P}\mathbf{E}(T_{\text{comm}}(i)) \\ &= \frac{1}{2}\sum_{i=1}^{P}|V_i|(N-|V_i|)p_N C \\ &= \frac{1}{2}(N^2-|V_1|^2-|V_2|^2-\cdots-|V_P|^2)p_N C.\end{aligned}$$

Whether this kind of task assignment optimizes the expected overall execution time $\mathbf{E}(T_P)$ is still an open question, though certain attempt has been made (Indurkhya *et al.* 1986, Nicol 1989).

If we regard $\mathbf{E}(T_P)$ as a function of P, then $\mathbf{E}(T_P)$ takes its smallest value only at $P = 1$ or $P = N$. (A result similar to this has been observed in (Indurkhya *et al.* 1986); however, it was proved under special assumptions (Nicol 1989).) To see this, let us consider the derivative $d\mathbf{E}(T_P)/dP$. We assume that $B_P \approx \sqrt{2\log P}$ for simplicity. Thus,

$$\frac{d\mathbf{E}(T_P)}{dP} \approx -\frac{1}{P^2}\left[NR - \frac{1}{2}p_N C N^2 + \sigma_r\sqrt{\frac{N}{2}}\sqrt{P\log P}\right].$$

1. When
$$C < \frac{2R}{p_N N},$$
that is, $NR > p_N C N^2/2$, it is always true that $d\mathbf{E}(T_P)/dP < 0$, which implies that $\mathbf{E}(T_P)$ is a decreasing function of P, i.e., more processors take less execution time.

2. When
$$C > \frac{2R}{p_N N} + \frac{\sqrt{2\log N}}{p_N N}\sigma_r,$$
that is,
$$NR + \sigma_r\sqrt{\frac{N}{2}}\sqrt{P\log P} < \frac{1}{2}p_N C N^2,$$
we always have $d\mathbf{E}(T_P)/dP > 0$, i.e., $\mathbf{E}(T_P)$ is an increasing function of P. In other words, there is no benefit to use more processors because of large communication cost.

3. When C is in the following small interval,
$$\frac{2R}{p_N N} \le C \le \frac{2R}{p_N N} + \frac{\sqrt{2\log N}}{p_N N}\sigma_r,$$
there is a solution to the equation $d\mathbf{E}(T_P)/dP = 0$, which implies that $\mathbf{E}(T_P)$ first increases as P increases, and after certain value P^*, $\mathbf{E}(T_P)$ starts to decrease, i.e., $\mathbf{E}(T_P)$ takes its maximum value at P^*. By adjusting C, $\mathbf{E}(T_P)$ could take its maximum value at all possible points $P = 1, 2, 3, \ldots, N$.

The actual behavior of $\mathbf{E}(T_P)$ is slightly different from what mentioned above due to the approximation of B_P in the discussion. For example, when

$$C = \alpha\left(\frac{R}{p_N N}\right),$$

if we want $\mathbf{E}(T_P)$ to be a decreasing function of P, then in addition to the condition $\alpha < 2$, we also require $P \geq 3$.

To discuss the scalability, we assume that

$$C = \alpha\left(\frac{R}{p_N N f(N)}\right),$$

where $f(N)$ is bounded above by $O(N)$. Otherwise, the total communication cost would be

$$\frac{1}{2}N(N-1)p_N C \approx \frac{\alpha R}{2}\left(\frac{N}{f(N)}\right) = o(1),$$

i.e., it vanishes to zero as N becomes large. Then,

$$\mathbf{E}(T_P) = \frac{NR}{P}\left[1 + \beta_r B_P\sqrt{\frac{P}{N}} + \frac{\alpha}{2}\left(\frac{P-1}{f(N)}\right)\right].$$

To keep the mean efficiency at a constant value K, it suffices to have

$$\beta_r B_P\sqrt{\frac{P}{N}} + \frac{\alpha}{2}\left(\frac{P-1}{f(N)}\right) = \psi.$$

The scalability depends only on $f(N)$. Larger growth rate of $f(N)$ implies higher scalability. We consider several typical cases.

1. $f(N) = \Theta(1)$. That is, $f(N)$ is a constant δ. As P becomes large, especially when $P > 2\psi\delta/\alpha + 1$, the equality is no longer valid, no matter how we choose N. Therefore, the system is not scalable.

2. $f(N) = (\log N)^m$. Clearly, as N gets large, $1/(\log N)^m$ dominates $1/\sqrt{N}$. Thus, roughly we need

   $$\frac{\alpha}{2}\left(\frac{P}{(\log N)^m}\right) = \psi,$$

   that is, $N = \Theta(b^{\sqrt[m]{P}})$, where

   $$b = \exp\left(\left(\frac{\alpha}{2\psi}\right)^{1/m}\right).$$

   Since N should grow exponentially in P, the system exhibits a poor scalability. In general, if $f(N)$ is a slowly growing infinity, then we have

   $$N = f^{-1}\left(\frac{\alpha P}{2\psi}\right).$$

3. $f(N) = N^\epsilon$, where $0 < \epsilon < 1$. Again, we only need to consider

   $$\frac{\alpha}{2}\left(\frac{P}{N^\epsilon}\right) = \psi,$$

   which gives

   $$N = \left(\frac{\alpha P}{2\psi}\right)^{1/\epsilon} = \Theta(P^{1/\epsilon}).$$

   The scalability is determined by ϵ. For small values of ϵ, a polynomial with a large degree is still considered as impractical. However, when $\epsilon \geq 0.5$, $N = O(P^2)$ which is reasonable.

4. $f(N) = N$, i.e.,

   $$C = \alpha\left(\frac{R}{p_N N^2}\right).$$

   Now we need to have

   $$\beta_r B_P\sqrt{\frac{P}{N}} = \psi,$$

   that is, $N \sim (\beta_r/\psi)B_P^2 P = \Theta(B_P^2 P) = \Theta(P\log P)$, which means a good scalability. In particular, the following equation

   $$\frac{\alpha}{2}\left(\frac{P-1}{N}\right) + \beta_r B_P\sqrt{\frac{P}{N}} - \psi = 0,$$

   namely,

   $$\psi N - \beta_r B_P\sqrt{P}\sqrt{N} - \left(\frac{\alpha}{2}\right)(P-1) = 0$$

   gives

   $$N = \left[\frac{1}{2\psi}\left(\beta_r B_P\sqrt{P} + \sqrt{\beta_r^2 B_P^2 P + 2\alpha\psi(P-1)}\right)\right]^2.$$

Finally, it is easy to verify that in case 4,

$$R = \Theta(P\log P(\log P + \log\log P)C),$$

and the isoefficiency function is

$$W = \Theta(P^2(\log P)^3 C),$$

where, we assume that $p_N = \Theta\left(\frac{\log N}{N}\right)$.

Notice that the ratio R/C is called task granularity (Stone 1990), which measures the size of an individual task to be executed on a processor. In coarse-grain parallelism (e.g., scalable computations on bus connected systems), R/C is relatively high, so that each task requires a relatively small amount of communication; while in fine-grain parallelism (e.g., scalable computations on completely connected systems), R/C is relatively low, so each task can incur relatively large communication overhead.

7 A Lower Bound

It should be noticed that $\Theta(P \log P)$ is the best possible scalability of N in our probabilistic computational model for random parallel algorithms on multicomputers. The growth rate of B_P is a key to the $\Theta(P \log P)$ scalability of the number of tasks N that have been obtained in Sections 5 and 6. From this statistical property of random variables, we can derive the following lower bound for the scalability of N.

Proposition 9. $\Theta(P \log P)$ is the best possible growth rate of N if a constant efficiency is to be maintained.

Proof. Among the three factors of parallel processing overhead, let us focus on the effect of load imbalance. We consider the case when there is no communication overhead, i.e., $c(v_i, v_j) = 0$ for all i and j, and there is no extra computation. There are NR amount of computation to be performed by P processors. Now, we consider a perfectly balanced load assignment, i.e., the $T_{\text{comp}}(i)$'s are i.i.d. normal random variables with mean $(N/P)R$ and $(N/P)\sigma_r^2$ (cf. Proposition 5). The overall parallel execution time of the entire program is $T_P = \max_{1 \le i \le P}(T_{\text{comp}}(i))$. From Proposition 3, $\mathbf{E}(T_P)$ is optimized. Notice that the above balanced load distribution only implies that processors receive the same number of tasks. The execution times, i.e., $T_{\text{comp}}(1)$, $T_{\text{comp}}(2)$, ..., $T_{\text{comp}}(P)$, are not necessarily the same due to statistical fluctuations of task running times. By Proposition 8,

$$\begin{aligned}\mathbf{E}(T_P) &= \left(\frac{N}{P}\right) R + B_P \sqrt{\frac{N}{P}}\,\sigma_r \\ &= \frac{NR}{P}\left(1 + \beta_r B_P \sqrt{\frac{P}{N}}\right),\end{aligned}$$

and from which, we easily see that

$$N = \left(\frac{\beta_r}{\psi}\right)^2 B_P^2 \log P = \Theta(P \log P),$$

so that a constant efficiency is kept. ■

Since communication costs and mechanisms are not considered, the above discussion and lower bound are applicable to multiprocessor systems with shared memories. Also notice that the lower bound is valid only in our probabilistic model, i.e., for those computations which can be broken into numerous interacting components of similar characteristics.

8 Summary and Further Research

The main contributions of the paper are summarized as follows.

- First, we establish a model of random parallel programs executing on multicomputer systems supporting message passing based on random task interaction graphs.
- Second, using tools like the Central Limit Theorem and results from order statistics, we derive mean execution times of random parallel programs on multicomputers with bus connections and completely connected networks.
- Third, we characterize the maximum communication overhead allowed in such a way that constant efficiency could be maintained, and that the number of tasks has a growth rate $N = \Theta(P \log P)$.
- Fourth, we derive isoefficiency functions for these systems.
- Finally, we prove that due to load imbalance caused by statistical fluctuations of task running times, $N = \Theta(P \log P)$ is the best possible scalability for N in our probabilistic model of parallel computing.

The results obtained in this paper provide certain insights into the nature of parallel computations in average-cases. To the best of the authors' knowledge, such probabilistic scalability analysis has rarely been conducted before.

It is noticed that in recent years, networks of workstations (Efe & Krishnamoorthy 1995, Zhang 1996) and cluster computing (Turcotte 1996) are becoming the major computing infrastructure for science and engineering due to rapid development in networking hardware and software technologies. An important property of these systems is that a network provides the same communication bandwidth to all pairs of processors. Hence, our analytical results in this paper are directly applicable to these systems.

Nevertheless, parallel computing on traditional message-passing systems with point-to-point direct interconnection networks is still of fundamental importance (Stojmenović 1996). In these multicomputers, processors communicate by sending/receiving messages that should traverse communication links. Hence, the communication overhead is not just in proportion to the amount of information transmitted, but also distances among the processors. Link contention and traffic delay have tremendous effects on system performance. It is conceivable that scalability analysis for multicomputer systems with static interconnection networks such as meshes and hypercubes will be considerable more involved than what we have done for bus connected systems and completely connected systems. It is likely that more mathematical tools like queueing theory (Kleinrock 1976) are required. Since the appearance of an earlier version of this paper (Li & Pan 1996), there has been growing interest in average-case scalability analysis for static networks. Though some progress has been made in this direction (Li *et al.* 1997), much work remain to be done.

Appendix 1

As mentioned in Remark 1, the interprocessor communication times are not independent of each other; they are positively correlated. However, it turns out that the effect of such correlation on the quality of our analysis of $\mathbf{E}(T_P)$ is negligible.

Extensive computer simulations have been conducted to justify the above claim. Table 1 demonstrates a typical set of experimental data. Let P be in the range [10..40]. Assume that

$$N = \lceil 2 \log P \rceil P,$$

and

$$p_N = 5\left(\frac{\log N}{N}\right).$$

The values of P, N and p_N are given in the first three columns.

Table 1. Expected maximum interprocessor communication time.

P	N	p_N	Sim.	C. I.	Ana.	Error
10	50	0.391	1479	0.096%	1497	-1.219%
12	60	0.341	1570	0.107%	1608	-2.326%
14	84	0.264	2010	0.342%	2104	-4.476%
16	96	0.238	2170	0.229%	2197	-1.239%
18	108	0.217	2351	0.276%	2277	3.236%
20	120	0.200	2494	1.484%	2349	6.176%
22	154	0.164	2790	0.420%	2874	-2.942%
24	168	0.153	2879	0.845%	2941	-2.095%
26	182	0.143	2981	1.261%	3002	-0.697%
28	196	0.135	3072	1.103%	3058	0.485%
30	210	0.127	3092	1.134%	3109	-0.557%
32	224	0.121	3140	0.528%	3157	-0.543%
34	272	0.103	3674	1.398%	3715	-1.100%
36	288	0.098	3842	0.859%	3763	2.122%
38	304	0.094	3904	1.179%	3807	2.547%
40	320	0.090	4075	1.097%	3850	5.854%

For each set of P, N, and p_N, one hundred random parallel programs were generated. The inter-task communication overhead is characterized by a uniform distribution in $[a, b]$, where $a = 10$, and $b = 20$. Such a distribution yields $C = 15$, and $\sigma_c^2 = 10$.

For each program, the maximum communication cost over all the P processors, i.e.,

$$\max_{1 \le i \le P} (T_{\text{comm}}(i))$$

was calculated. The 100 instances of the maximum communication cost were summarized using the standard method in statistics (Freund 1981). The fourth column shows the expectation of the maximum communication cost obtained from simulations (the data have been rounded to integers for clarity), and the fifth column gives

the 99.9999998% confidence interval, which is less than 1.5%. The sixth column provides our analytical results (rounded to integers), and the last column shows the relative error of the simulation results as compared to the analytical results. The data in the last column is very encouraging, since the relative error is typically within ±5%.

Appendix 2

To analyze the mean parallel execution time $\mathbf{E}(T_P)$, it is necessary to know the value of B_P. Proposition 8 shows that

$$B_P \approx \sqrt{2\log P} - \frac{\log\log P + \log 4\pi - b\gamma}{2\sqrt{2\log P}},$$

where b is a small constant. For P in the range [5..100], which is the typical size for the current multicomputer systems, we suggest $b = 1.436$, which is selected such that our estimation is most accurate for $P = 50$. The value of B_P given by the above equation is listed in Table 2, where the exact value of B_P (taken from (Arnold *et al.* 1992)) is also listed for the purpose of comparison. The data in Table 2 show that the value of B_P given by our closed form estimation is quite accurate.

Table 2. The value B_P.

P	Estimation	Exact Value
5	1.18713	1.16296
10	1.55505	1.53875
15	1.74752	1.73591
20	1.87593	1.86747
25	1.97147	1.96531
30	2.04715	2.04276
35	2.10959	2.10661
40	2.16259	2.16078
45	2.20855	2.20772
50	2.24906	2.24907
55	2.28523	2.28598
60	2.31788	2.31928
65	2.34760	2.34958
70	2.37487	2.37736
75	2.40003	2.40299
80	2.42339	2.42677
85	2.44517	2.44894
90	2.46556	2.46970
95	2.48473	2.48920
100	2.50282	2.50759

Acknowledgment

The authors are grateful to the editor and three anonymous referees for their constructive and helpful comments. Keqin Li's research was supported by a SUNY-New Paltz/IBM joint study under Agreement No. 42150052, and also funded by National Aeronautics and Space Administration and the State University of New York through the NASA/University Joint Venture in Space Science Program under Grant NAG8-1313, and the 1996 NASA/ASEE Summer Faculty Fellowship Program. He appreciates IBM Thomas J. Watson Research Center, and the Center of Excellence in Space Data and Information Sciences of Universities Space Research Association at NASA Goddard Space Flight Center for providing necessary facilities and resources to conduct this research. Yi Pan's work was supported in part by the National Science Foundation under Grants CCR-9211621 and CCR-9503882, the Air Force Avionics Laboratory, Wright Laboratory, Dayton, Ohio, under Grant F33615-C-2218, and an Ohio Board of Regents Investment Fund Competition Grant.

