Software Theories for Machines: Open-World Semantics for the Knowledge in Machines

Lai Heng Chua, Research Assistant
Boyd C. Paulson, Jr., Ohbayashi Professor of Engineering
Thomas Michael Froese, Research Assistant

Department of Civil Engineering
Stanford University
Stanford, CA 94305-4020
U.S.A.

ABSTRACT

The knowledge needs of an agent may be met by exploiting the knowledge environment. In particular, knowledge can be obtained from other agents or a variety of knowledge repositories such as databases. Humans readily satisfy much their knowledge needs in the way just mentioned, but there is still a lack of a conceptual framework to open the way for machines to use knowledge more extensively and intelligently. A major part of the problem lies in the difficulty of accessing knowledge in the environment, but even if that were to be solved we would still need to revise our concepts of how machine agents are to use information. The latter problem is explored taking the viewpoint that an agent is a user of knowledge. Humans and programs could be considered knowledge-using agents. We can evaluate their techniques and effectiveness in using knowledge. Comparing and contrasting the ways humans and intelligent programs use knowledge helped us identify and better understand one of the sources of human management abilities. A major part of management involves intelligently seeking, obtaining and assembling knowledge. The incorporation of some of these abilities in automation systems will increase their autonomy and management capabilities. A simple way for an agent to begin actively tapping into external knowledge that involves modifications to the reasoning cycle is suggested and research in the area of environmental intelligence is continuing.

1. INTRODUCTION

1.1 Knowledge Requirements

To design and build a facility requires the application of considerable amounts of knowledge and effort. However, each agent has only a fraction of this total knowledge and of these abilities. Therefore, there must be a process of assembling the required knowledge and abilities, and this process must continue over the duration of the project. While we are familiar with the need to assemble materials, equipment and energy, and perhaps tools, and are explicit about it, we perform the tasks of assembling knowledge so smoothly and automatically that they are hardly noticed. The need to assemble knowledge becomes much more visible when automation in considered. Automated machines need knowledge to drive them and suddenly we are faced with a host of questions: Where is the knowledge? When is the knowledge available? What determines what knowledge is needed? And many more.

Knowledge and abilities are assembled from the surrounding milieu – the environment. In particular, knowledge is assembled from sources of knowledge in the knowledge environment. The knowledge needs of an agent may be met by other agents or a variety of knowledge repositories such as databases. Humans readily satisfy their knowledge needs in the way just mentioned, but there is still a lack of a theory to open the way for machines to use knowledge more extensively and intelligently. A major part of the problem lies in the difficulty of accessing knowledge in the environment, but even if that were to be solved we would still need to revise our concepts of how machine agents are to use knowledge.

1.2 Computerization of Knowledge

There is a trend toward computerization of knowledge. This offers many advantages, such as the ability to apply the computational power of computers to interpret and render information. We also get compact storage of information, long distance retrieval over computer networks, and ease of duplication and modification relative to centuries-old paper-based techniques of information management. These efforts are directed towards supporting human knowledge needs. But they will also help support the knowledge needs of automation systems.

Nevertheless, we do face difficulties which suggest that it may not be appropriate to require computerization of all potentially required knowledge. A lot of knowledge will have to be left outside of computer systems. Consider the problem of maintaining a complete and correct database of the phone numbers and addresses of agents that potentially interact with an organization. It would be useful to have such a database, but to ensure that it is complete and correct goes too far. The effort required, even if it were possible, would be out of proportion to the benefits derived. This is particularly so if only a fraction of the information, albeit an unknown fraction, will actually be used. Maintaining a complete and correct model of a dynamically changing construction site might be another expensive exercise. It would be necessary to have a model, but we should relax the conditions of correctness and completeness. The spatially distributed aspect of these problems makes it difficult to collect the information and the dynamic nature quickly renders the collected information obsolete.

1.3 Environmental Intelligence

Having to use information from information sources that are possibly incorrect or incomplete presents many difficulties for an agent to deal with. It is suggested that agents be provided with information gathering abilities which are generally demand or failure-driven. The ability of machine agents to intelligently seek information when needed may be exploited to reduce the need to prime such agents with information or enter considerable amounts of information into computer systems. We may also be able to obtain a greater degree of robustness. Providing such capabilities is a challenge. While we seek to expand the intelligence of agents, we would like to avoid falling into computational black-holes, something which is all too likely.