References

[1] Ang A. H.-S. & Tang W. H. (1984) *Probability Concepts in Engineering Planning and Design, Volume II - Decision, Risk, and Reliability.* John Wiley & Sons.

[2] Arnold B. C., Balakrishnan N. & Nagaraja H. N. (1992) *A First Course in Order Statistics.* John Wiley & Sons.

[3] Efe K. & Krishnamoorthy V. (1995) Optimal Scheduling of Compute-Intensive Tasks on a Network of Workstations. *IEEE Transactions on Parallel and Distributed Systems*, 6, 6, p. 668-673.

[4] Feller W. (1968) *An Introduction to Probability Theory and Its Applications, Volume I*, 3rd edition. John Wiley & Sons.

[5] Freund J. E. (1981) *Statistics: A First Course*, 3rd edition. Prentice-Hall.

[6] Gustafson J. L. (1988) Reevaluating Amdahl's Law. *Communications of the ACM*, 31, p. 532-533.

[7] Indurkhya B., Stone H. S. & Xi-Cheng L. (1986) Optimal Partitioning of Randomly Generated Distributed Programs. *IEEE Transactions on Software Engineering*, 12, 3, p. 483-495.

[8] Karp A. H. & Flatt H. P. (1990) Measuring Parallel Processor Performance. *Communications of the ACM*, 33, p. 539-543.

[9] Kleinrock L. (1976) *Queueing Systems. Volume 2: Computer Applications.* John Wiley & Sons.

[10] Kumar V., Grama A., Gupta A. & Karypis G. (1994) *Introduction to Parallel Computing.* Benjaming/Cummings.

[11] Kumar V. & Gupta A. (1991) Analysis of Scalability of Parallel Algorithms and Architectures: a Survey. *Proc. International Conference on Supercomputing*, p. 396-405.

[12] Kumar V. & Rao V. N. (1987) Parallel Depth First Search. Part II. Analysis. *International Journal of Parallel Programming*, 16, p. 501-519.

[13] Li K. & Pan Y. (1996) Characterizations of Communication Overhead for Scalable Random Parallel Algorithms on Multicomputer Systems. *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, p. 474-485, Sunnyvale, California, August 1996.

[14] Li K., Pan Y., Shen H. & Zheng S.-Q. (1997) A Study of Average-Case Speedup and Scalability of Parallel Computations on Static Networks. Working Paper #97-03, Center for Business and Economic Research (CBER), University of Dayton, Dayton, Ohio, USA.

[15] Monakhov O. G. (1991) Parallel Algorithm for Mapping of Program Graphs into Parallel Computers. *Proceedings of International Conference on Parallel Computing Technologies*, p. 471-476, Novosibirsk, Russia.

[16] Nicol D. M. (1989) Optimal Partitioning of Random Programs Across Two Processors. *IEEE Transactions on Software Engineering*, 15, 2, p. 134-141.

[17] Norman M. G. & Thanisch P. (1993) Models of Machines and Computation for Mapping in Multicomputers. *ACM Computing Surveys*, 25, 3, p. 263-302.

[18] Palmer E. M. (1985) *Graphical Evolution.* John Wiley & Sons.

[19] Sarukkai S. R., Mehra P. & Block R. J. (1995) Automated Scalability Analysis of Message-Passing Parallel Programs. *IEEE Parallel and Distributed Technology*, 3, 4, p. 21-32.

[20] Singh J. P., Hennessy J. L. & Gupta A. (1993) Scaling Parallel Programs for Multiprocessors: Methodology and Examples. *Computer*, 26, p. 42-50.

[21] Stojmenović I. (1996) Direct Interconnection Networks. In *Parallel and Distributed Computing Handbook*, Zomaya A. Y. H. ed., p. 537-567, McGraw-Hill.

[22] Stone H. S. (1990) *High-Performance Computer Architecture*, 2nd edition. Addison-Wesley.

[23] Sun X.-H. & Rover D. T. (1994) Scalability of Parallel Algorithm-Machine Combinations. *IEEE Transactions on Parallel and Distributed Systems*, 5, 6, p. 599-613.

[24] Sun X.-H. & Ni L. M. (1993) Scalable Problems and Memory-Bounded Speedup. *Journal of Parallel and Distributed Computing*, 19, p. 27-37.

[25] Turcotte L. H. (1996) Cluster Computing. In *Parallel and Distributed Computing Handbook*, Zomaya A. Y. H. ed., p. 762-779, McGraw-Hill.

[26] Zhang X. (1996) Coordinating Parallel Tasks Across Networks of Workstations. *Proceedings of 8th International Conference on Parallel and Distributed Computing and Systems*, p.6, Chicago, Illinois, USA, October 1996.

[27] Zorbas J. R., Reble D. J. & VanKooten R. E. (1989) Measuring the Scalability of Parallel Computer Systems. *Proc. Supercomputing '89*, p. 832-841.

Algorithms in the Method of Paired Comparisons

George J. Miel and P. Diane Turnbough
University of Nevada, Las Vegas, NV 89154, U.S.A.
Phone: 702-895-0360 and 702-895-0182, Fax: 702-4343
E-mail: miel@cs.unlv.edu

Keywords: Rank Deficient Least Squares, Graphs, Pairwise Comparisons, Psychometrics

Edited by: Marcin Paprzycki
Received: October 15, 1996 **Revised:** March 12, 1997 **Accepted:** May 1, 1997

A description is given of a computational model for ranking elements of a given set based on pairwise comparisons of the elements when the set is large and the comparisons are unstructured. The model involves a large sparse overdetermined linear system Cx=d, where C is the mxn incidence matrix of a graph with n nodes (elements of the set) and m arcs (paired comparisons) and where d is the vector of observed differences in worth. Under the assumption that the graph has q connected components, simple algorithms are given for computing efficiently the corresponding least squares estimation in terms of a maximum of q nonsingular dense systems the sum of whose dimensions is bounded by n.

1 Introduction

A paradigm for ranking elements of a given set involves pairwise comparisons of the elements, each consisting of a measured or perceived estimate of the difference or ratio of worths of the two items, followed by a statistical or mathematical framework for estimating the worths, which are then sorted in order to obtain a ranking (Cook & Kress 1992). The method has roots in experimental psychology going back at least to 1860 in an account by Fechner of his extensive experiments examining Weber's law (Fechner 1860, batschelet 1973). Statistical foundations of the method were built in the 1950's by several researchers (mosteller 1951, Kendall 1955, David 1963). Examples of models based on the method of paired comparisons include the Bradley-Terry model, based on the logistic distribution and related to the choice axiom of Luce, and the Analytic Hierarchy Process presented by Saaty, based on eigenanalysis of reciprocal matrices (Bradley & Terry 1952, Luce 1959, Saaty 1977).

In addition to psychology, the method of paired comparisons has applications in diverse disciplines such as sports, finance, economics, and policy making (Saaty & Varga 1991, Cook & Kress 1992). A prototype example consists of rank ordering the participants of one or more chess tournaments in order to reflect accurately the performance and ability of the players. Applications using pairwise comparisons are usually performed when the set of items is either small or in some sense homogeneous and under controlled conditions to ensure that the comparisons are balanced, e.g., in incomplete block designs or linked paired-comparison designs (David 1963).

Due to high performance computers, the scope of the method can be considerably enlarged to the case when the set of items is very large and the paired comparisons are highly unstructured or even randomly selected. We outline a model targeted for this case, using a difference scale to measure relative worths of pairs of items, based on graph-theoretic considerations and on rank deficient least squares estimation. [1] Paradoxically, although the problem appears highly unstructured, the resulting linear system is in fact structured and remarkably simple algorithms lead to an efficient solution.

[1] Parts of the results were presented at the Joint Statistical Meetings in Chicago, Illinois, on August 7, 1996.

2 Theorem

Consider a graph with n nodes and m directed arcs, with $m \gg n \gg 0$, in which each node represents an item with a numerical measure of worth attached to it, and where an arc from node j to node k means that the two items have been pairwise compared. Each row of the mxn incidence matrix C has exactly two nonzero entries, with -1 in column j and +1 in column k indicating that there is an arc from node j to node k. If x_j is the worth of the jth item and d_i denotes the observed difference in worth attached to the ith paired comparison then the mxn overdetermined system

$$Cx = d \tag{1}$$

asks to estimate the worths $x_1, \cdots, x_n$ knowing pairwise differences of observed performances represented by the arcs of the graph. From graph theory (Carré 1979), we know that the incidence matrix C has rank n-q where q is the number of connected components of the graph. It follows that in the corresponding nxn normal system

$$Ax = b \tag{2}$$

the matrix $A = C^tC$ also has rank n-q. Diverse methods are available (Golub & Van Loan 1989) for finding the general least squares solution of the system (1). However the computation can be carried out more efficiently by solving a maximum of q smaller and nonsingular systems the sum of whose dimensions is bounded by n.

The case with q=1, when the graph is fully connected, provides a building block for our procedure. This case can be handled as follows: replace anyone equation of the nxn normal system (2) by an appropriate equation

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n = c$$

thus getting a full-rank nxn auxilliary system

$$\widehat{A}x = \hat{b} \tag{3}$$

whose unique solution can then be used to get the general solution.

Theorem. *If the graph is connected and the auxilliary equation satisfies* $c_1 + c_2 + \cdots + c_n \neq 0$ *then the matrix* $\widehat{A}$ *is nonsingular and the set of least squares solution of* $Cx = d$ *is given by*

$$x = \widehat{A}^{-1}\hat{b} + se \tag{4}$$

where $e = (1, 1, \cdots, 1)^t$ *and* s *is an arbitrary scalar.*

Proof. An outline follows. Since A has rank n-1, its nullity is 1. The n columns of A sum to the zero n-vector. Thus the general solution of Ax=b is

$$x = x^p + se \tag{5}$$

where x^p is any one particular solution. In fact, (5) is the general solution to the (n-1)xn system obtained by deleting any one equation of the normal system. The addition of the auxilliary equation yields the nxn system $\widehat{A}x = \hat{b}$. This auxilliary system is equivalent to the system

$$\begin{array}{llllll} x_1 & & & -s & = x_1^p \\ & x_2 & & -s & = x_2^p \\ & & \ddots & & \\ & & x_n & -s & = x_n^p \\ c_1x_1 + & c_2x_2 + & \cdots + & c_nx_n & & = c \end{array}$$

of n+1 equations in the n+1 unknowns $x_1, x_2, \cdots, x_n, s$. Consider the determinant of the coefficient matrix. For $1 \leq i \leq n$, subtract c_i times row i from the $(n+1)$th row to show that this determinant is $\gamma = c_1 + \cdots + c_n$. The condition $\gamma \neq 0$ implies that the auxilliary system is uniquely solvable. The desired result follows.

The general solution (4) consists of the null space of A, represented by the arbitrary vector $x^h = se$, translated by a particular solution $x^p = \widehat{A}^{-1}\hat{b}$. The arbitrary vector x^h shows that since the general solution depends on observed differences of worths it is invariant to an additive shift. In the general case with rank n-q, there will be q linearly independent arbitrary vectors, each corresponding to an additive shift in the worths of items in a connected component.

3 Algorithms

A simple data structure for storing the incidence matrix C consists of two m-vectors MINUS and PLUS whose jth elements denote the column index of -1 and +1 respectively in the jth row of

C. A key to efficient computation of $A = C^tC$ and $b = C^td$ is the representation of matrix products by outer products of vectors. Given an mxn matrix M and an nxp matrix N, then

$$MN = \mathrm{col}_1(M)\mathrm{row}_1(N) + \cdots + \mathrm{col}_n(M)\mathrm{row}_n(N)$$

where each outer product $\mathrm{col}_j(M)\mathrm{row}_j(N)$ is an mxp matrix. Use of the above formula yields a remarkably simple algorithm for evaluating the matrix $A = C^tC$:

$a_{ik} := 0 \quad (i,k = 1,2,\cdots,n)$
For $j = 1,2,\cdots,m$ do :
$i :=$ MINUS(j)
$k :=$ PLUS(j)
$a_{ii} := a_{ii} + 1$
$a_{kk} := a_{kk} + 1$
$a_{ik} := a_{ik} - 1$
$a_{ki} := a_{ki} - 1$
Enddo

Similar use of outer products yields an equally simple algorithm for evaluating the vector $b = C^td$:

$b_i := 0 \quad (i = 1,2,\cdots,n)$
For $j = 1,2,\cdots,m$ do :
$i :=$ MINUS(j)
$k :=$ PLUS(j)
$b_i := b_i - d_j$
$b_k := b_k + d_j$
Enddo

The first algorithm points to useful properties of the matrix A. The diagonal elements are nonnegative and the off-diagonal elements are nonpositive. The element a_{ii} is the total number of arcs to or from node i whereas $-a_{ik}, i \neq k$, is the number of arcs between nodes i and k.

These properties of A are combined with a labelling procedure from graph theory (Carré 1979) in order to get an efficient algorithm for identifying the connected components of the graph. Let node(i) denote the ith node of the graph. The output of the algorithm is a "characteristic" vector c = [c(1), c(2), ..., c(n)] in which c(i) has a value j in {1,2,...,q} indicating that node(i) is in the jth component. The algorithm uses a list T and a list L_i for each node(i).

1. Initialize $c(k) := 0 \quad (k = 1,2,\cdots,n)$.
2. Start loop for $k = 1,2,\cdots,n$.
3. If $c(k) \neq 0$ go to 5.
4. Find all nodes of the component containing node(k):
   (a) Set $q := q+1$, $c(k) := q$, and enter k in the list T.
   (b) Let i be any index in T, delete i from T, find the list L_i of neighbors of node(i).
   (c) If node(i) has no neighbor go to (f).
   (d) Pick a neighbor node(j) of node(i) and delete j from L_i.
   (e) If $c(j) = 0$ then put j in T and set $c(j) := q$. Go back to (c).
   (f) If T is not empty then return to (b).
5. End of the k-loop started at 2.

In (b), the list L_i of neighbors of node(i) is obtained by using the property of A that node(j) is a neighbor of node(i) if $a_{ij} \neq 0$.

At the end of the algorithm the n nodes of the graph are partitioned into q components of $n_1, n_2, \cdots, n_q$ nodes. Observe that $n_1 + n_2 + \cdots + n_q = n$ and that a component with $n_k = 1$ consists of an isolated node which has not been compared with any other node. Items in separate components are also nonrankable. The strategy for computing the general solution of Ax=b is to solve the least squares problem corresponding to each connected component, simply assigning an arbitrary value to the isolated node when $n_k = 1$ and using the theorem when $n_k \geq 2$, and then to embed the resulting q solutions in n-space.