2. Information User

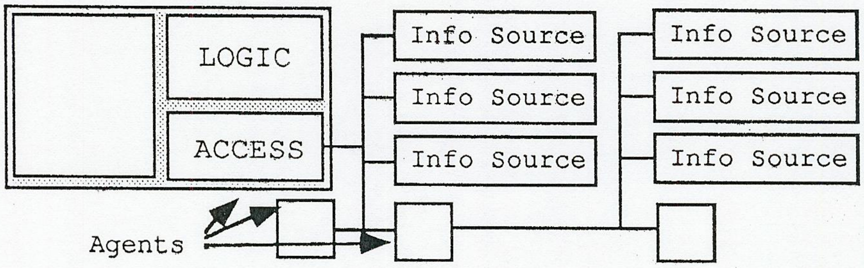


Figure 1. The Information User

First take the point of view that an agent is a user of information (Figure 1). This gives us a useful perspective because we can now ask questions about how information is used by an agent. What information can be used by the agent? Where does the information originate? Was the agent a passive recipient or the information or does the agent actively participate in obtaining the information?

2.1 Extent of Reachable and Useable Information

What information can be reached by the agent? First, we distinguish between information that can be reached by an agent and the information that the agent can use directly. An agent can make a direct contribution from the information that can be both reached and that it has the ability to use. This is depicted in Figure 2. It might not appear obvious, but there is utility in the ability to reach information that the agent does not know how to use. We are also concerned about information that an agent can use but cannot reach.

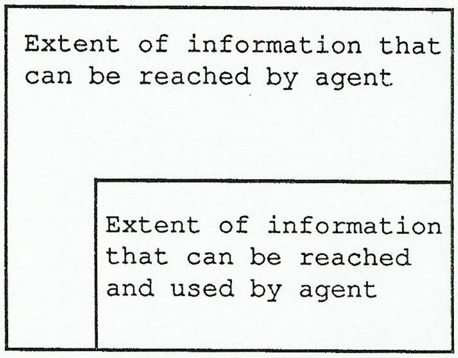
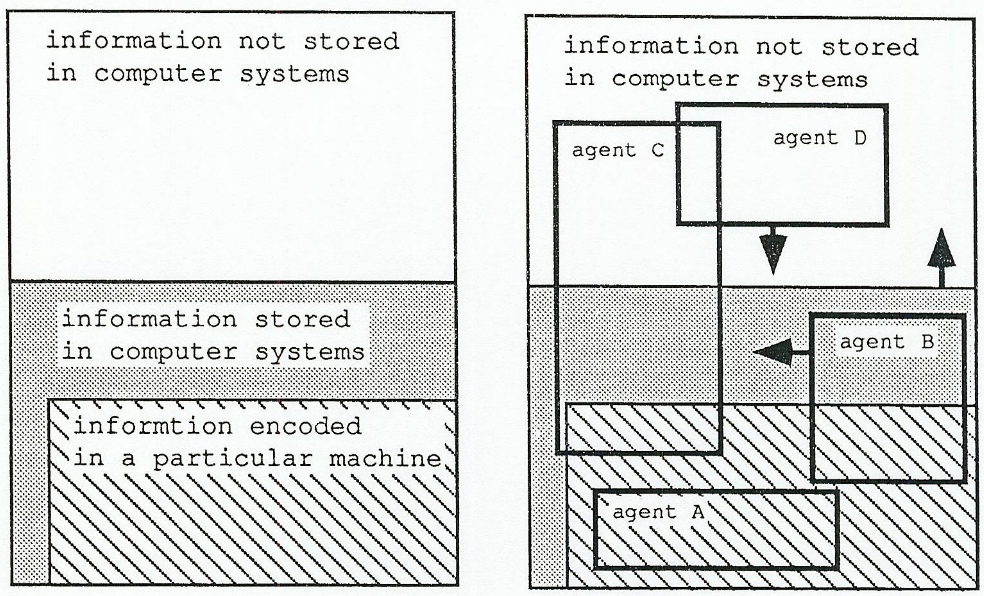


Figure 2. Accessible and Useable Information



a) Location of Information

b) Extent of Accessible Information

Figure 3. Extent of Active Accessibility of Information in Different Sources

In terms of the location of information, for our discussion it seems that a division of information according to where it is stored would be appropriate. This is depicted in Figure 3 (a). On the adjacent figure we indicate the extent of actively-reachable information by several agents relative to the original location of the information. Agent A only uses information previously present in a particular machine. Many computer programs would have the characteristics of agent A. Agent B uses information previously residing on several machine sites. A number of such programs are in existence. Typically, the information use and access is hard-wired. Several such approaches have been devised. In distributed databases the agents do not need to know about the logical or physical distribution of the information. Finding and transporting the information is handled for them transparently. On the other hand, some approaches in federated databases allow agents to refer to sites. When we talk about integration, we typically talk about expanding agent B's reach across more physical and logical machine boundaries to encompass the shaded region (as indicated by arrow). We also refer to efforts to move the boundary of the shaded region into the white region as indicated by the arrow. There is a large number of interesting problems and much work to do in these two directions. Agent C can reach information previously in many computer systems as well as information not yet in such systems. This might be the typical modern human agent. We also want to get computer inaccessible agents such as agent D to access information more extensively.

2.2 Semantics of Information Use

Suppose that an agent has easy access to information in a single knowledge base. For example, a Prolog program has easy access to predicates and rules in the current Prolog database and a human has easy access to information stored in human memory. If that agent restricts its use of information to that single knowledge base, treating the information as being "all there is," then we say that type of information use follows closed-world semantics. On the other hand, if the agent recognizes the existence of external information (reachable or unreachable) and considers that the information it has on hand possibly represents only a limited extent of information, then we regard that agent's use of information as following open-world semantics. We can generalize this to evaluate the use of any block of information by an agent.

To llustrate suppose there is a file, *f-subs*, containing a short list of building subcontractors and that our agent of interest, say agent A, knows about it. We ask agent A to provide us with a sorted list of building subcontractors. How does agent A treat the information in *f-subs*? An agent who follows closed-world semantics on information would consider the information as "all there is," and would return a sorted list of the building subcontractors contained in *f-subs*. It has no inkling that such a list may be incomplete or incorrect, and it would never dream of taking any further action. This is fine for a sorting program but limited for an agent. On the other hand, a human agent would treat *f-subs* as containing only partial information, and if resources and time permit it will do a better job of providing you with the information you requested. These abilities rest on open-world semantics, knowledge about the surrounding knowledge environment, and various other aspects of environmental intelligence.

So far we have provided an informal explanation of different semantics of information use that help us distinguish between the different knowledge-using behavior of agents. Actually, there is a range in sophistication for agents to use external information. The hard-link is one possible and currently practical way. [Levesque 83] provides a formal framework to allow a system to determine what is known and what is unknown. It introduces a connective K which is to be interpreted as "known to be true." Therefore, an agent can include statements about what is known to be true. It was shown to be very useful in making fine distinctions about what is known and unknown. This has strong correspondence to the intuitive sense of these terms. However, the proof theory is impractical for actual use because it is computationally intractable.

2.3 Lack of Intelligence in the Way Information is Used

Intelligence is difficult to define but we can still make general observations about the intelligence, or the lack thereof, of the way an agent works with information. For example, if an agent continues to insist on using information that results in failure, that agent is lacking intelligence in using information. An agent that keeps trying to achieve a goal but continually fails might be regarded as a fanatic. It is sometimes forgivable to be a fanatic, but if the same deterministic method is used to try to achieve the goal again and again, then the agent might be thought as showing lack of intelligence.

The ability to deal with some aspects of incomplete or incorrect information is part of the intelligent way for agents to work with information. We have already hinted that there may be better ways than blindly shuffling information from one world to another. We shall discuss two other aspects of intelligent use of information in Section 4.

3. PARTIAL DATA

An aggregation of data, for example that in a database or knowledge base, is partial if it does not represent all the data that is potentially relevant to the functionality of the database or knowledge base. For example, a database that contains information about only half the employees of an organization but that is supposed to answer queries about the company's employees is a partial database. Let *W* be the totality of information available in the world about that topic and let *D* be the information captured in the knowledge base (Figure 4). *D* is a subset of *W*. If *D* is a proper subset of *W*, then there may be ways to extend *D* by obtaining from the environment information in *W* that is not already in *D*. If such extension is actively made by the knowledge base itself, then we call that process self-enhancement. The knowledge base itself recognizes and tries to make up for its lack.

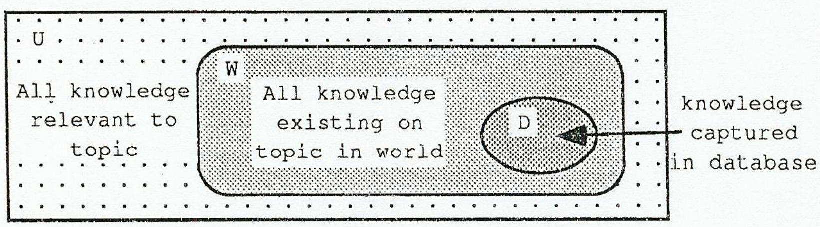


Figure 4. Partial Data and Knowledge

The notion of self-enhancement is somewhat different from the notion of default reasoning. In default reasoning [Reiter 78] assumptions are made about the nature of *U* - *D* (Figure 4). For example, in default reasoning, if neither p (a fact) or $\neg p$ can be determined with *D*, the agent might assume that $\neg p$ (or p) holds in *U* - *D*. In self-enhancement agents may take actions to obtain the knowledge in *W* - *D* to reduce *U* - *D*. Of course, it is not possible to obtain knowledge in *U* - *W* from the knowledge environment. If the agent cannot spare the effort to increase its knowledge, then it might fall back on default assumptions such as using default rules or circumscription.