4 Example

Suppose that the six elements of a set $\{h_1, h_2, h_3, h_4, h_5, h_6\}$ have been pairwise compared as indicated by the following eight directed arcs:

1. h_1 to h_2
2. h_2 to h_1
3. h_2 to h_1
4. h_5 to h_3
5. h_6 to h_3
6. h_3 to h_6

7. h_5 to h_6

8. h_5 to h_6

Assume also that element h_j has true intrinsic worth w_j. The 8×6 system $Cx = d$ is represented by the three vectors MINUS, PLUS, d shown in the table below:

i	MINUS (i)	PLUS (i)	d_i
1	1	2	$w_2 - w_1 + \epsilon_1$
2	2	1	$w_1 - w_2 + \epsilon_2$
3	2	1	$w_1 - w_2 + \epsilon_3$
4	5	3	$w_3 - w_5 + \epsilon_4$
5	6	3	$w_3 - w_6 + \epsilon_5$
6	3	6	$w_6 - w_3 + \epsilon_6$
7	5	6	$w_6 - w_5 + \epsilon_7$
8	5	6	$w_6 - w_5 + \epsilon_8$

The element d_i is the observed difference in worth corresponding to the i-th comparison, namely, the difference in worth between the two elements being compared perturbed by some noise ϵ_i. For purposes of illustration, assume that the observations are in fact exact so that each $\epsilon_i = 0$. The resulting system of normal equations is then given by

$$Ax = \begin{bmatrix} 3 & -3 & 0 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -2 \\ 0 & 0 & -2 & 0 & -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$= \begin{bmatrix} -d_1 + d_2 + d_3 \\ d_1 - d_2 - d_3 \\ d_4 + d_5 - d_6 \\ 0 \\ -d_4 - d_7 - d_8 \\ -d_5 + d_6 + d_7 + d_8 \end{bmatrix}$$

Use of the matrix A in the labelling algorithm described earlier indicates that the graph has three connected components

$$\{h_1, h_2\}, \ \{h_3, h_5, h_6\}, \ \{h_4\}.$$

Since the element h_4 is an isolated node that has not been compared with any other element, its worth cannot be estimated. The worths of the elements of the other two connected components can be estimated as solutions of a 2×2 and a 3×3 linear system of rank 1 and 2 respectively:

$$\begin{array}{c} \\ h_1 \\ h_2 \end{array} \begin{array}{c} \begin{array}{cc} h_1 & h_2 \end{array} \\ \begin{bmatrix} 3 & -3 \\ -3 & 3 \end{bmatrix} \end{array} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} =$$

$$\begin{bmatrix} -d_1 + d_2 + d_3 \\ d_1 - d_2 - d_3 \end{bmatrix}$$

$$\begin{array}{c} \\ h_3 \\ h_5 \\ h_6 \end{array} \begin{array}{c} \begin{array}{ccc} h_3 & h_5 & h_6 \end{array} \\ \begin{bmatrix} 3 & -1 & -2 \\ -1 & 3 & -2 \\ -2 & -2 & 4 \end{bmatrix} \end{array} \begin{bmatrix} x_3 \\ x_5 \\ x_6 \end{bmatrix} =$$

$$\begin{bmatrix} d_4 + d_5 - d_6 \\ -d_4 - d_7 - d_8 \\ -d_5 + d_6 + d_7 + d_8 \end{bmatrix}$$

We obtain full-rank linear systems by replacing any one equation in each system by an appropriate auxilliary equation. For example, we could replace the first equation in the first system by

$$x_1 = \widehat{w}_1$$

and the last equation in the second system by

$$x_6 = \widehat{w}_6.$$

The values $\widehat{w}_1$ and $\widehat{w}_6$ are meant to be "reference" estimates of the true worths w_1 and w_6. Use of the theorem then yields the solutions

$$\begin{aligned} x_1 &= \widehat{w}_1 + s_1 \\ x_2 &= w_2 - w_1 + \widehat{w}_1 + s_1 \end{aligned}$$

and

$$\begin{aligned} x_3 &= w_3 - w_6 + \widehat{w}_6 + s_2 \\ x_5 &= w_5 - w_6 + \widehat{w}_6 + s_2 \\ x_6 &= \widehat{w}_6 + s_2 \end{aligned}$$

where s_1 and s_2 are arbitrary scalars. Observe that if $s_1 = s_2 = 0$ and $\widehat{w}_1 = w_1$ and $\widehat{w}_6 = w_6$, that is, the reference estimates are exact, then the above solutions yield the exact worths for x_1, x_2, x_3, x_5, x_6. (Recall that we are assuming zero noise.) Otherwise, for arbitrary values of s_1, s_2 and $\widehat{w}_1$, $\widehat{w}_6$ we get exact *differences* of worths for the computed values, namely, $x_i - x_j = w_i - w_j$ whenever $i, j \in \{1, 2\}$ or $i, j \in \{3, 5, 6\}$.

5 Conclusion

Putting the pieces together, the general solution to the estimation problem represented by system (1) is determined as follows:

1. The nxn matrix A and the n-vector b are computed using the simple algorithms described in the previous section.

2. The matrix A is used in the labelling algorithm to partition the n nodes of the graph into q components with $n_1, n_2, \cdots, n_q$ items respectively.

3. For each component with $n_k \geq 2$, form an $n_k x n_k$ system $A_k x = b_k$ of rank $n_k - 1$ by straightforward selection of proper elements of A and b.

4. In each such system $A_k x = b_k$, one equation is replaced by an appropriate auxilliary equation in order to get a corresponding nonsingular system $\widehat{A}_k x = \hat{b}_k$.

5. Each nonsingular system is solved using a conventional algorithm such as $P\widehat{A}_k = LU$ factorization followed by front and back substitutions.

The theorem is applied on each nonsingular system, the q solutions corresponding to the q components of the graph are then embedded in n-space, in order to get the general solution to the original system of normal equations.

References

[1] E. Batschelet, *Mathematics for Life Scientists*, Springer-Verlag, New York, 1973.

[2] R.A. Bradley and M.E. Terry, The rank analysis of incomplete block designs: The method of paired comparisons, *Biometrika*, 39, 1952, pp. 324-345.

[3] B. Carré, *Graphs and Networks*, Clarendon Press, Oxford,1979, Section 2.4.

[4] W.D. Cook and M. Kress, *Ordinal Information & Preference Structures*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[5] H.A. David, *The Method of Paired Comparisons*, Charles Griffin Publications, London, 1963.

[6] G.T. Fechner, *Elemente der Psychophysik*, Britkopf and Härtel, Leipzig, 1860.

[7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1989.

[8] M.G. Kendall, Further contibutions to the theory of paired comparisons, *Biometrics*, 11, 1955, pp. 43-62.

[9] R.D. Luce, *Individual Choice Behaviour: A Theoretical Analysis*, Wiley, New York, 1959.

[10] F. Mosteller, Remarks on the method of paired comparisons (three parts), *Psychometrika*, 16, 1951, pp. 3-9, 203-206, 207-218.

[11] T.L. Saaty, A scaling method for priorities in hierarchical structures, *Journal of Mathematical Psychology*, 15, 1977, pp. 234-281.

[12] T.L. Saaty and L.G. Vargas, *Prediction, Projection, and Forecasting: Applications of the Analytic Hierarchy Process*, Kluwer Academic Publishers, Boston, 1991.

Development of Human Self in Novel Environment

A. Dreimanis
Environmental State Inspectorate
Riga, Latvia

Keywords: environment, adaptation, self-organization, variety, creativity, chaos

Edited by: Anton P. Železnikar
Received: April 5, 1997 **Revised:** May 12, 1997 **Accepted:** June 7, 1997

A synergetic analysis of the human Self - environment interaction is proposed from the viewpoint of the necessary development of human capacity of his harmonious involvement and viability in novel environment. The concept environment, being treated in line of the Poppers and Eccles thesis of humans three worlds would include a multitude of physical, ecological, mental, socio-cultural, economical, technological and other factors. According to the Ashby principle of requisite variety, the necessary condition of humans successful survival and viability in the more and more complex, novel and varying environment will be predominance of human's inherent variety over environmental variety.
The primary source of growth of human's inherent variety - information and its organized form - knowledge. Mutual interrelations between information, adaptation and self- organization are specified. The importance of creativity as information generation source in evolving systems is emphasized. The roles of self-creativity and flexibility are considered in the context of interrelations of active and passive adaptation. A possible synergetical mechanism of creativity is deduced, and the probable routes of development of human's creative potential and creative approach to the problem solving are pointed out. The beneficial effect of moderate chaos in adaptation and developmental routes is emphasized.

1 Introduction

During the last decades we have observed so striking evolution of our world towards its increasing complexity that a whole set of principally novel complex problems - ecological, technological, economical, social, psychological - have been emerged. Moreover, most of them (e.g., transformations in the global ecosystem, scientific-technological revolution, aggravation of socio-economical problems, especially in the East European countries) are inherently connected with such common fundamental concepts as adaptation, self-organization (SO), harmonization and optimization, in view of an urgent necessity of purposeful development of human personality and his capabilities for successful survival in a novel set of complex conditions.

For seeking of basic principles of beneficial development of human Self to such novel conditions we use following approaches: 1) concepts of a generalized non-linear science - synergetics, in particularly: of SO processes of formation of qualitatively novel states, as well as of chaotic phenomena, being the basis of adaptation, information processes and of development, and 2) W. R. Ashby [2] principle of requisite variety, intrinsically connecting such key concepts as complexity, environment and adaptation.

2 The human three worlds and his generalized environment

When considering interaction of an developing system with changing environment, the term "environment" is to be treated as an open non-equilibrium creation including natural environment, social world as well as artificial environment (i.e., a set of objects, conditions and requirements having been emerged as the result of human's and

societys activities). In particularly, for the purposes of analysis of a "human-environment" interaction, the term "environment" can be understood in line of C. Popper's and J. Eccle's [11] concept of human's three worlds - the set of 1) physical and 2) mental objects and states, as well as that of 3) mental products. Therefore, the concept environment would include a set of various physical, mental, socio-economical and psychological factors and conditions.

In order to reveal basic features of human-environment interaction as well as the possible routes of optimization and harmonization of such interactions, especially in conditions of crucially changing environment, let us refer to the aforementioned principle of requisite variety which states that for successful development and survival of the given system (e.g., a human), its own inherent variety should exceed the variety of its environment. Moreover, as a voluntary decrease of environmental variety (or diversity) could lead to destabilization effects, then the basic problem of a human's adequate development in conditions of altered environment could be defined in the following way: how should a human organize his activities and develop his own worlds, in order to enhance his own variety (or his self-variety) and, therefore, to provide his favourable adaptation and survival in his changing world and in a novel complex environment.

3 Self-variety factors: knowledge, self-organization, adaptation

Let consider possible ways of enhancing of human's inherent self-variety and specify essential factors which determine the growth of self-variety. To do this, let develop an unified approach to SO, adaptation and information and specify possible interrelations between these factors.

One of the basic routes for systems to evolve themselves would be acquisition of such complexity factor as knowledge being a form of organized information and envisaged to better comprehend the system's functioning in complex environment [8]. Moreover, it is reasonable to consider that just information, knowledge and meaning will be the ways we relate to our environment by means of SO processes [6]. In the context of acquisition of adequate knowledge about the world for aims of successful adaptation in novel environment, among various kinds of such knowledge [7] it seems purposeful to distinguish following components: a) knowledge about our own abilities (or limitations thereof), and b) possible human interactions.

In order to specify further relations between the self-variety growth factors, it should be emphasized that both - adaptation and information - are ultimately based on such concepts as chaos and diversity, taking into account that possibility of adaptation occurence as well as of information generation is basically provided by chaotic and strange attractors: just such attractors, due to random changes in systems s conditions, are most capable to create sufficient flexibility and to provide generation and processing of novel information [10].

Furthermore, the system's adaptability will emerge from the systems capacity to anticipate environmental perturbations [12]. At the same time, one of the basic adaptability components would be behavioural uncertainty component of adaptability [4]. It is just such behavioural uncertainty which allow the system (e.g., a human) to cope with the environment unpredictability and to develop non- standard approaches in the process of choice and achieving his own aims in a novel environment.

On the other hand, adaptability is the use of information to handle environmental uncertainty [4] and is brought in, in order to replenish deficiency in the necessary information. The existing marked uncertainty, corresponding to a lack in necessary information in conditions of changing environment where there are no certain guidelines for actions, is put in the forefront of generation of adequate reliable information which is urgently necessary for harmonization of human Self in a changing environment.

Further, according to the thesis about a self-organizing system as a system consisting of an organism and its environment [2] as well as a similar viewpoint about a system with self-learning ability as reacting with an environment at two different levels [13], one should distinguish a "pure" or passive adaptation (i.e., a system is mainly evolving in its environment) and active adapta-

tion where the system evolves its environment.

The dual nature of human adaptation is due also to following point: environmental novelty could be formed by two distinguishing phenomena: a) the processes in a local area of human Self, and b) the second-level adaptation of the local environment to more global environment. On this level, due to duality of local environment (being the adaptation target as well as adaptation subject), it is reasonable to expect synergic coadaptation of the human Self and of his environment to global environment and its requirements to which is faced up a developing human Self.

4 Creativity - its synergetical mechanism and possible stimulation ways

For the problem of harmonious involvement of a human into a changing environment seems to be essential that between "traditional stabilizing adaptation and active self-creativity various tensions exist " [13] which have harmonizing functions and are manifested most strongly at the border between passive and active adaptation. Just this border region is characterized by optimal flexibility which is necessary for beneficial multilevel coadaptation of humans and of their environment. In particularly, a human should also by himself create such environment, in order to provide most favourable conditions for development of his own Self.

Thus, we bring in consideration the creativity problem and specify mutual interrelations between active self-creativity, information and adaptation. In conditions of crucially changing environment where there are numerous degrees of freedom for the system to evolve, just the necessity of flexible thinking and of generation and/or acquisition of novel information comes in the forefront, which will provide creative reaching out of the system's own boundaries [3], where "under self-creativity and adaptability self-organizing system will reach the upper limit of ordinary environment at the end of evolution by gradual and sudden changes and finally break the limit of ordinary environment" [13].

Thus, we consider creativity as the information generation source for a developing system, in order to design its evolving activities towards its adaptation to a novel environment, which proceeds mostly due to SO processes, often being initiated by small internal as well as environmental fluctuations or "stressors" of different nature. To reveal a possible synergetical mechanism of creativity, let us note: according to neurobiology and non-linear physics data, SO is a fundamental principle of structure and functioning of human brain. Neural networks (NNs) of the brain possess the main features for a generalized net to be a self-organizing system high level of complexity, functional redundancy, distributiveness, hierarchical organization and multifunctionality. Besides, NNs themselves are capable of SO and of various modes of information processing, (e.g., formation of and associations) [1].

An elementary creative act we refer to generation of dissipative structures (DS): various fluctuations, inherent to macrosystems (also to NNs), would, due to their amplification capability, cause a rise of DS in the brain [5]. Such structures we treat as qualitatively novel states of neural interconnection and activity patterns. Subsequent fixation of spontaneously arisen novel combination of associations signifies the generation act of novel information. Such qualitative jump in generation of novelty one can refer to the jump into the region which is hardly accessible from the classical logic's approach. Thus, due to self-creativity, self-organizing systems in this case break through the tie of the old structure and stay outside the old set of stases(a set of steady states) [13] so, breakdown of classical logic in creative act embodies Godel's incompleteness in evolution of self-organizing system.

Efficient use of available elements of information and knowledge for beneficial development of human Self in a novel environment would require a proper creative potential and "spirit" or readiness, especially in case of considerable environmental uncertainty. So, in conditions of a multitude of environmental agents there rises the necessity of tuning of a human to the dominance state which is characterized by synergetical slaving of neural activities towards problem solving and, therefore, providing a more reliable route of evolutionary SO. A possible mechanism of synergistic strengthening of creative actions over the competing ones could be internal motivation. A

possible route to develop creative potential would be training (at first, for children) of associative right-hemisphere thinking, by means of various arts activities.

5 The role of chaos in adaptation and development of human Self

Besides the aforementioned essential role of chaos and uncertainty in adaptation and creativity processes, one should emphasize such fundamental tendency of most of complex systems - to behave between chaos and order [9]. Therefore, just on such border region between traditional states and the novel still uncertain factors, will most likely emerge the necessary for a human high flexibility and efficient adaptability and, in general, sufficient level of complexity and self-variety of a developing system. Also, in creativity there should exist some optimal combination of order and chaos elements, in particularly, in the state of moderate chaos near the order-chaos border region. Such a state would assist to destroy rigid thinking schemes and, thereby, provide favourable conditions for a rise of broad spectrum of various versions of the problem solution, thus favouring to finding most optimal routes.

Regarding to human's adaptation to increasing complexity of social interrelations, the problem arises: how to solve and minimize possible controversies and conflicts being natural by-products of development processes. It is just an intermittent chaos in interpersonal relations (Nicolis, 1986) which would prevent such conflicts, by ensuring that one's person's internal SO would not destroy the other person's SO process. It is purposeful to recommend to develop concerted, mutually contributing SO processes, as "self-organizing one another means to create and maintain conditions allowing the richness of human relations with a view to increasing the possibilities for all" [14]. Thus, moderate chaos in the sense of flexibility of interpersonal relations would ensure proper self-variety.

6 Conclusion

Using inter-disciplinary approach and concepts of modern non-linear science, we have studied features of human environment interaction (where "environment" is treated in its extended sense as a set of physical, mental, social and other factors) and revealed typical elements of human adaptation and development of his Self in conditions in crucially changing environment. The favourable effect of chaos-order border mode on efficient developmental and evolution routes, in particularly, on passive and active adaptation and their optimal balance, flexibility, information generation and creativity is deduced. A key point of human favourable adaptation to novel complex environment would be growth of human's self-variety by developing his self-creativity, SO and his knowledge about his novel environment.

References

[1] Amari, S. (1988). Dynamical stability of formation of cortical maps. In M. Arbib and S. Amari, editors, Dynamic Interactions in Neural Networks: Models and Data, 15-34, Springer, Berlin.

[2] Ashby. W.R. (1959). The Introduction to Cybernetics (in Russian). Inostrannaya Literatura, Moscow.

[3] Banathy, B. (1993). From evolutionary creativity to guided evolution, World Futures, 36, 73-79.

[4] Conrad, M. (1983). Adaptability. Plenum Press, New York.

[5] Dreimanis, A. (1994). Creativity of natural and artificial brain: towards an unified synergetical approach. In F.G. Bobel and T. Wagner, editors, Proceedings of the 1-st International Conf-ce on Applied Synergetics and Synergetical Engineering, 161-166, FhG IIS, Erlangen.

[6] Keel-Sleswik, R. (1992). Artifacts in software design. In C. Floyd et. al., editors, Software Development and Reality Construction, 168-188, Springer, Berlin.

[7] Kobsa, A. (1984). Knowledge representation: a survey of its semantics, a sketch of its semantics. Cybernetics and Systems, 15, 41-89.

[8] Kuhn, H. and Lehman, U. (1984). Transition from the non-living state into the living state. In R. K. Mishra, editor, The Living State, 300-317, Delhi.

[9] Langton, C. G. (1990). Computation at the edge of chaos. Physica, 42 D, 12-37.

[10] Nicolis, J. (1986). Chaotic dynamics applied to information processing. Rep. Progr. Phys.,49, 1109-1196.

[11] Popper, C. and. Eccles, J. (1977). The Self and Its Brain. Springer International, Berlin.

[12] Salthe, S. (1992). Hierarchical self-organization as the new post-cybernetic perspective. In G. van de Vijer, editor, New Perspectives on Cybernetics: Self-Organization, Autonomy and Connectionism, 49-58, Kluwer, Dordrecht.

[13] Tao, H. (1993). The Structure of Multistasis: on the evolution of self-organizing systems. World Futures, 37, 1-28.

[14] van Foerster, H. and Floyd, C. (1992). Self-Organization and software development. In C. Floyd et. al., editors, Software Development and Reality Construction, 75-85, Springer, Berlin.

Overview paper

Information conundrum: semantics ... with a payoff!

Jacek Kryt
Retired from Ryerson Polytechnic University. Toronto, Canada.
res.: address: 21 Heathcote Avenue, Willowdale ON, M2L 1Y6, Canada
phone 416-445-4547; E-mail jkryt@acs.ryerson.ca

Keywords: computer science, curriculum, data, end user, information, information highway, information systems, requirements, systems failures

Edited by: Marcin Paprzycki
Received: January 16, 1996 **Revised:** May 4, 1996 **Accepted:** May 26, 1996

The sorry state of business information systems is no longer a secret. Since 1968 we have had a computer science curriculum which is supposed to provide us with properly trained graduates to handle the problems related to this new technology. But there is criticism of those graduates voiced from various quarters. They have been very helpful developing this new technology, but they have failed to satisfy the needs of those who in search for information expected to benefit from the flow of data facilitated by computers. This prompts the question: what is this information anyway? Computer scientists assert that information is just data processed by computers. I will show that this view is unacceptable as it causes considerable harm to information systems now, and is likely to have a negative impact on the information society. We need a clear understanding of the term "information", the most fundamental term for the Information Age.

1 Semantic Trap for Humans (Kryt 1996)

The sorry state of information systems or "the artifacts (the combination of technology, data and people) that produce the information resource for use by individuals, organizations and society" (AIS 1995) is a fact. Our society is fascinated with technology, fooled by the deification of computers and the hype fed to people by numerous self-serving powerful lobbies, attempting to silence our concerns.

Decades which have been spent on academic quantitative research on the need for end users to be involved in information systems development activities seem like wasted time. Information systems specialists graduating from computer science programs are disappointed and confused. They know computer programming and use the best technology, yet, when they build information systems, they often fail to cause the expected information in targeted humans and achieve their satisfaction with the systems.

We spend mind-boggling amounts of money on information processing. Lately, looking for reasons why productivity growth slowed down considerably in the mid 1970s despite two decades of exploding investment in computers, some economists have suggested that benefits promised from investments in this technology have not materialized. (Little, 1997)

Unaware of ambiguity of terms used, we fall into a treacherous semantic trap which results in havoc and wasted resources in information systems applications, academic curricula, research, and information highway discussions. Besides the term "information," computer scientists, when they speak about "cognition", "perception", "learning", "intuition" and "thinking", give the terms a meaning very different from those that we attach to human functions bearing these names.

To end this confusion, our information systems function specialists who conceive, design, develop and implement the organization's information and communication systems (AIS 1995) have

to accept, as most of us do, that information for humans is data of interest to the receiver that affect recallable memory. Everything else outside the mind of the receiver is data, even that which the potential informer considers to be valued information. The current mindless alternating of these terms in communications has to stop. We all have to say so forcefully, as "[s]emantic and methodological discussions are to the ears of some scientists what symphonic and chamber music is to the tone-deaf. They do not hear anything that makes sense or seems to matter." (Machlup 1983)

The phenomenology of human cognition is the cornerstone on which I will build my interpretation of the term "information." The brain is not simply a meat machine, as Minsky says. (Weizenbaum 1995, p. 259). Humans, unlike computers, have bodies, instincts, feelings, life experience, intuition, common sense, common knowledge, values, personal knowledge, wisdom, and beliefs which guide their interests and choices. Because of our limited speed and ability to select and process data coming at extreme rates from computers, we allow only a small proportion to reach our cognitive structures. We ignore, as it were, the rest to avoid data overload and mental chaos.

In this paper I will begin with what is hurting us: the failures of many information systems. In turn, I will describe the saga of computer science curriculum. Some have begun to question its relevance. Comments come from various quarters and the critics are getting organized. I will examine the history and use of the terms "data" and "information" to understand the unnoticed persistent confusion in the use of these terms. Finally the study of human cognition and the analysis of what we experience on the Internet will enable me to determine the underlying cause of the information conundrum and redefine the term "information."

2 Why Do Information Systems Development Projects Fail?

Failures of computerized information systems in the private sector are usually denied or kept secret. One can hardly imagine conducting a successful survey asking IS managers to report on their failures! You hear these stories occasionally in confidential, hushed conversation, at the conclusion of cocktail parties. In the public sector, however, they are discovered routinely by governmental auditors. This was the case with the Canadian air-traffic control system. The failure of an Automated Trading Floor at the Toronto Stock Exchange to materialize as announced, was repeatedly noted in the press. It was the common knowledge that the Denver airport established new record numbers of travelers tracing their misdirected suitcases. The costs of these projects run into tens or hundreds of millions of dollars.

"Computerized systems do not do, what they are supposed to. The failure of the information technology industry to be able to build reliable systems within a specified time frame, for a set price, is no secret anymore." (Rowan 1995)

It is not always clear what constitutes the failure of a system. Is it the project not being finished on time or within budget, or is it the project not being finished at all, or finished but not implemented, or if implemented, sabotaged, not used, or used only for lack of anything better? The figures quoted are very unreliable as one can hardly imagine conducting a successful survey asking information systems function managers to report honestly on their failures!

The figures arrived at by the US General Accounting Office are a biased sample. They cover systems that were investigated because the office considered that they were on the point of failing (Glass 1994). The reality, however, remains that systems fail, that the information systems industry is doing soul-searching, and that "horror stories of long delays and serious foul-ups are forcing information technology consultants to develop tools to rebuild client confidence." In their own words, they admit that "if you are looking for a major front-end fault in the whole process, the projects are often technology-driven and not business-driven." (Rowan 1995).

3 The Saga of Computer Science

3.1 End Users Become First Programmers

A computer science curriculum was published in the USA in 1968 by the Association for Computing Machinery in order to prepare graduates for work in this new domain.

"Basically computer science programs are focused on computer programming, logic, and mathematical problem solving. Reference is often made to information processing, but the term data processing is often more correct. Criteria such as user needs analysis and a global view of the definition of the problem that an application is created to solve may be of secondary importance." (Debons et al. 1988, p 33)

Computer manufacturers loved graduates who were "computer- literate." We have to give them due credit for their contribution to progress in technology and systems software.

They were not, however, "people-literate." or "business-literate." With more powerful machines and languages, business applications such as inventory control and sales forecasting were more complex, needing more of human decision making expertise. Integration of systems was attempted. The elementary lack of familiarity of computer science graduates with business began to show in voluminous and useless management information systems reports and failures of our systems.

At this time some computer science departments organized cooperative programs (called sandwich programs in the UK). Maybe it was not intended this way, but in reality computer science students alternating periods of study and employment, through osmosis were getting some education and training in business knowledge and procedures, at the expense of the industry that hired them. Nevertheless, many of them would return to what they liked best - dealing with obedient machines away from end users.

3.2 Computer Science has a Powerful Lobby

Computer science has a powerful self-serving lobby. We all know how nice it is to teach programming in any of a large number of languages. There is a valuable windfall to boot:

"[T]hose who cultivate competence in the use of new technology become an elite group that are granted undeserved authority and prestige by those who have no such competence. [They] accumulate power and inevitably form a kind of conspiracy against those who have no access to the specialized knowledge made available by the technology." (Postman 1992, p. 9).

Academics have an enormous appetite for cutting-edge technology. Universities receive generous donations of hardware and software. Everybody enjoys this clever marketing strategy of vendors. Roszak (1986, p. xi) wrote that "Gifted minds in the field of computer science have joined the cult [of information] for reasons of power and profit."

In a similar vein, Dreyfus and Dreyfus (1995 p. 428) complain that the supporters of strong artificial intelligence (AI) "had undisputed control of resources, graduate programs, journals, symposia, etc., that constitute a flourishing research program." Herbert Simon, the champion of strong AI, in supporting his philosophical position, unwittingly confirmed this criticism himself, stressing the rate of new publications on AI as a proof of the scientific success of the research. That was the way he tried to refute the statement that meaningful progress in AI had slowed down, if it had not altogether stopped (Simon 1995, p. 245).

There were numerous cases of individual research funding. At least one researcher publicly complained that because of a non-disclosure clause, he ran "into the problem of reporting results" of his study involving introduction of microcomputers into the schools. A colleague of his spoke bitterly about the takeover of computer education by commercial forces, out of the hands of educators (McLean 1985). It is common knowledge in North America that educational software, until now, has been scarce and of poor quality. Yet the bulk of funding always goes for hardware!

Big expenses for advertising budgeted by computer manufacturers helped to make the newspapers very technology-friendly and eager to print articles full of hype. Scare-mongering was common. The public was constantly admonished: "Your children will be eating in soup kitchens if they do not have a computer now." Programming, misnamed "problem solving," would supposedly

spare them this misery. Nowadays, the similar threats are addressed to women, the current target group.

For years employment ads in newspapers and journals prepared by human resources departments in organizations required a computer science degree for any job related to computers. Rare were the exceptions to this practice, made only at the request of smart information systems managers. This produced a computerized perpetuum mobile - an infinite loop.

A. Students were told to enroll in computer science programs.

B. Human resources departments, not knowing any better but playing it safe, required a computer science degree for any job related to computers.

C. Quantitative surveys, in turn, were all this time "discovering", that computer science is the most frequently required qualification for any job related to computers.

D. GO TO A.

By 1967 some Schools of Business began to train systems analysts, knowledgeable in business administration, programming and information systems development functions. Alas, these schools did not stand a chance in face of the established computer science stronghold.

4 Computer Science Curriculum Under Fire

4.1 Misgivings on the Part of Industry

Companies "hiring from universities complain that graduates don't have the right skills to do the jobs they are hired for ...[and] that students aren't being taught the right skills needed for the real world." Such were the findings of a leading Canadian university in 1995. In a recent panel, other universities defended their positions with well- known arguments: Preparing graduates who in their long-term professional careers are able to adapt to ever changing technology, universities do not have the time to make graduates immediately productive in the company.

It is symptomatic that in the same debate, a major computer vendor said that her company invests a "huge amount of money in training employees. Generally, [it] is more likely to promote and train internally than to go outside to look for the right person for the job. And new employees go through intensive training to get them up to speed on [the company's] way of doing business." (Wintrob 1995). Eleven years earlier, the chief of corporate IBM research, together with the company psychologist, had objected that

"There is rarely any explicit requirement for any courses on human factors or basic psychology. It is quite possible for a person to earn a Ph.D. in computer science, anticipating a lifetime career building tools for human beings to use, by spending several years learning how computers work, and yet no time learning how human beings work" (Branscomb, Thomas 1984).

This produced concerns about usability and the human-computer interface focused on technology such as use of color, icons, the mouse, the keyboard, an ergonomic profile of the desk, (cf. Norman 1990) but not on data provided to humans. "[E]fective information management must begin by thinking how people use information - not with how people use machines." (Davenport 1994, p.120)

4.2 Criticism From Computer Scientists

Computer science graduates should broaden their research agenda "to the increasing number of business, commercial, scientific, and engineering problems that have (or ought to have) a significant computing component." Such was the conclusion of a 1992 report entitled "Computing the Future" It was prepared by a high-powered USA committee. The authors in one of the recommendations required "Ph.D. students either to take a graduate minor in a non-Computer Science & Engineering field or enter the Ph.D. program with an undergraduate degree in some other science or engineering or mathematical field." (Hartmanis, Lin 1992, p. 8)

For students planning a career in business information systems an undergraduate degree in business administration would be my logical choice. This idea is also the essence of articles in Informatyka (1994) and of recommendations of the Council of European Professional Informatics Societies (CEPIS 1995).

4.3 Questions From Information Systems Academics

In October 1995, the following message was posted on the ISWorld discussion list:

"Is there a difference between technical programming skills and generalized systems analysts' competence? It has been my experience that companies want systems analysts who understand organization systems and programmers who understand computers. It has also been my experience that the skills needed for creating new technologies and the skills needed to integrate new technologies into existing systems are different and usually mutually exclusive. It is a rare person that can do both. The traditional skills of a systems analyst (being able to look at an organization and analyze how it works) are still in great demand..." (Binning 1995)

In my concise response, to reinforce this message to all members of the list, I deliberately used recent stereotypes:

"We know that what you see depends on where you sit. This is why an end-user (business-oriented) views the target system differently than a technical expert (computer science-oriented and -trained). The corollary is that in computerized systems development it would be wrong to say to the user, "What you see is what you get" unless the technical expert knows something substantial about the context of the user's work and business environment. Then he can discuss intelligently and effectively users' requirements, without a need to make arbitrary assumptions (often wrong) in the design and programming stages. A good combination in my experience is the program [of studies] combining 75% of business management courses with 25% of SAD [Systems Analysis and Design courses] and programming." (Kryt 1995).