Self-enhancement is, in a sense, learning, but it lacks the generality and power of what is commonly construed as learning. Some of the latter type of learning might be considered as expanding *W*. Instead of finding patterns in data or determining models, self-enhancement is merely the process of actively reaching out to obtain already existing knowledge. Since the amount that is already known is considerable, it would seem that providing agents with self-enhancement abilities would confer useful advantages. We are moving toward the construction of computer-based agents that can treat the world as a database.

3.1 Extension of data

This section discusses a failure-driven self-enhancement of data scheme using metalevel information. A failure-driven scheme appears to be better than an anticipatory one.

The systematic way of using knowledge in Prolog might be described by the metalevel program below.

```
solve(true).                                                    (1)
solve((GoalA,GoalB)) :- solve(GoalA),solve(GoalB).              (2)
solve(Goal) :- clause(Goal,SubGoals),solve(SubGoals).           (3)
```

This procedure is SLD-resolution. To prove a conjunction of goals, prove each separately, and to prove a goal, prove its subgoals. State information is kept to enable the system to determine the position with regard to its use of knowledge in the Prolog database.

```
solve(true).                                                    (4)
solve((GoalA,GoalB)) :- solve(GoalA),solve(GoalB).              (5)
solve(Goal) :- clause(Goal,SubGoals),solve(SubGoals).           (6)
solve(Goal) :- extensible(Goal),
               extend?(Goal),
               selfEnhance(Goal),
               solve(Goal).                                     (7)
```

The metalevel program above has an additional procedure (Equation 7, cuts left out). This statement is meant to express the strategy that if you have failed to prove the goal, then check if the goal is extensible. If it is an extensible goal, then obtain information about the topic of the goal and add it to the system's knowledge base. If you manage to obtain and add new information about the topic of the goal, then use the new information to continue solving for the goal. The scheme can be seen as one of failure-driven data self-enhancement.

3.2 Data Extension Considerations

There can be much sophistication in determining what to extend. As a suggestion, since the human user probably knows more about the nature of the knowledge environment, he might simply provide the information. Further considerations could include when and how to self-enhance. This would require additional information for reasoning about and carrying out these data-gathering activities, as well as maintaining bookkeeping information about such activities.

For example, one might have facts asserting that we know only a fraction of the topic *companyName*. This is all right for practical purposes since one can never realistically exhaust all the company names. In other cases we might have to be more subtle. For instance, a person has only two parents, and while we do not know who those are, we might try to find out about them. After finding out about both of them we would no longer wish to try to find more parents for that particular individual. These are known as predicate completeness conditions which are stopping conditions. Currently such stopping conditions would have to be provided by the human user. The number of employees of a company might be a parameter in several of the stopping conditions for the data extension of the employee database. Once we have obtained information for all employees, we should stop. But we should resume data extension if the number of employees increased and we are lacking in information about some of them which we now need.

That we know our data is partial is not licence to rush out to fix the problem. This issue relates to intelligent use of information and resources. Is the "goal" worthy of extension now? Again one might take simple or increasingly sophisticated approaches or combinations thereof, depending on the user. It makes sense to combine "extend?" and "selfEnhance" of Equation 7. Depending on the techniques available to obtain external information, the system might elect to make individual decisions about whether to use each of them. In a sophisticated system one might compare the expected cost of doing data extension to the expected benefits. The agent can then have the sense to use cheaper, more easily accessible information sources before attempting to use more expensive ones, or to use more reliable information sources before using less reliable ones, depending on the situation. The accessibility of information is graded.

One useful heuristic when using relatively static information sources is not to ask exactly the same questions of the same information source. One reason is that if information in a source is not expected to change much, then asking the same question would be likely to produce the same answer, with no resulting gain in information to the agent. Of course, if the information source happens to be a human, then repetition of the same question could be an annoyance. It might be useful then for the system to keep track of information requests and the success or failure of these requests. A search-tracking system can be built that uses concepts of truth maintenance.