Many messages on the Internet in the last two years concern the contents of information systems academic curriculum. Here is an example of strongly worded advice:

"It seems as if some members of the IS community still haven't internalized the oft repeated messages that user requirements drive the process; that errors in the requirements phase take 1000 times as long to fix as programming errors and 100 times as long as detailed design errors and 10 times as long as logical design errors; and that quality is heavily reliant on the very first phase [of information systems development]." (Clarke 1996)

Other letters have questioned the need for tenured faculty to teach programming or software package courses, which have masqueraded as information systems courses. Some messages report that in response to the objections by non-information-systems faculty a few universities have simply removed information systems courses from programs.

4.4 Harmful Technical Intermediaries

Instant and continuous feedback from receivers of our messages is essential in human communications to maintain their interest and the high proportion of data that will evoke information in their minds. But equally important is our ability to direct our correct message to the right receiver. Person A, the speaker, the writer of a letter, or the author of a book, pays a great deal of attention to that to ensure that person Z will receive what he sent.

This process becomes very complicated in the case of computerized information systems because of the need for computer-science trained systems analysts and programmers, who, however, typically are not knowledgeable in the subject matter to be computerized.

1).User A explains to analyst B, who later, with what he understood, instructs programmer C how and what should be communicated to user Z. Programmer C adds his own interpretations and assumptions. The first distortions of the message appear:

Direction of flow of instructions: ======>>

User A ===>> Analyst B --^~_/~.>> Programmer C -/_|\--/\>> User Z

2).User Z, in turn, explains to analyst B, who later instructs programmer C, what output he hopes to receive from the system that they are designing for him and what input he thinks user A should feed to the system. Direction of flow of instructions: <<======

User A <<---/~~^ Programmer C <<^~_/^~\.~ Analyst B <<=== User Z

3).Information systems experts B, and C try to understand and reconcile the explanations of both A and Z, whose functional domain is to them mostly unfamiliar and of little interest. They try to grasp how all this should be done. They design

the system. Where they are not quite sure what A or Z wanted them to do, instead of asking for additional clarification, they make assumptions - frequently wrong ones. They tend to include their own "elegant" technical solutions. Users A and Z occasionally try to check what's going on in the information systems development. Because of their own technical ignorance and the technical jargon used by experts, this is usually inconclusive, as they are unable to talk intelligibly to each other. At the completion of the project, instead of the expected flawless communications:

User A <<==========>> User Z

they get distorted messages, like in the game of "broken telephone."

User A <<'''|_/^^^^^--^_--^^^===/\/\--###\ ^^^^^^>> User Z

There is a difference: in the game the distortions in communication produce hilarity; in information systems they produce huge financial losses. The system is a failure.

These considerations about how people communicate, what data people select, and how they "construct" information are extremely important for all who deal with these concepts, from information systems analysts developing a system, to academics designing a curriculum, to agencies involved in the debate about the information highway, to... futurists probing the unknown.

The controversy about various aspects of computerization of society has been present from the times of Norbert Wiener (1948). In a 1991 book we find 746 pages of articles, most of which raise searching questions about the new technology and its applications in society (Dunlop & Kling 1991). These are strong winds of change, and more have come since.

Thus far, I have examined the symptoms of our information systems development malaise and some remedies suggested both by industry and by academia. Now I will deal with the meaning of this key term "information", which has preoccupied me for years. First, a little history.

5 From Oral Tradition to the Internet

5.1 Era of Scarcity of Data

From 15.000 to 3.100 B.C., it took humanity at least 12 millennia to pass from oral to written communications. (see Table 1) In oral communication, data were scarce, the subject was of vital interest to the partners, face-to-face situations would allow them a perfect bi-directional feedback and maintenance of high interest and accuracy of the message. Whatever data people were exchanging would become interesting information for both of them. "[H]umans achieved economical and effective distribution of information through existing channels in natural and intuitive ways." (Dervin, Nilan 1986, p. 1) Those were "the good old days when human talked directly to human and through gesture, talk, song, story, drum, and crafted sign and picture conveyed to each another something we call information." (ibid. p. 2)

At that time there was no need for a distinction between data and information.

Written tradition was initiated by counting. Simple notched-stick tallies or pebbles gave way to more complex tokens and incised tablets. It was as late as 3.100 B.C., in Sumer, that we can find abstract numerals, ideographic writing, and phonetic coding (Logan 1995). Writing dominated human communications for the next 4.500 years. There were 30.000 books in Europe by the middle of the 15th century (Ramo 1996). "Fifty years after the press was invented, [i.e., by the year 1500] more than eight million books had been printed" covering a wide variety of subjects "introducing the age of information" (Postman 1992, p.61) that I would more accurately call the age of data.

I submit that the appearance of a citation dating from 1646, which supports the definition of datum as "a thing given or granted; something known or assumed as fact..." (The Oxford... 1989, p. 264) was the direct result of this sudden explosion of data available to the readers.

5.2 Era of Flood of Data

With radio (1920) and television (1945) the air waves became saturated with sound, text, and pictures aimed at everybody and therefore at no-

Approx. date	Medium of communication	Volume of data available to society	Use of terms "Data" and "information"
15.000 B.C.	Oral tradition Discourse	Scarcity of vital data to solve problems	If terms used, no distinction implied
3.100 B.C.	Tally, tablets, phonetic alphabet, manuscripts	By 1456 only 30.000 books in West. Europe	As above Information first quoted in 1330
1456	Gutenberg . Printing press	By 1500 8 million books	No distinction in general use
1646	Printing spreads	Variety of subjects covered	As above In 1646 data first quoted in the modern sense
1800	newspapers appear	Volume of impersonal data grows	No distinction in traditional general use
1840	Telegraph, Photography	Data become commodity	As above
1920	Radio	Rapid saturation with impersonal data. Some data not selected	As above no distinction in traditional use of terms
1945	Television	As above	As above
1990	The Internet	Flood of data. Mismatch of human/electronic speed & volume	As above. Differentiation of data and information is VITAL NOW

Table 1: History and Evolution of the Terms "Data" and "Information."

body in particular. They were directed at potential receivers whose interest was estimated statistically, by polls, or by wishful thinking of marketing managers. Then, within a mere 25 years, the facsimile and the Internet, with email, multimedia and virtual reality, annihilated the global space.

With the increase in use of unnatural channels offered by the new communication technology the equality between data and information gradually vanishes. The more we distance ourselves in communications from face-to-face discourse and continuous, instant feedback, the smaller the proportion of data that evokes or causes information. One can question how much of the information at the receiver's end is caused by the source? (Mingers 1995, p. 289). In discourse this link between the source and the receiver is very strong and much information is caused by the source. It is weaker in the case of print and very weak for mass media, where it can be simply refused by the targeted receiver, who wants to stop meaningless data overload and the mental chaos it produces. On the other hand, these technologies, enabling ease and speed of communications and the existence of nearly continuous feedback or even quasi-physical presence of distant partners, may give us the illusion of a face-to-face discourse. Used judiciously, they may be helpful in improving human communications.

5.3 Our Defenses

This data overload happens at a time when all our traditional defenses against useless or unwelcome data are down. In the past we had social institutions which advised us on how much value we must give to data:

1) Courts of law with their strict rules of evidence told us what to consider as true.

2) School-designed curricula assigned priorities to what was worthwhile to learn.

3) Families, would decide what was suitable for children to know.

4) Political Parties suggested to people what value to assign to political events and commentary.

5) Religion and

6) The State, were important institutions that

controlled unwanted data.
7) The theory of science guided scholars.
8) Marxism used to control the thinking of millions. (Postman 1992, p. 73 - 80).

All these institutions are now under attack, losing their authoritative status, at a time when technology has exponentially increased the amount of available data. There is also a drastic mismatch between computer extreme speed to supply data and the slowness of human mental processing.

Many authors report that we experience "[i]nformation anxiety [that] is produced by the ever-widening gap between what we understand and what we think we should understand... We read without comprehending, see without perceiving, hear without listening" (Wurman 1989, p. 34).

We have to control our boundless greed for more information, to avoid overloading ourselves with data that we are unable to process properly into information. "An excess of information may actually crowd out ideas, leaving the mind distracted by sterile, disconnected facts, lost among shapeless heaps of data." (Roszak 1986, p.88). "Data, data everywhere, and not a thought to think."(Shera 1983, p. 384).

I will turn now to the human cognition: what do we do, when we see or hear something, or more to the point, when we receive, perceive, understand or interpret messages on the screens, examine reports, and interpret signs and graphics. We may not realize it, but we do all this in a special way.

6 Phenomenology of Human Cognition

6.1 This Fickle Selective Vision and ...Information

First let us examine an easier, popularized description of human perception:

"When someone asks, "Do you see what I see?" there's a simple answer: 'No.' What we see is as individual as our personalities, not only because of differences in our eyes but because of what our minds make of the scene. "What happens depends on our experience and the task we have in mind," says Dr. David Williams, an optical researcher at the University of Rochester. "If I am looking at my room, what I see depends on what I'm trying to accomplish. If my goal is to go out the door, I will interpret the scene differently than if I am searching for a book." And while it's common to speak of the mind's eye, there's no master screening room in the brain that turns what our eyes see into a picture. The brain works on many levels simultaneously to give meaning to what we see." (Immen 1995).

The philosopher and expert on human intelligence and artificial reason, Hubert Dreyfus, describes the same process in rigorous scientific terms, for those who prefer them:

"So how does the brain do it? No one knows. But certain facts seem relevant. First, it appears that experience statistically determines individual neural synaptic connections, so that the brain, with its hundreds of thousands of billions of adjustable synapses, can indeed accumulate statistical information on a scale far beyond current or foreseeable computers. Second, the reinforcement-learning procedures now being studied generally produce simple stimulus-response behavior in the sense that the input, a situation description, maps directly forward into the output, an action or situation value. The brain clearly has internal states that we experience as moods, anticipation, and familiarities that are correlated with the current activity of its hidden neurons when the input arrives. These are determined by its recent inputs as well as by the synaptic connection strengths developed on the basis of long-past experiences, and these as well as the input determine the output...[T]here is evidence that the internal brain state interacts with an input and then feeds its output to motor-control neurons as well as back into the input pathways, affecting receptors through motor control so that they actively seek information, and simultaneously influencing perceived relevance through the feedback into input pathways. This would be the brain basis of the phenomenon of global sensitivity that enables a skilled person to see directly what is relevant in his or her skill domain. This feedback, based on the interaction of sensory input and internal brain state, would be powerful mechanism for dealing with information pickup and relevance problems,

but currently no details of this mechanism are understood or even hypothesized in a way that could guide AI research."(Dreyfus 1992).

Checkland says that "'information equals data plus meaning'. That is, by attributing [in our mind] meaning to data, we create information." (Checkland and Scholes 1990, p.303). Enactive cognition, as discussed by Varela (1991, p.9), is based "on the growing conviction that cognition is not the representation of a pregiven world by a pregiven mind but is rather the enactment of the world and a mind on the basis of a history of the variety of actions that a being in the world performs." All the views in this section use slightly different terminology but appear to come from the same cognitive research.

As a metaphor for the process of selective vision, we can consider the way we read our newspaper. Various sections contain what the publisher intends to become information for us. Each section potentially interests somebody. But whatever the publisher thinks about these sections, if after a very superficial scan, I discard one because it does not interest me, I throw away not my information but some data presented to me. I subjectively decide in my judgment what out of this flood of data meets my requirements and deserves a place in my memory. The same applies to books, radio, surfing the Internet and... 500 TV channels of the "Information Highway."

Data presented to their intended receiver in a variety of formats have to meet his subjective selection criteria. Using information systems development terminology, we would say that input and output data have to satisfy the requirements of the system as defined by the end user in cooperation with the information systems expert. Using a metaphor, we can compare input and output data meeting these criteria to a color print that one wants to see. A system's requirements are equivalent to the negative to be prepared by information systems experts, without the help of a modern camera. It looks so different, yet must produce the desired color print. In practice, it often doesn't!

6.2 Information Does not Equal Data

There is some controversy over the definition of data that most of us consider as things that have been given. Some authors define them as facts in isolation, others as records stored in a computer. Debons (1988, p. 8) proposes a standard definition. "Data: letters, numbers, lines, graphs, and symbols, etc., used to represent events and their state, organized according to formal rules and conventions." Buckland, (1991, p. 46) sees it differently: "There is a tendency to use data to denote numerical information and to use text to denote natural language in any medium, but it is wise not to assume any firm distinction among data, document and text." In this paper I want to focus on the semantics of information.

There is a growing number of serious researchers, experts, and top business consultants like Peter Drucker, who voice the view that data and information are not synonymous. They think that data, in order to acquire the status of information, must be processed by the cognitive system of the receiver. I forgot that it helps to find out what other disciplines have to say on subjects of our research, (Boulding 1956) and I quoted a number of them in an earlier article (Kryt 1995).

Dervin and Nilan, Information Science authors, researched the evolution of the meaning of the term information and uses of information. They report that "the review of post-1978 literature reveals [that] a conceptual paradigm shift in this area has been taking place since 1978." In the preceding, traditional 1966-1978 research, coinciding, as I think, with the publication of the Association for Computing Machinery computer science curriculum (1968) and probably the period of its big impact on the literature,

"[I]nformation is seen as objective and users as input-output processors of information...Information has been variously defined as 1) a property of matter, 2) any message, document, or information resource, 3) any publicly available symbolic material, and 4) any data. Information need is defined as a state of needing anything called information. Information need is not defined as what users think they need, but in terms of what the information system possesses."

All of this, I think, was true at a time when discourse was the main medium of human communication in what I would call the era of scarcity of data. The data flowing in both directions in discourse were selected as relevant by the two parties involved and became a part of their information and knowledge.

By 1978, beyond any doubt, we were in the era of flood of data. The post-1978 literature is characterized by constructionist,

"alternative research approaches [that] focus on the user. They examine the information system only as seen by the user. Information is seen as constructed by human beings. Users are viewed as beings who are constantly constructing and are free to create from information systems and situations whatever they choose... Alternate studies have defined information as 1) that which is capable of transforming image structure and 2) any stimulus that alters the cognitive structure of the receiver... Information need has been variously defined as 1) a conceptual incongruity in which a person's cognitive structure is not adequate to a task, 2) when a person recognizes something wrong in their state of knowledge and desires to resolve the anomaly, 3) when the current state of possessed knowledge is less than is needed, 4) when internal sense runs out, and 5) when there is insufficient knowledge to cope with voids, uncertainty or conflict in a knowledge area." (Dervin, Nilan 1986).

In the early 1980s, Machlup published his monumental works about "Knowledge and Knowledge Production" and "The Study of Information." He terminated his detailed analysis of the term "information", as used by 39 authoritative researchers in 56 contributions, with a somewhat obscure and inconsistent statement which placed him between the two approaches:

"Information is addressed to human minds and is received by human minds, though the recipient need not always be chosen by the informant or transmitter. Pieces of information carry meanings and are interpreted by cognitive processes, but not necessarily by all intermediaries. A thoughtful analysis of this basic sense of information would probably make the informants' intention to inform a criterion of information." (Machlup 1983, p.660).

The last sentence seems to be simply out of tune with what precedes it. The informants' intention to inform may contribute to a better or more attractive presentation of data or to a resort to coercion or subliminal stimuli. But as we will show, the recipient has to select and interpret data by his cognitive processes. If he does not, all the intentions of the informer are to no avail.

6.3 Redefining Information

In any good dictionary, there are many definitions of information. In the prevailing modern sense of the verb "to inform", information is "knowledge communicated concerning some particular fact, subject, or event; that of which one is appraised or told; intelligence, news." (The Oxford. Eng. Dict., 1989). This definition is ambiguous. The knowledge communicated may originally belong to either party, to the communicator or the recipient. "That of which one is appraised or told", however, points to the targeted recipient only. The informer knows it and does not need to be appraised or told. The recipient, on the contrary, has to select these data and process them mentally to "be told or appraised."

To eliminate this ambiguity, my expanded definition, below, builds on previously discussed current findings of cognitive research. Stressing the importance of the subjective human interest of the receiver constructing his information, I attempt to establish the hierarchy of attributes that determine this interest. Starting with instincts, as a common denominator of all living creatures, I list other traits, some of which, to some degree, we still share with animals. I rank them by what I think is their contribution to our efforts to elevate ourselves above animals, trying to subordinate to our values, beliefs and wisdom our cognitive activities motivated by instincts or feelings.

Information, as processed by humans, is data perceived or noticed, selected and organized by their receiver, because of his subjective human interests, originating from his instincts, feelings, experience, intuition, common sense, values, beliefs, personal knowledge, or wisdom simultaneously processed by his cognitive and mental processes, and seamlessly integrated in his recallable knowledge.

Logan points in the same direction when he says that "[t]he human mind is shaped as much by the endocrine system and endorphins as by neurons and the flow of electrons. Human thought is as much controlled by pleasure, passion, love, morality, beauty, mysticism, and curiosity as by reason and logic" (Logan, 1995, p. 260).

Information defined this way stresses subjectivity of human judgment, instead of denying its existence, as is done to suit the limitations of computers. (cf. Kent 1978, p. 94). It emphasizes the

exclusive primacy of the receiver of data. The receiver's requirements determine whether data will be selected as having relevance and purpose and become a meaningful contribution to his information.

Such information is an ideal, seldom reached outside of an involved discourse. For practical purposes, we do not always have to consider all traits of persons involved in its acquisition. "If the purpose is to arrive at an absolute definition of truth and beauty, the chances of reconciliation [of our subjective views] are nil. But for the purposes of survival and the conduct of our daily lives (relatively narrow purposes), chances of reconciliation are necessarily high." (Kent 1978, p. 202). Such would be the case of a small set of people (e.g. accounting clerks) working in an area where syntax is precise and there is no ambiguity in meanings. For them, instructions defining their function and its purpose normally override all other personal traits and considerations. Their subjective views of their tasks differ negligibly and a consensus can be engineered by a person who understands their area of activity and terminology. On the contrary, the analyst developing an Executive Support System may have to consider all the executive's attributes, while fitting the system to his requirements.

6.4 Benefits of Revised Definition

Accepting such a definition of information as created not by computers, but enacted by human receivers of data, is the essential step in changing the mindset of information systems people who delude themselves that data processed by computer will satisfy End User's' search for information. When this change occurs, instead of concentrating their attention on the processing of data, the newest technology and the systems software to use it efficiently, knowledgeable information systems people will build all systems with maximum attention to the requirements of the targeted users. They will even consider those "nice to have" features of the system that might lead to improvements or to re-engineering, as we ought to counter bureaucratic complacency and acceptance of any system that users hope will free them from responsibility.

A large portion of interesting and relevant data will be selected to become meaningful information and will affect the knowledge of the receivers, who will be able to convert them into useful information, important for the success of the organization and justifying the practical use of technology. This will emphasize the importance of intelligent and creative people, not just the newest expensive devices. The frenetic and expensive search for them would be balanced by increased staff training, and upgrading of the skills of innovation and systems design pertaining to the reshaping of the traditional ways we work. This would be also the answer to the productivity paradox of high investment and operation costs with their unrealized potential for improvement of the organizational performance. (Albrecht, 1996).

Information systems will become true Knowledge Support Systems. They will indeed be business-driven.

7 What Should We Do Now?

7.1 A Call to Action by the Association for Information Systems (AIS)

The creation of the AIS was spearheaded by William R. King of Pittsburgh University. The news came to me in the Internet newsletter (Infosys, 1994). It announced the AIS Inaugural Americas Conference to be held on August 26-27, 1995 in Pittsburgh. Its objective was to encourage diversity in the program and fairness (sic) in the review process of submitted papers. By January 1996 the association had 1600 members, of which around 200 were from Europe, the Middle East and Africa, and another 200 from Asia and Pacific regions (Bjoern- Andersen 1996). It "is a global organization, primarily of information systems academics and researchers, that are seeking to develop a positive vision for, and image of, the information systems field" (King 1996). In December 1995 the AIS Council approved a strong "Policy Statement" on "The Role of information systems in American Schools of Business."

"The underlying academic discipline 'information systems' supports the information business resource and information systems business function... The accomplishment of this critical business-value-creation objective requires that information systems specialists have a solid under-

standing of business processes and that business managers have an equally strong understanding of the information resource and of information systems." (AIS Policy 1995).

The second sentence allows us to hope that at last curricula will be developed in universities which will recognize that the parties in a discussion, such as systems analysts and end users, have to have a common language i.e., have to know something about the context of the work of the other partner.

7.2 Forthcoming Curriculum Changes

At the August 1996 Second AIS conference a session was devoted to "A Report on the Panel Discussion on the Joint Association for Computing Machinery (ACM), Association for Information Systems (AIS) and Data Processing Management Association (DPMA) Information Systems Curriculum." The chairman of the team designing it is Gordon B. Davis, the incoming president of the AIS. He said that "The cooperative curriculum report by AIS, DPMA and ACM is in final stages of approval. It is a step forward but not revolutionary – hopefully, there will be an updating process that will be faster than in the past." (Khazanchi 1996).

In another special session at the AIS conference on the information systems master's degree which does not require a technical background for admission, two professors presented a report from a quick survey which showed that in the USA there are at present 53 schools that offer such a degree. Of those, 27 are housed in schools of business, and 26 in other schools (Gray 1996).

The cautious comment of Davis and the low count of 53 schools show that we are facing a slow progress in implementing the policy statement of the AIS. This is not surprising. It is hard to say how many professors there are who are knowledgeable enough in both information systems and business processes to discuss real-life cases in a classroom in order to mold and eventually graduate "information systems specialists [who] have a solid understanding of business processes and business managers [who] have an equally strong understanding of the information resource and of information systems." (AIS Policy 1995). Will those brilliant educators be willing to stay in academia, instead of earning high salaries in industry where there is a great demand for them? If so, this would guarantee that the emphasis would shift away from the current and very costly fascination with the newest technology to a profitable deployment of systems that would contribute to the bottom line. It would give more power to those whose intelligence is indispensable to convert data supplied into useful information. (Albrecht 1996).

8 Further Research

The domain of information and knowledge will preoccupy many generations of scholars to come. There are a few hypotheses, assumptions, and suggested actions referred to in this paper. They should be further examined to check which of them are borne out by the facts.

There is no proof known at present on which to base the philosophical hypothesis about subjectivity of human cognition as accepted by Hubert Dreyfus and others, as quoted above. Dreyfus states this very clearly. There are many known experiments reported in phenomenology which support his position. Will they be generally accepted or refuted?

In my definition of information I ranked the causes of human interests by what they contribute to our efforts to distance ourselves from animals, from a low humanizing contribution for instincts to a high one for wisdom. However, the frequency of occurrence is high for instincts and low for wisdom. Negroponte seems to think similarly when he reassures the readers that "[t]oday's TV set lets you control brightness, volume, and channel. Tomorrow's will allow you to vary sex, violence, and political leaning" (1995, p. 48). This also should be researched.

It is necessary to check in other European languages whether, as in English, the use of the term "data" coincides in literature with the sudden increase in available data as it happened after Gutenberg's invention, and more importantly, after the "explosion" of data on the Internet.

What is the best way to educate those who deal with information systems providing data for humans, so that they use the proper meaning of the term "information"? This will focus their work on the interests of targeted recipients of data, and on the valuable human intelligence that can ad-

vantageously apply the mechanics of information systems.

What is the best way to promote the correct use of the terms "data" and "information" in academia, the information systems industry, and by the public at large, to eliminate the present confusion and damage done to human communications and information systems?

How big a loss of productivity from exploding investments in information technology should be attributed to the lack of concern on the part of systems analysts for the interests and needs of end users? Has it contributed significantly to the appearance on our agenda of the productivity paradox puzzle?

What was the advantage to society of continuous flow of ever newer versions of information technology hardware? Was its acquisition, need of conversion of systems to the new equipment and training justified by a real advantage? Was it just the result of planned obsolescence and an incessant marketing drive of hardware vendors or was it due to the frequently misguided belief that one must be fully aware at all times of the emerging technology, and attempt to acquire it sooner rather than later? damaging confusion?

Is it correct to assume that other things being equal, the closer the medium for human communications approaches the modalities of the face-to-face discourse, the higher is the proportion of data selected by the targeted receiver to construct information?

9 Conclusions

Assuming the subjectivity of human cognition and stressing the exclusive importance of the targeted receiver of data I eliminated the ambiguity in the almost universally accepted understanding (definition) of the term "information."

Our curricula for information systems analysts and scholarly papers should use this definition. This way they will be focused on the interests of end users, treating technology as a tool

Social scientists and planners will be able to have a realistic view of the future spiritual, intellectual and material needs of our society to properly deal with them. Their speculations about the future of humanity and culture may have to be revised for better or for worse?

Marketing experts will be able to prognosticate better where the future markets will be for programming of mass media and screens. This will be invaluable for planners and builders of the information highway.

The acceptance of this subjective definition of information dictates changes in the way we refer to it. We have to proceed with analyzing our use of the term "information" and replace it with "data" where needed. In so doing, we have to reverse the habit of possibly 20 millennia of our past. This will help us in our information-related tasks increasing clarity and order where there were confusion and chaos.

Computer scientists, as before, will concentrate on what they do best, on developing new technology and on efficient processing of data, which are not information.

Living in an information society, we have to be clear what we mean by the term "information." This is the gist of this paper. All the rest, is just commentary..

Acknowledgments

I thank Rob Kling and Marcin Paprzycki, who read an early version of my paper, as well as my anonymous reviewers, for their valuable criticism, comments and hints. I am indebted to my daughter Magda Kryt for careful editing.

References

[1] AIS, (1995): Association for Information Systems, The Role of information systems in American Business Schools, AIS Policy Statement, of Dec. 1995. `http://hsb.baylor.edu/ramsover/ais.ac.96`

[2] Albrecht L. K., retired executive and consultant in a letter to the author, July 15, 1996

[3] Binning, B. email 22 Oct 1995, `binning@AIX1.UCOK.EDU`

[4] Bjorn-Andersen, N., (1996), AIS President's Newsletter, `hhtp://hsb.baylor.edu/ramsover.ac.96`

[5] Boulding, K. E., General systems theory - the skeleton of science, Management Science, II:3 (April 1956) p. 97-98.

[6] Boulding, K. E., (1983), Systems theory, mathematics and quantification, in "The Study of Information: The Interdisciplinary Study" Machlup, F., and Mansfield, U., eds., John Wiley & Sons, New York.

[7] Branscomb, L. M., and Thomas, J. C., Ease of use: a system design challenge. IBM Systems Journal, Vol. 23 No.3 1984 p. 224-235.

[8] Buckland, M., (1991) "Information and Information Systems". Praeger, New York.

[9] CEPIS, (1995), Phase 3 Report, Council of European Professional Informatics Societies, Professional Development and Qualifications Task Force, January 11, 1995, cepis@bcs.org.uk or hhtp://www.bcs.org.uk/cepis.htm.

[10] Checkland, P.,and Scholes, J., (1990), "Soft Systems Methodology in Action," Wiley, Chichester. p.303 in Mingers (1995) p.286.

[11] Davenport, T. H., (1994) Saving information technology's Soul: Human-Centered Information Managagement, Harward Business Review, March-April 1994

[12] Debons, A, Horne, E., Cronenweth, S.,(1988) "Information Science: an integrated view." G. K. Hall & Co. Boston.

[13] Dervin, B.and Nilan, M., (1986) Information needs and uses,. in Annual Review of Information Science and Technology, vol. 21, Williams, M., E., ed., American Society for Information Science., White Plains, New York. pp. 3-33. Quoted according to the abstract available from hhtp://www.sils.umich.edu/mjpinto/ILS609Page/Bibliography/DervinAbstract.html.

[14] Dreyfus, H. L., (1992) "What Computers Still Can't Do. A Critique of Artificial Reason." The MIT Press, Cambridge, rev. edit. p. xliv.

[15] Dreyfus, H. L., and Dreyfus S. E., (1995), Making a mind vs. modeling the brain: AI back at a branchpoint. Informatica, vol. 19 Nr 4, The Slovene Society Informatika, Ljubljana, Slovenia.

[16] Dunlop, C. and Kling, R., eds. (1991), "Computerization and Controversy: Value Conflicts and Social Choices." Academic Press, Inc. San Diego

[17] Glass, R. L., (1994) quoted in Infosystems vol.1 Nr. 09, extracted from "Surveys on information systems failure", icis-l@uga.cc.uga.edu March 2-4-1994.

[18] Gray, P., MS Programs in Information Systems Without a Technical Background, email on ISWorld-L Oct. 28, 1996 gray@CGS.EDU

[19] Hartmanis, J., & Lin, H., eds., (1992). Committee to Assess the Scope and Direction of Computer Science and Technology; Computer Science and Telecommunications Board; Commission on Physical Sciences, Mathematics, and Applications; National Research Council. "Computing the Future. A Broader Agenda for Computer Science and Engineering". National Academy Press, Washington, D.C.

[20] Immen, W., (1995), The truth about selective vision., The Globe and Mail, Toronto, April 19, 1995, p. A 11. Reprinted with permission from the Globe and Mail.

[21] Informatyka (1994). A number of articles about the information systems curriculum (in Polish) Wydawnictwo Czasopism i Ksiazek Technicznych SIGMA NOT S.z.o.o. Warszawa, July 1994.

[22] Infosys., (1994) In its v1 n 43 of Nov 8 and v1 n 50 of Dec. 16,1994, edited by Denis W. Viehland,D.Viehland@massey.ac.nz

[23] Kent, W. (1978), "Data and Reality.", North-Holland Publishing Company, Amsterdam.

[24] Khazanchi, D., (1996), Suggestions, email letter of Sep. 17, 1996 on ISWorld KHAZANCHI@NKU.EDU

[25] King, W.R., (1996),email of Nov., 13-14, 1996, on ISWorld-L, BILLKING@KATZ.BUSINESS.PITT.EDU

[26] Kryt, J., (1995), Information highway or commercial data dump?, Pigulki, ISSN 1060-9288, Number 20, (text of the article available

by email from jkryt@acs.ryerson.ca or the complete issue js@uci.agh.edu.pl.

[27] Kryt, J., (1996), All that glitters on the Internet is not information, Proceedings of the Fifth International Conference, Information Systems Development - information systems development'96, Gdansk, Poland 24-26 September 1996, eds. Wrycza S. and Zupancic J., publ. Fundacja Rozwoju Uniwersytetu Gdanskiego Zaklad Poligrafii, Sopot. ***This paper is a considerably expanded and revised version of the above paper presented at the ISD96 conference. Quoted with permission.

[28] Little, B. Productivity paradox puzzles experts. Globe and Mail, Toronto, Apr.14, 1997 p. B1.

[29] Logan, R. K., (1995). "The Fifth Language: Learning the Living in the Computer Age". Stoddart Publishing Co. Limited, Toronto. p.260.

[30] McLean, R., (1985) quoted in: The Influence of business on educational practice, Ecoo Output, Toronto, vol.6 number 3, December 1985.

[31] Machlup F., (1983), Semantic quirks in studies of information, in "The Study of Information: The Interdisciplinary Study" Machlup, F., and Mansfield, U., eds., pp. 641-671., John Wiley & Sons, New York.

[32] Mingers J. C., (1995), Information and meaning: foundations for an intersubjective account, Information Systems Journal, vol. 5, pp. 285-306.

[33] Mingers J. C., (1996), Embodying information systems, in "Information Technology and Change in Organizational Work". Jones, M., Orlikowski, W., Walsham, G., and de Gross, J., eds. Chapman Hall, pp. 272- 292.

[34] Negroponte N. (1995), "Being Digital," Alfred A. Knopf, New York

[35] Norman, D. A., (1990), "The Design of Everyday Things", Doubleday, New York

[36] The Oxford English Dictionary (1989), Clarendon Press, 2nd ed. pp. 943-944.