4. Describing and modelling knowledge

In section 3.1 we described a simple technique of describing the extensibility of data, and here we would like to take the concept further. Two directions to address what we saw as the problem of lack of intelligence in using information will be described. We are in the process of implementing such a system. In particular, we would like to discuss capturing some knowledge about the dynamics of the contents of knowledge bases. For a simple approach we might start with the notion of the dynamic characteristic of certain data which we call half-life. The half-life of information on a topic is the expected time for half the information on the topic to have changed. For instance, the half-life of employee information would correlate with the turnover rate of employees in a company. If you are using information about employees of the company which you do not update regularly, then you can estimate the probability that a particular use will involve incorrect information. This is a rather crude measure of the dynamics of information on particular topics.

More intelligent decisions might be made on the basis of information about the dynamics of the information to be used. To know that it is reasonably safe to use some data (say home address, phone number, prices), an agent has to understand the dynamism of that particular piece of information. Home address information is typically stable over long periods of time, whereas price data, as we have seen, is likely to change. Thus, if you have a piece of information about the address of someone that is a year old, it is safe to use it. On the other hand, if you are using particularly significant price information as the basis of your bid, then it would not be safe to use price information that you had acquired more than a month ago; information that is a week old is borderline. It is best for you to obtain the most current information. The concept of half-life is not quite enough to deal with the price information problem, but we will leave discussion on it to a future report.

The next aspect of the problem we are trying to elucidate is the characterization of the usefulness of particular information for an agent involved in particular tasks — usefulness is usually relative and expresses relationships among knowledge. Usefulness of information can range from almost always useful to useless. We do not foresee being able to fully elucidate this problem, but rather we hope to understand enough to begin the development of schemes that will allow us to develop machine agent that use information more intelligently. First, we consider the far end of the spectrum, the idea the some block of information is almost useless for a particular problem. This is brother to the notion of independence. For instance, for the problem of driving around in San Francisco, information of and about other places is quite irrelevant. This is still too vague for an implementable concept, but it gives us a hint. Suppose we can provide a way to ascribe a location metric to location-affected information. Then we have to devise a scheme to select information for use for a problem that is spatially sensitive. The spatial problem is multi-dimensional and may be a little difficult, so we are looking at the single dimensional temporal relevance problem and temporally sensitive problems. For instance, if we are looking for contemporarily old information we would not look into recent newspapers or journals. It is more likely for us to find the information in old books or archives. From this example, it can be seen that the previously mentioned problem on the handling the dynamics of information is an expression of the more general problem of information utility. Searching efficiently for information in the appropriate temporal space would, we believe, require an efficient indexing and retrieval scheme.

5. Problems and Tradeoffs

Having described a scheme that might help in extending data, is the method generalizable to the extension of knowledge? Can a system be devised that can actively drive toward obtaining knowledge in areas of demand? While we might have some success in using modified techniques to develop extensible databases, there are many more problems in the active extension of knowledge in general.

We have also mentioned the computation problem that would attend using some of the available formal approaches for reasoning about knowledge in incomplete knowledge bases [Levesque 83] or about knowledge in distributed systems [Halpern and Vardi 87]. There was an explosion of research on what should constitute commonsense implication of knowledge in cases where actual information is unavailable.

6. Conclusions

We have explored the question of how an agent or a system uses information and concluded that the current methodology places restrictions on intelligent machines. While capabilities are being developed to enable machine to interact more extensively with the physical environment, the allowed scope of interaction of machines with the knowledge environment remains confined. To enable intelligent programs and automation systems to achieve greater autonomy requires us to reexamine the fundamental underpinnings of knowledge use by such systems. Some of these restrictions may be relaxed. Merely placing knowledge within reach of machine agents does not quite solve the problem. It is a first step, but what we are ultimately reaching for is the intelligent exploitation of knowledge by such agents. [Chua 90a, 90b] describe in more detail the capabilities of the system.

References

Chua, L. H. 1990a. 'A Theory of the Knowledge Environment for Construction Automation,' Engineer thesis, Civil Engineering Department, Stanford University, Stanford, California.

Chua, L. H. 1990b. 'Automation Systems that can use Knowledge in their Environment,' PhD thesis, Civil Engineering Department, Stanford University, Stanford, California. (in preparation).

Halpern, J. Y., and Vardi, M. Y. 1987. 'The Complexity of Reasoning about Knowledge and Time, I: Lower Bounds,' Research Report RJ 5764 - 58103, IBM Almaden Research Center, San Jose, CA.

Levesque, H. J. 1983. 'The Interaction with Incomplete Knowledge Bases: A Formal Treatment,' 7th IJCAI, Vol. 1, pp. 240-245.

Reiter, R. 1978. 'On Closed World Data Bases,' Logic and Data Bases, Gallaire, R. and Minker, J. (eds), Plenum Press, pp. 55-76.