[37] Postman, N., (1992), "Technopoly: The Surrender of Culture to Technology". A.A.Knopf, New York.

[38] Ramo, J. C., (1996, Finding God on the Web, Time, vol. 149 No. 1, Dec. 16, 1996, p. 57

[39] Roszak, T., (1986) "The Cult of Information: The Folklore of Computers and the True Art of Thinking". Pantheon Books, New York.

[40] Rowan, G., (1995), Reality bites information technology industry, The Globe and Mail, Toronto, April 17, 1995 p. B1-2.

[41] Shera, J. H., (1983), Librarianship and information science, in "The Study of Information: Interdisciplinary Messages." Machlup, F., and Mansfield, U., eds., John Wiley & Sons, New York.

[42] Simon, H. A., (1995) Technology is not a problem, in "Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists". Baumgartner, P., and Payr, S., eds., Princeton University Press, Princeton, NJ.

[43] Sowa, J. S., (1984) "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley Publishing Corporation, Reading.

[44] Varela, F. J., Thompson, E., and Rosch, E., (1991) "The Embodied Mind: Cognitive Science and Human Experience". Cambridge, The MIT Press.

[45] Weizenbaum, J., (1995, The myth of the last metaphore, in "Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists". Baumgartner, P., and Payr, S., eds., Princeton University Press, Princeton, NJ.

[46] Wintrob, S., (1995), information technology talent pool set to decline: study. Computing Canada, vol.21 Nr.19 Toronto.

[47] Wurman, R., S., (1989), "Information Anxiety", New York, Doubleday

Grundlagenstudien aus Kybernetik und Geisteswissenschaft

— GrKG / Humankybernetik

a quarterly review of the academia libroservo (AL)

An International Review for Modelling and Application of Mathematics in Humanities

GrKG/Humankybernetik (ISSN 0723–4899) belongs to the oldest [Vol. 38 (1997)], traditional, disciplinarily and internationally open journal for challenging themes, mission, and professional treatment of cybernetics. The publisher (AL) stresses that it is aimed to ensure a permanent and organized liaison between researchers whose work in various countries is related to different sectors of cybernetics, stressing the topics of the journal as follows.

Cybernetics of social systems comprises all those branches of science which apply mathematical models and methods of analysis to matters which had previously been the exclusive domain of the humanities. Above all this includes *information psychology* (including theories of cognition and 'artificial intelligence' as well as psychopatometrics and geriatrics), *aesthetics of information* and *cybernetic educational theory, cybernetic linguistics* (including text-statistics, mathematical linguistics and constructive interlinguistics) as well as *economic, social* and *juridical cybernetics.* — In addition to its principal areas of interest, the GrKG/HUMANKYBERNETIK offers a forum for the publication of articles of a general nature in three other fields: *biocybernetics, cybernetic engineering* and *general cybernetics* (theory of informational structure). There is also room for *metacybernetic* subjects: not just the history and philosophy of cybernetics but also cybernetic approaches to education and literature are welcome.

The papers are published in German, English, French and ILo, according to the choice of authors.

The journal *Grundlagenstudien aus Kybernetik und Geistewisseschaft* was founded in 1960 by Max Bense, Gerhard Eichhorn, and Helmar Frank. Currently, it is an official publication of the following scientific institutions:

Institut für Kybernetik Berlin e. V.
Gesellschaft für Kommunikationskybernetik
(Director: Prof.Dr.phil.habil. Heinz Lohse, Leipzig)

TAKIS—Tutmonda Asocio pri Kibernetiko, Informadiko kaj Sistemiko
(President: Dr. Dan Maxwell, Washington, USA; General Secretary: Ing. Milan Zvara, Poprad, Slovakia)

Akademio Internacia de la Sciencoj San Marino publikigadas siajn oficialajn sciigojn komplete en grkg/Humankybernetik

Annual Subscription Rates

A single copy 20,– DM; a volume 80,– DM plus mail expenses.

Regulations Concerning the Preparation of Manuscripts

Articles occupying more than 12 printed pages (ca. 36,000 type strokes) will not normally be accepted; a maximum of 8 printed pages is preferable. From 1982 onwards, articles in the three working-languages of the Association Internationale de Cybernetique, namely English, French and Internacia Lingvo will appear in addition to those in German. Literature quoted should be listed at the end of the article in alphabetical order of authors' names. Various works of the same author should appear in chronological order of publication (marked by a, b, ...). Quotations within articles must name the author and the year

of publication together with a letter if necessary. An abstract (500–1,500 type strokes including the title translation) in at least one of the other languages of publication should be submitted.

A Look into grkg/Humankybernetik

Let us make a short look into grkg/Humankybernetik, Vol. 38 (1997) No. 1:

Zum formellen Verstehen des Informationsphänomenalismus
[Towards a Formal Understanding of Informational Phenomenalism]
A.P. ŽELEZNIKAR (SLO) 3–14;

Subjektoperator als Raumkurven-Dreikant
[Subject Operator as a Space Curve Triangle]
H. STACHOWIAK (GER) 15–22;

Stokasta optimumigo de decidoj
[Stochastic Optimization of Decision]
I. RATKÓ (HUN) 23–30;

Mathematisch-logische Modellierung didaktischer Entscheidungsprozesse—Thesen zum Dresdner Ansatz
[Mathematical-logical Modeling of Didactic Decision Processes—Theses to the Dresden Annex]
H. KRESCHNAK (GER) & K. KARL (GER) 31–35.

Die Strukturierung des Lehrstoffes und erziehungswissenschaftliche Probleme
[Structuralization of Instruction Matter and Educational-scientific Problems]
J. ČIPERA (CZE) 36–40; and

Mitteilungen [News] 41–48.

Citations

Let us list some of citations from *grkg/Humankybernetik* with commentaries which characterize the contents of the journal.

— **38** (1997) 1, pp. 3–14, *A.P. Železnikar*, Towards a Formal Understanding of Informational Phenomenalism.

This paper, written in German, may be seen as a phenomenalistic extension of the phenomenological method of Husserl[1] and Heidegger[2], but in a substantially different way, oriented in a formal (mathematics-like) presentation, and building up a new philosophical and formal approach to the problems of the informational. One of the goals of this paper is to introduce an adequate German terminology, proceeding from the author's informational theory[3]. A short English-German dictionary of terms is the following:

arise, emerge	entstehen, werden
Being-in	Insein
Being-of	Vonsein
circular	zirkulär
circular formula	Zirkelformula
consciousness	Bewußtsein
counterentity	Gegenseiende
counterinforming	Gegeninformieren
decomposition	Zerlegung
embedding entity	Einsetzungsseiende
entity	Seiende
externalism	Externalismus, Nach-außen-sein
impacted, the	Eingewirkte
impacting, the	Einwirkende
inform	informieren
informational	informationell
informational, the	Informationelle
informational embedding	Informationseinsetzen
informational entity	Informationsseiende
informational graph	Informationsgraph
informed, the	Informierte
informedness	Informiertsein
informer	Informator
informing	Informieren
informing, the	Informierende
informingness	Informierendsein

[1] HUSSERL, E. 1900. Logische Untersuchungen. Erster Band. Prolegomena zu reiner Logik. Sechste Auflage. 1980. Max Niemeyer Verlag. Tübingen.

[2] HEIDEGGER, M. 1927. Sein und Zeit. Sechszehnte Auflage. 1986. Max Niemeyer Verlag. Tübingen.

HEIDEGGER, M. 1962. Being and Time. Translated by J. Macquarrie & E. Robinson. Harper & Row. New York.

[3] ŽELEZNIKAR, A.P. 1993. Formal Informational Principles. Cybernetica 36:43–64.

ŽELEZNIKAR, A.P. 1994. Informational Being-in. Informatica 18:149–171.

ŽELEZNIKAR, A.P. 1994. Informational Being-of. Informatica 18:277–298.

ŽELEZNIKAR, A.P. 1996. Organization of Informational Metaphysicalism. Cybernetica 39:135–162.

intention	Intention, Absicht, Existenzausdauer, Vorhaben
intentional informing	Intentiosinformieren
internalism	Internalismus, Nach-innen-sein
marker	Bezeichner
metaphysicalism	Metaphysikalismus, In-sich-selbst-sein
observer	Beobachter
phenomenalism	Phänomenalismus
star gestalt	Sterngestalt
vanish	ausklingen

This dictionary brings into the foreground some intrinsic semantics of terms which can be more originally (primordially) understood in German. In this way an English reader, with knowledge of German, can obtain a meaningly deeper and diverse insight into the informational terminology.

The rest of the paper might sound familiar to the readers of *Informatica*. There are, indeed, essential linguistic differences in writing in English and in German, however, the background of the informational philosophy remains unchanged. In German, it is possible to take into consideration the traditional philosophical terminology, which has its roots in the German philosophy.

In the Conclusion of the paper some explaining questions are answered. Did the concept of informational entity, quite at the beginning of informational theory, aim at phenomenalism of consciousness? The answer to this question is, indeed, negative. The starting point of the concept development has been the pure generality of informing of entities, which can be recognized as things, phenomena, processes. In formulas of the kind $\alpha \models$, $\models \alpha$, or $\alpha \models \beta$, there is not hidden something consciousness-like. But, is it so, in fact? Where does lie the factical potential (power) of something conscious, to be presented or described by the formalistic apparatus? Why should the traditional mathematics by it logic and its computable objects and methods not suffice?

It was stressed several times that informational operands as well as informational operators are arising (emerging) entities, which can, within the informational field of impactingness and impactedness, behave extremely changeably. In the mathematical sense, such objects are non-computable. To come to the nature of things as close as possible, simultaneously to the so-called computability, the non-computability[4] has to be considered. Saying by other words, informational entities are cybernetic objects in any respect, concerning their entirety as well as their particularity (their mutual impactingness). Thus, the presented informational concept corresponds the nature of all kinds of things, originating in their physical or mental existence.

— **38** (1997) 1, pp. 15–22, *H. Stachowiak,* Subject Operator as a Space Curve Triangle.

This article, written in German, deals with the analogy between the general model theory (GMT) of the author and the space curves (differential geometry) in the three-dimensional Euclidean space. Thus, the comparison between the GMT notion of *subject operator* of the "model relation" and the notion of differential geometrical triangle is discussed. What does the subject operator mean at all?

The connection between the cybernetic and pragmatic reasoning does not root in a positivistic arbitrariness but in a mapping of cognitive or behavioral subject to the cognitive, especially interpretation model. The Pierce's triad connects the objective Being through its symbol representation to a subjective momentum. In this way does Pierce reconnects the ontology (which is narrowed on a pure "thing as such") through a symbol theory to cognitive theory.

GMT grasps this triad, and calls the Pierce's object the original O, his "representation" the model M (of O), and his "interpretation" the subject operator S (of mapping O in M). Thus, a three-part *model relation* $P = \langle O, M, S \rangle$ with $S = \langle k, t, Z \rangle$ is obtained, where k is (the individual or collective) cognizing or acting subject, t is the time interval of the O-M-mapping existence, and Z is the goal of modeling (modeling intention).

Let's observe the space curve through its in-advance-given form of course, especially its behavior of bending. In this case, the notion of tangent

[4]Hameroff, S. & R. Penrose. 1996. Conscious Events as Orchestrated Space-Time Selections. Journal of Consciousness Studies 3:35–55.

comes in the foreground. The tangent, in a point A on the curve, is a boundary case of a sequence of secants through A and a second secant point B on the curve, where $B \equiv (x+\mathrm{d}x, y+\mathrm{d}y, z+\mathrm{d}z)$ becomes infinitely close to A. Then, the perpendiculars to the tangent through A lie in the normal plane through A. A third, to B infinitely close point C determines the so-called *snuggle plain* which normal is called the binormal. From now on the game of the curve bending begins. The curve bending in a point will be explained through the bending of a circle positioned through the points A, B and C. This circle lies on the snuggle plain of A. Its radius is called the bending radius and its reciprocal value is called the bending of the curve in A. The bending-center point in regard to point A, that is the center of the bending circle through A, lies in the normal plane of A and simultaneously on that line, in which the normal plane and the snuggle plain intersect. This curve normal is called the *main normal* of A. Yet, the notion of *triangle* D can be defined. It consists of the following three lines: the tangent t, the binormal b, and the main normal h, that is $D = \langle t, b, h \rangle$. This expression represents the system of three paired convergent lines with the common intersection point A, and corresponds to a system of mathematical equations—the so-called Frenet's formulas.

If point A, in dependence of its coordinates x, y, z, a time parameter τ, passes through the curve, then the triangle, during this passage, changes its position. On contrary, if the triangle, for a certain passage form, in a certain interval, is given by the appertaining Frenet's formulas (with the known bending and torsion), then the curve can be uniquely produced (generated). In the first case, within it lies a copy drawing, a cognitive and describing act. In the second case, there is a pre-determining, a determining and pre-scribing act, one could say a model *creative* act too, with possible aims and goals, with the meaning of pragmatic intentions. How should the space curve look like? Possibly, it could serve the technical reasons, to which a modeling operator is attributed, that is something subjective too, by which a mathematical operator becomes a pragmatic, model-theoretic "subject operator". In this way, an exchange of roles has taken place: without a semantic change of Frenet's formulas, the pragmatic 'descriptors' and 'prescriptors' (or producers) have been changed. So, the subject operator can produce a descriptive model from an intentional original.

The author dedicates a separate discussion on the problem of analogy between the three-tuple GMT subject operator and the described geometric triangle. The obvious meaning of them differs in a substantial way. In the domain of GMT, the comparison between the original and the model takes place. and this comparison can be transferred further on a model-to-model domain. The attribute and predicate classes, respectively, can be understood in the following way: In case of predicate classes K_1 and K_2, class K_2 is analogous to class K_1 (and vice versa), then and only then, if K_1 and K_2 are *isomorphic* in regard to their predication (attributing). This mean that they are structurally equal.

Being essentially different from allegories, metaphors, etc., the analogies possess often a high heuristic value in respect of the preceding. Such an example is the analogy between the atom structure and the planet system. A contrary-free isomorphism produces a speech game, by which automatically the "world" comes to the surface. Out of an analogy extracted speech form can produce the reality—or point backwards to the already produced reality.

After this, more can be said about the analogy of the subject operator and the triangle. A space curve can be interpreted as a prototype of models of arbitrary originals changing through time. Such originals are, for instance, cognitive processes in the spirit of the Pierce's triad or the model relations of GMT, respectively, where the triangle within the subject operator a descriptive as well as producing element in concern to a real process finds the analogy. In the discussion of the paper, a question is put into the foreground in the following way: Which are the possibilities to handle, in the presented manner, the so-called I-integration into the Self (e.g., consciousness, I-emerging, development of self-consciousness, etc.)?

The GMT subject operator and the triangle are model generators in the describing and prescribing view. The original of *other* models being directed by space-curve triangle (via Frenet's formulas), that is the space curve itself, belongs

to the intelligible objects of mathematics. For a strong formalist—and who of the older generation does not remind of his/her frivolous Bourbaki juvenile love—this is a horror of digressions to metaphysics of the mathematical. But already Imre Lakatos called the formalism accurately a "bulwark of philosophy of logical positivism". And already Plato taught that one can approach such transcendental metaphenomena only by '$\lambda o \gamma o \iota$' (calculation), far away from the sensual and the purposeful; in this way the true Being of the absolute remains hidden.

— **38** (1997) 1, pp. 23–30, *I. Ratko*, Stochastic Decision Optimization.

This paper, written in ILo, introduces a number $\rho(L) = h_1 + h_2 + \cdots + h_k + a_{k+1}$ concerning the logical expression $L = L_1 \vee L_2 \vee \ldots \vee L_k$, where $L_i = L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{iq_i}$ $(i = 1, 2, \ldots, n)$, and (a) a_i is the number of conjunctions of L_i $(i = 1, 2, \ldots, n)$, (b) the h_j-th conjunction of L_j is false, but staying before a true conjunction, and (c) L_{k+1} is true. If the first disjunction is true, $\rho(L) = a_1$.

The evaluating number $\rho(L)$ changes, if a change in the order of the disjunction L occurs, or a change takes place in a member conjunction L_i. The paper shows the minimization of the evaluating number and a case being the cause for the study of the problem.

— **38** (1997) 1, pp. 31–35, *H. Kreschnak & K. Karl*, Mathematical-logical Modeling of Didactic Decision Processes.

The integration of various research works in pedagogic cybernetics seems to be necessary. Such an integration can be supported by short descriptions of the works (e.g., the Dresdener Ansatz, in the presented paper). For this purpose a decision mechanism can be used considering didactic variables. The result of learning is of a special importance in the decision process of the teacher. The decision depends on the dispositional characteristic of the learner and is a component of the psychological structure. The model of the decision making is constructed by application of the Many-sorted Predicate Logic. A component is the dispositional regularity, including the learning result and the other variables. This kind of regularities form the basis for explanations and predictions. The needed deductional conclusions can be produced by computer. The mathematical-logical model is based on the Logic of Decisions. Some aspects of the "Dresdener Ansatz" are coming close to the results known in the decision theory and game theory.

— **38** (1997) 1, pp. 36–40, *J. Čipera*, Structuring of Instruction Matter and Educational-scientific Problems.

Psychological laws of the learning process define uniquely the relation between the structure of the cognitive inclination and the degree of its understanding and impression. From these psychological cognition the following can be demonstrated: If the learner inclines to a certain sort of cognition (epistemology), and puts it into the relation to the other sorts of cognition, the does the learner grasp a certain sort of cognition not only deeper but he preserves it also for a considerable longer period of time. The possibilities of instruction matters are searched for, in which an inner logical structure could be established. The paper deals with problems of structuring of instruction matter, by means of the objective methods, and methods of proposition and predicate calculus.

*
* *

The undersigned believes that these short look into grkg/humankybernetik, a journal dedicated to the informational studies too (Grundlagenstudien des Informationsbereiches in Kybernetik, Psychologie, Sprache, Ästhetik, Pädagogik usw.) will be of interest for both the readers and authors of *Informatica*.

Selected, summarized, and commented
by *A.P. Železnikar*

ERK'97

Electrotechnical and Computer Science Conference
Elektrotehniška in računalniška konferenca

September 25–27, 1997

Conference Chairman
Baldomir Zajc
University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 349, Fax: (061) 1264 630
E-mail: baldomir.zajc@fe.uni-lj.si

Conference Vice-chairman
Jurij Tasič
University of Ljubljana
Faculty of Electrical Engineering
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 440, Fax: (061) 1264 630
E-mail: jure.tasic@fe.uni-lj.si

Program Committee Chairman
Saša Divjak
University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 260, Fax: (061) 1264 647
E-mail: sasa.divjak@fri.uni-lj.si

Programe Committee
Tadej Bajd
Gerry Cain
Saša Divjak
Janko Drnovšek
Matjaž Gams
Ferdo Gubina
Marko Jagodič
Jadran Lenarčič
Drago Matko
Miro Milanovič
Andrej Novak
Nikola Pavešić
Franjo Pernuš
Borut Zupančič

Publications Chairman
Franc Solina
University of Ljubljana
Faculty of Comput. and Inform. Science
Tržaška 25, 1001 Ljubljana, Slovenia
Tel: (061) 1768 389, Fax: (061) 1264 647
E-mail: franc@fri.uni-lj.si

Advisory Board
Rudi Bric, Dali Djonlagić, Karel Jezernik, Peter Jereb, Marjan Plaper, Jernej Virant, Lojze Vodovnik

Call for Papers

for the sixth **Electrotechnical and Computer Science Conference ERK'97**, which will be held on 25–27 September 1997 in Portorož, Slovenia.

The following areas will be represented at the conference:

- *electronics,*
- *telecommunications,*
- *automatic control,*
- *simulation and modeling,*
- *robotics,*
- *computer and information science,*
- *artificial intelligence,*
- *pattern recognition,*
- *biomedical engineering,*
- *power engineering,*
- *measurements,*
- ...

The conference is organized by the **IEEE Slovenia Section** together with the Slovenian Electrotechnical Society and other Slovenian professional societies:

- Slovenian Society for Automatic Control,
- Slovenian Measurement Society (ISEMEC '97),
- SLOKO–CIGRE,
- Slovenian Society for Medical and Biological Engineering,
- Slovenian Society for Robotics,
- Slovenian Artificial Intelligence Society,
- Slovenian Pattern Recognition Society,
- Slovenian Society for Simulation and Modeling.

Authors who wish to present a paper at the conference should send two copies of their final camera-ready paper to as. Dr. Andrej Žemva to Faculty of Electrical Engineering, Tržaška 25, 1001 Ljubljana. The paper should be max. four pages long. More information on `http://www.ieee.si/erk97/`

Time schedule: Camera-ready paper due: *July 22, 1997*
Notification of acceptance: *End of August, 1997*

Call for Papers

1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)

Sponsored by the IEEE Computer Society and Co-located with
the 9th IEEE Tools with Artificial Intelligence Conference
November 3, 1997, Newport Beach, California, U.S.A.

The 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97) will provide an international forum for researchers, educators and practitioners to exchange and evaluate information and experiences related to state-of-the-art issues and trends in the areas of artificial intelligence and databases. The goal of this workshop is to expedite technology transfer from researchers to practitioners, to assess the impact of emerging technologies on current research directions, and to identify emerging research opportunities. Educators will present material and techniques for effectively transferring state-of-the-art knowledge and data engineering technologies to students and professionals. The workshop is currently scheduled for an one-day duration, but depending on the final program it might be extended to a second day.

Submissions can be in the form of survey papers, experience reports, and educational material to facilitate technology transfer. Accepted papers will be published in the workshop proceedings by the IEEE Computer Society. A selected number of the accepted papers will possibly be expanded and revised for publication in the IEEE Transactions on Knowledge and Data Engineering (IEEE-TKDE) and the International Journal of Artificial Intelligence Tools. Educational material related to papers published in the IEEE-TKDE will be posted on the IEEE-TKDE home page.

The theme of the workshop is "AI MEETS DATABASES". Topics of interest include, but are not limited to:

- Computer supported cooperative processing and interoperable systems
- Data sharing, data warehousing and metadata management
- Distributed intelligent mediators and agents
- Distributed object management
- Dynamic knowledge
- Evaluation and measurement of knowledge and database systems
- High-performance issues (including architectures, knowledge representation techniques, inference mechanisms, algorithms and integration methods)
- Information structures and interaction
- Intelligent search, data mining and content-based retrieval
- Knowledge and data engineering systems
- Quality assurance for knowledge and data engineering systems (correctness, reliability, security, survivability and performance)
- Software re-engineering and intelligent software information systems
- Spatio-temporal, active, mobile and multimedia data
- Emerging applications (biomedical systems, decision support, geographical databases, Internet technologies and applications, digital libraries, etc.)

All submissions should be limited to a maximum of 5,000 words. Six hardcopies should be forwarded to the following address.

Xindong Wu (KDEX-97)
Department of Software Development
Monash University
900 Dandenong Road
Caulfield East, Melbourne 3145, Australia
Phone: +61 3 9903 1025
Fax: +61 3 9903 1077
E-mail: xindong@insect.sd.monash.edu.au

Please include a cover page containing the title, authors (names, postal and email addresses, telephone and fax numbers), and an abstract. This cover page must accompany the paper.

Important Dates

6 copies of full papers received by: June 15, 1997
acceptance/rejection notices: July 31, 1997
final camera-readies due by: August 31, 1997
workshop: November 3, 1997

Honorary Conference Chair

Benjamin W. Wah, University of Illinois, Urban

Steering Committee

C.V. Ramamoorthy, UC Berkeley (Chair)
Farokh Bastani, University of Houston
Nikolaos Bourbakis, SUNY at Binghamton
Jeffrey J.P. Tsai, University of Illinois, Chicago
Benjamin W. Wah, University of Illinois, Urban

Conference Chair

Jeffrey J.P. Tsai, University of Illinois, Chicago

Program Chair

Niki Pissinou, University of Southwestern Louisiana

Program Co-Chairs

Xindong Wu, Monash University
Kia Makki, University of Nevada

KDEX-97 Publicity Chair

Honghua Dai, Monash University

Program Committee

Sharma Chakravarthy, University of Florida
Wesley Chu, UCLA
Honghua Dai, Monash University
John Debenham, University of Technology, Sydney
Asuman Dogac, Middle East Technical University
Opher Etzion, Technion
Ophir Frieder, Florida Tech University
Matjaz Gams, Jozef Stefan Institute
D Georgakopoulos, GTE Laboratories Incorporated
Angela Goh, Nanyang Technological University
Forouzan Golshani, Arizona State University
Abdelsalam Helal, MCC
Louis J Hoebel, USAF Rome Laboratory
Angelika Kokkinaki, University of Cyprus
Jonathan Lee, National Central University, Taiwan
K S Leung, Chinese Univ of Hong Kong
Huan Liu, National University of Singapore
Fred Lochovsky, Hong Kong Univ of Science & Technology
Hongjun Lu, National University of Singapore
Hiroshi Motoda, Osaka University
Mike Mulder, University of Nebraska
Tamer Ozsu, University of Alberta
Eun Kyo Park, UMKC
Dave Robertson, Edinburgh University
Hayri Sever, Hacettepe University
Peter Scheuermann, Northwestern University
Heinz Schmidt, Monash University
Il-Yeol Song, Drexel University
Leon Sterling, University of Melbourne
V.S. Subrahmanian, University of Maryland
X. Sean Wang, George Mason University
Joe Wigglesworth, IBM
Philip Yu, IBM
John Zeleznikow, La Trobe University
Chengqi Zhang, University of New England
Ning Zhong, Yamaguchi University

Further Information on WWW
`http://www.sd.monash.edu.au/kdex-97`

THE MINISTRY OF SCIENCE AND TECHNOLOGY OF THE REPUBLIC OF SLOVENIA

Address: Slovenska 50, 1000 Ljubljana, Tel.: +386 61 1311 107, Fax: +386 61 1324 140.
WWW:http://www.mzt.si
Minister: Lojze Marinček, Ph.D.

The Ministry also includes:

The Standards and Metrology Institute of the Republic of Slovenia
Address: Kotnikova 6, 61000 Ljubljana, Tel.: +386 61 1312 322, Fax: +386 61 314 882.

Slovenian Intellectual Property Office
Address: Kotnikova 6, 61000 Ljubljana, Tel.: +386 61 1312 322, Fax: +386 61 318 983.

Office of the Slovenian National Commission for UNESCO
Address: Slovenska 50, 1000 Ljubljana, Tel.: +386 61 1311 107, Fax: +386 61 302 951.

Scientific, Research and Development Potential:

The Ministry of Science and Technology is responsible for the R&D policy in Slovenia, and for controlling the government R&D budget in compliance with the National Research Program and Law on Research Activities in Slovenia. The Ministry finances or co-finance research projects through public bidding, while it directly finance some fixed cost of the national research institutes.

According to the statistics, based on OECD (Frascati) standards, national expenditures on R&D raised from 1,6 % of GDP in 1994 to 1,71 % in 1995. Table 2 shows an income of R&D organisation in million USD.

Total investments in R&D (% of GDP)	1,71
Number of R&D Organisations	297
Total number of employees in R&D	12.416
Number of researchers	6.094
Number of Ph.D.	2.155
Number of M.Sc.	1.527

Table 1: Some R&D indicators for 1995

	Ph.D.			M.Sc.		
	1993	1994	1995	1993	1994	1995
Bus. Ent.	51	93	102	196	327	330
Gov. Inst.	482	574	568	395	471	463
Priv. np Org.	10	14	24	12	25	23
High. Edu.	1022	1307	1461	426	772	711
TOTAL	1565	1988	2155	1029	1595	1527

Table 2: Number of employees with Ph.D. and M.Sc.

Objectives of R&D policy in Slovenia:

- maintaining the high level and quality of scientific technological research activities;
- stimulation and support to collaboration between research organisations and business, public, and other sectors;
- stimulating and supporting of scientific and research disciplines that are relevant to Slovenian national authenticity;
- co-financing and tax exemption to enterprises engaged in technical development and other applied research projects;
- support to human resources development with emphasis on young researchers; involvement in international research and development projects;
- transfer of knowledge, technology and research achievements into all spheres of Slovenian society.

Table source: Slovene Statistical Office.

	Basic Research		Applied Research		Exp. Devel.		Total	
	1994	1995	1994	1995	1994	1995	1994	1995
Business Enterprises	6,6	9,7	48,8	62,4	45,8	49,6	101,3	121,7
Government Institutes	22,4	18,6	13,7	14,3	9.9	6,7	46,1	39,6
Private non-profit Organisations	0,3	0,7	0,9	0,8	0,2	0,2	1,4	1,7
Higher Education	17,4	24,4	13,7	17,4	8,0	5,7	39,1	47,5
TOTAL	46,9	53,4	77,1	94,9	63.9	62,2	187,9	210,5

Table 3: Incomes of R&D organisations by sectors in 1995 (in million USD)

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 61000 Ljubljana, Slovenia
Tel.:+386 61 1773 900, Fax.:+386 61 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in `.eps` format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Rudi Murn, Informatica, Institut Jožef Stefan, Jamova 39, 61111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than five years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Referees:

Witold Abramowicz, David Abramson, Kenneth Aizawa, Alan Aliu, John Anderson, Daniel Beech, Catriel Beeri, Fevzi Belli, Istvan Berkeley, Azer Bestavros, Balaji Bharadwaj, Jacek Blazewicz, Laszlo Boeszoermenyi, Jeff Boone, Ivan Bratko, Jerzy Brzezinski, Marian Bubak, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Netiva Caftori, Ryszard Choras, Jason Ceddia, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, David Cliff, Travis Craig, Tadeusz Czachorski, Milan Češka, Pavol Ďuriš, Sait Dogru, Georg Dorfner, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Hesham El-Rewini, Pierre Flener, Terrence Forgarty, Hugo de Garis, Eugeniusz Gatnar, James Geller, Michael Georgiopolus, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Inman Harvey, Elke Hochmueller, Rod Howell, Tomáš Hruška, Ryszard Jakubowski, Piotr Jedrzejowicz, Eric Johnson, Polina Jordanova, Li-Shan Kang, Roland Kaschek, Jan Kniat, Stavros Kokkotos, Kevin Korb, Gilad Koren, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Phil Laplante, Bud Lawson, Ulrike Leopold-Wildburger, Joseph Y-T. Leung, Raymond Lister, Doug Locke, Matija Lokar, Jason Lowder, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Roland Mittermeir, Madhav Moganti, Tadeusz Morzy, Daniel Mossé, John Mueller, Hari Narayanan, Jaroslav Nieplocha, Jerzy Nogieć, Stefano Nolfi, Tadeusz Pankowski, Warren Persons, Stephen Pike, Niki Pissinou, Gustav Pomberger, James Pomykalski, Gary Preckshot, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Luc de Raedt, Ewaryst Rafajlowicz, Wolf Rauch, Peter Rechenberg, Felix Redmill, David Robertson, Marko Robnik, Ingrid Russel, A.S.M. Sajeev, Bo Sanden, Vivek Sarin, Iztok Savnik, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, William Spears, Hartmut Stadtler, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Tomasz Szmuc, Jiři Šlechta, Zahir Tari, Jurij Tasič, Piotr Teczynski, Ken Tindell, A Min Tjoa, Wieslaw Traczyk, Marek Tudruj, Andrzej Urbanski, Kanonkluk Vanapipat, Alexander P. Vazhenin, Zyunt Vetulani, Olivier de Vel, John Weckert, Gerhard Widmer, Stefan Wrobel, Janusz Zalewski, Yanchun Zhang

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or Board of Referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board and Board of Referees are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Volume 21 Number 2 June 1997 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Contents: