Challenges for Machine Learning in Cooperative Information Systems

Munindar P. Singh[1] * and Michael N. Huhns[2]

[1] Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
singh@ncsu.edu
[2] Department of Electrical & Computer Engineering
University of South Carolina
Columbia, SC 29208, USA
huhns@sc.edu

Abstract. *Cooperative Information Systems (CISs)* are multiagent systems with organizational and database abstractions geared to the large **open** heterogeneous information environments of today. CIS is also the name of the associated research area, which has emerged from the synthesis of distributed databases and distributed artificial intelligence. In CIS, software agents mitigate an information environment's heterogeneity by interacting through common protocols, and manage its large size by making intelligent local decisions without centralized control. In order to cope with the dynamism presented by open environments, CIS agents must have the ability to adapt and learn. We discuss some of the most important problems involving learning and adaptivity in CISs, including requirements for reconciling semantics and improving coordination. We present a "customers' view" of learning technology as might find ready application in CISs.

1 Introduction

Due to the proliferation of networking, the desires of almost everyone to be interconnected, and the needs to make data accessible at any time and any place, modern information environments have become large, open, and heterogeneous. They are composed of distributed, largely autonomous, often legacy-based components. *Cooperative Information Systems* introduce software agents into such environments to deal with these characteristics. The agents represent the components in interactions, where they mediate differences and provide a syntactically uniform and semantically consistent middleware. Their greatest difficulty

* Munindar P. Singh was partially supported by the NCSU College of Engineering, by the National Science Foundation under grants IRI-9529179 and IRI-9624425, and IBM Corporation.

in achieving uniformity and consistency is the dynamism that open environments introduce.

Open environments are becoming an increasing part of the modern milieu through applications such as information search, electronic commerce, and virtual enterprises. They typically have the following key distinguishing characteristics:

- span enterprise boundaries;
- have components that are heterogeneous in a number of ways, such as the underlying database management systems used, and the semantics associated with the information stored or manipulated;
- comprise information resources that can be added or removed in a loosely structured manner;
- lack global control of the content of those resources, or how that content may be updated; and
- incorporate intricate interdependencies among their components.

To build systems that work effectively within open environments requires balancing their ease of construction and robustness with their flexibility. There are a number of technical difficulties specific to building systems for open environments. Foremost among these are the need to handle the unpredictability in the environment as new components appear, and old ones disappear or change. Since the information components in the environment cannot easily be altered, the agents that represent them must be able to learn and adapt. This provides new challenges for machine learning, as summarized in Table 1:

Traditional Machine Learning	**CIS Machine Learning**
Agent learns about its environment, which is passive and has no intentions	Agent learns about its environment, which is *active*, because it includes other agents who have intentions, commitments, beliefs, and abilities, and can learn
Agent might have imprecise sensors that cause it to learn inaccurate information about the environment	Agent might deliberately be misled about the environment by other agents

Table 1. Machine Learning for Cooperative Information Systems

Section 2 introduces cooperative information systems, and their quintessential applications. Sections 3, 4, and 5 discuss the machine learning challenges in extracting semantics from passive components, coordinating active components, and abstracting and structuring CISs, respectively. Section 6 concludes with a discussion of the main themes of CIS, and how they relate to machine learning techniques.

2 An Overview of CIS

Cooperative Information Systems (CISs) are an increasingly popular approach that seeks to maximize the above properties through the use of combinations of techniques from distributed artificial intelligence, databases, and distributed computing. The term *cooperative information systems* also refers to the research area that focuses on building such systems.

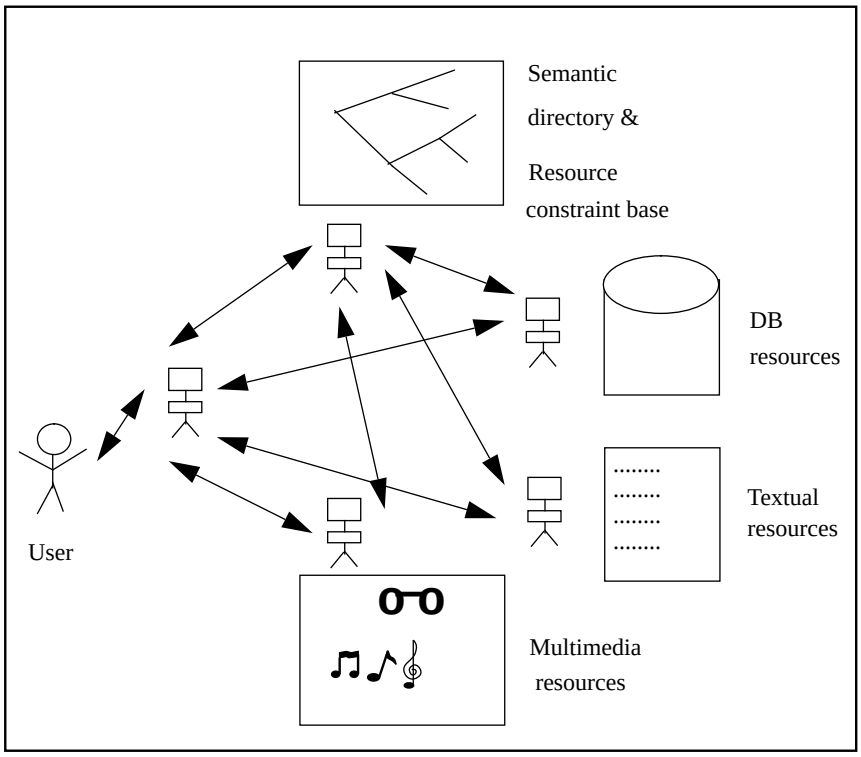

Fig. 1. A CIS Schematically

We define an *agent* as an active, persistent computational entity that can perceive, reason about, and act in its environment, and can communicate with other agents. Agents are autonomous to varying degrees to reflect the autonomy of the information resources or humans whom they represent. Figure 1 shows a CIS schematically. In this figure, we consider an environment consisting of a variety of information resources, coupled with some kind of a semantic directory. The semantic directory contains information about the resources, including any constraints that apply to their joint behavior.

Each component of the environment, as well as the human user(s), is modeled as associated with an agent. The agents capture and enforce the requirements of

their associated parties. They interact with one another appropriately, and help achieve the robustness and flexibility in behavior that is required. The charm of agents is that they provide a natural means for acquiring, managing, advertising, finding, fusing, and using information over uncontrollable environments. Further, agents are inherently modular, and can be constructed locally for each resource, provided they satisfy some high-level protocol of interaction.

The applications of CISs are varied. They involve the purely informational ones, such as database access, information malls, workflow management, electronic commerce, and virtual enterprises. They also include physical ones, such as sensor arrays, manufacturing, transportation, energy distribution, and telecommunications.

The above motivates the interest in cooperative information systems. But, as remarked above, CISs involve combining ideas not only from the study of agents, but also from databases and distributed computing. We discuss the specific challenges posed by CISs next. In doing so, we review two of the quintessential applications of CIS: information access and workflow management.

3 Extraction of Semantics

Learning about passive—and often preexisting—components, such as databases and knowledge bases.

Information access involves finding, retrieving, and fusing information from a number of heterogeneous sources. At the level of abstraction that concerns CIS, we are not concerned with network connectivity or the formatting variations of data access languages. Rather, our concern is with the meaning of the information stored. It is possible, and indeed common, that when different databases store information on related topics, each provides a different model of it. The databases might use different terms, e.g., *employee* or *staff*, to refer to the same concept. Worse still, they might use the same term to have different meanings. For example, one database may use *employee* to mean anyone currently on the payroll, whereas another may use *employee* to mean anyone currently receiving benefits. The former will include assigned contractors; the latter will include retirees. Consequently, merging information meaningfully is nontrivial. The problem is exacerbated by advances in communications infrastructure and competitive pressures, because different companies or divisions of a large company, which previously proceeded independently of one another, are now expected to have some linkage with each other.

The linkages can be thought of as semantic mappings between the application (which consumes or produces information), and the various databases. If the application somehow knows that *employee* from a database has one meaning, it can insert appropriate tests to eliminate the records it does not need. Clearly, this approach would be a nightmare to maintain. The slightest changes in a database would require modifying all the applications that access its contents! This would be a fundamental step backward from the very idea of the database

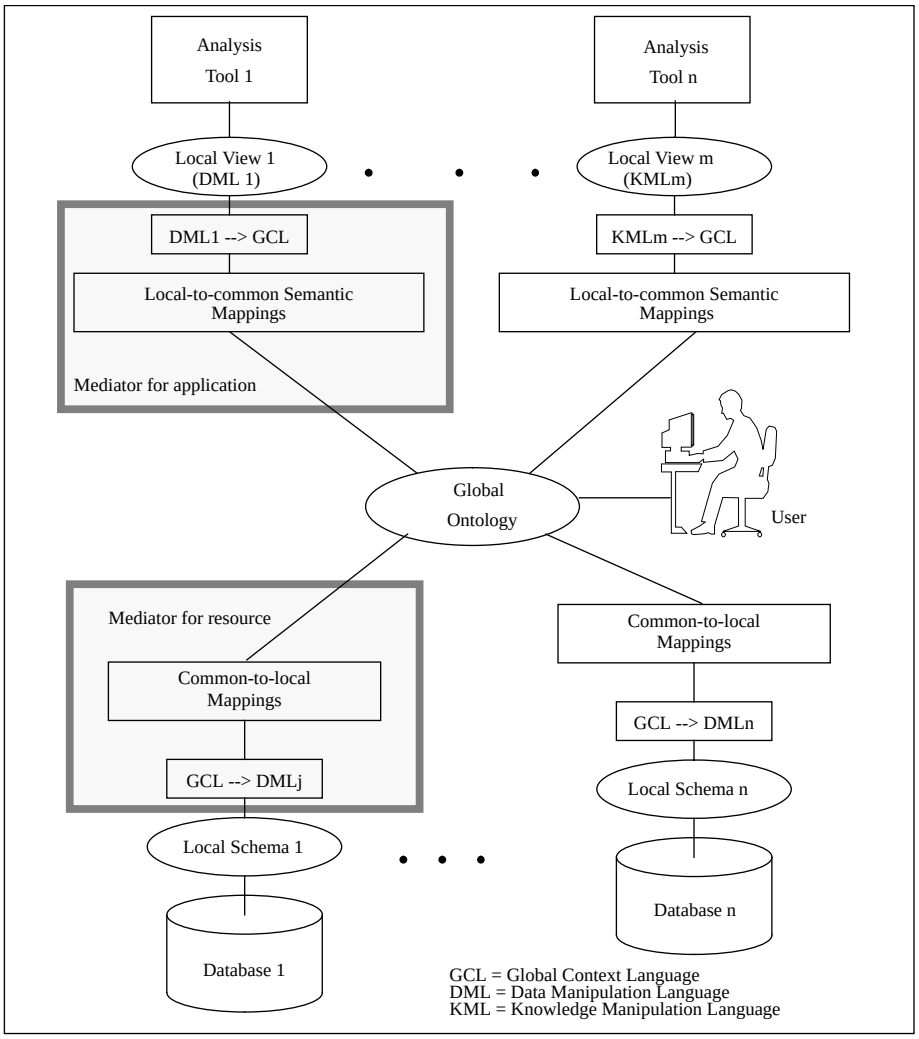

Fig. 2. Mediators

architecture [Elmasri & Navathe, 1994, ch. 1], which sought to separate and shield applications from the storage of data.

A promising approach is to use *mediators* [Wiederhold, 1992]. A mediator is a simplified agent that acts on behalf of a set of information resources or applications. Figure 2 shows a mediator architecture. The basic idea is that the mediator is responsible for mapping the resources or applications to the rest of the world. Mediators thus shield the different components of the system from each other. To construct mediators effectively requires some common represen-

tation of the meanings of the resources and applications they connect. Such a knowledge representation is called an *ontology* [Neches *et al.*, 1991]. The main learning challenges associated with ontologies include relationship and concept acquisition. Since these are related to traditional machine learning problems, some progress has already been made on them, but they are far from solved from the perspective of CIS applications.

3.1 Relationship Acquisition

The major problem with ontology-based approaches is the effort required to build them, and to relate different resources and applications to them. In order to extend their world model, agents need to be able to acquire and integrate ontologies autonomously. Agents also should learn the ontologies of other agents. In other cases, tools that assist a human designer are needed. These tools must have a strong machine learning component, to be able to not only relate concepts across databases, but also help identify relationships within an ontology. Such relationships, e.g., generalization or containment, are necessary for CIS query processing approaches, e.g., [Arens *et al.*, 1993; Huhns *et al.*, 1994]. For example, the concept *port* is a generalization of *airport* and can be used to answer queries about airports only if additional restrictions are added.

Further, different domains often have a rich variety of relationships that compose elegantly with each other [Huhns & Stephens, 1989]. To give a simple, albeit somewhat contrived, example, if a person owns a car, and the car contains a wheel, then the person also owns the wheel. These relationships form part of the common sense knowledge that is essential in relating information from different databases: two tables *car-ownership* and *car-parts* in one database may correspond to a single table *auto-part-ownership* in another database.

3.2 Concept Acquisition

We assumed in the above that the concepts that a given database is about are known. A more basic challenge is to identify those concepts. This is potentially useful, but extremely difficult, when dealing with a previously unknown source. It remains useful, and becomes more tractable, when the structure of the database is known, but the structure does not faithfully reflect the meaning of the content. A problem that arises in legacy databases is that they are often misused! For example, databases in the telecommunications industry store information about signal channels. When fiber optic technology was introduced, the databases were not redesigned to capture the new kind of channel. Rather, the existing fields in the databases were overloaded. Consequently, the *conductivity* field may reflect either the conductivity for a copper channel, or the bandwidth for a fiber channel! To access these databases systematically requires knowing what concepts they store, but the concepts are hidden inside the data values. A challenge is to discover the rules for partitioning the mixed up concepts into the correct categories. Some progress is already being made, mostly under the rubric of *data mining*, e.g., [Fayyad, 1996; Shen & Leng, 1996;

Zhang *et al.*, 1996]. An issue that has not drawn much attention is collaborative learning of the concepts. This can be important, because different uses of the data might treat the implicit concepts differently.

4 Coordination and Collaboration

Learning about active components, such as workflows and agents, and their interactions.

4.1 Workflow Acquisition

CISs not only involve retrieving information, but also updating it. Updates are qualitatively more complex than retrievals, because they can potentially introduce inconsistencies. This is especially the case when several databases are involved, and there are subtle interdependencies among them. A *workflow* is a composite activity that accesses different resources and has human interaction to solve some business need.

Traditional databases support so-called ACID transactions, which are computations that are atomic, consistency-preserving, isolated, and durable [Gray & Reuter, 1993]. In other words, a transaction happens entirely or none at all, does not violate consistency, does not expose any partial results, and if successful has permanent results. Transactions are effective in homogeneous and centralized databases, but do not apply in distributed and heterogeneous settings. This is because to ensure the ACID properties requires the component databases to expose their internal control states, and requires locking data items on a database even when those are not in use any more.

This has led to a number of extended transaction models [Bukhres & Elmagarmid, 1996; Elmagarmid, 1992]. Transaction models capture some of the aspects of workflows. Figure 3 gives a trip-planning workflow in the notation of [Buchmann *et al.*, 1992]. This workflow has a number of separate activities, such as opening an account, reserving a flight, booking a hotel, renting a car, and generating a bill. These execute on databases belonging to autonomous organizations, such as different airlines or hotels. Since the airlines make reservations independently of each other and of hotel bookings, the travel agency has to provide the control to make sure air tickets are not bought unnecessarily. Typically, a human would carry out the steps described in Figure 3. Approaches such as [Buchmann *et al.*, 1992] provide a way of representing the dependencies among the steps, and executing them appropriately. However, a major challenge is in determining the structure of workflows, possibly by observing how humans carry them out.

Because there are a large variety of extended transaction models, some so-called "RISC" approaches have been proposed that provide a small set of primitives with which to encode the behavior of different transaction models, e.g.,

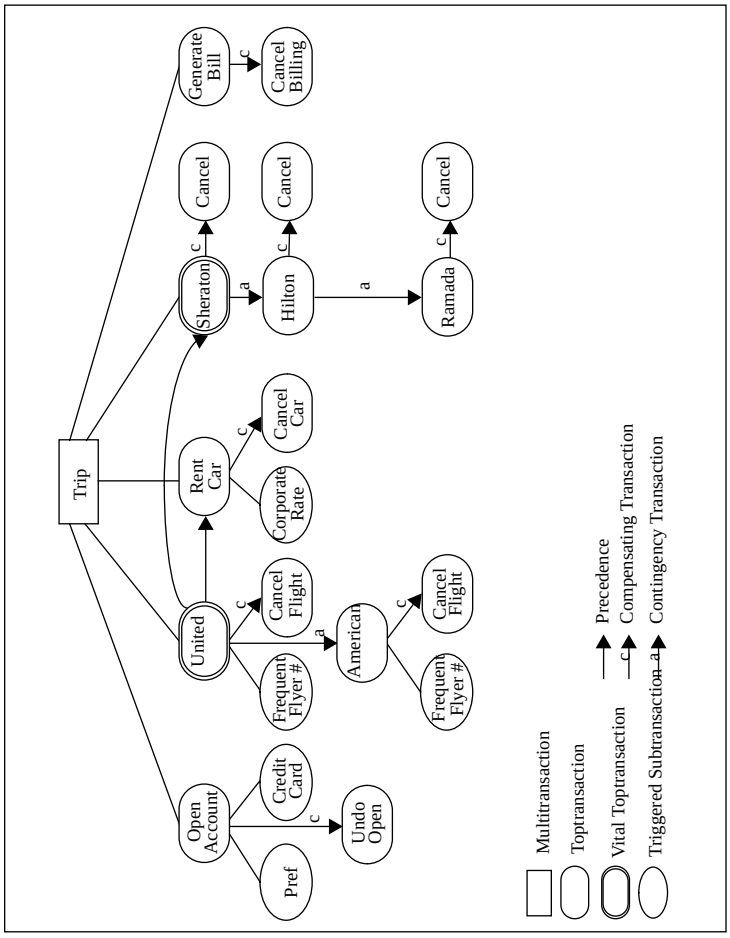

Fig. 3. Workflow for Trip Planning

[Attie *et al.*, 1993; Chrysanthis & Ramamritham, 1994; Singh, 1996]. These approaches provide some variant of a temporal language in which the coordination requirements of the transaction models can be expressed. The approaches of [Attie *et al.*, 1993; Singh, 1996] automatically produce schedules from those specifications. Our challenge can then be framed in terms of how the formal specifications are produced. We believe that the RISC approaches will facilitate learning, because they are declarative and offer a small set of primitives.

4.2 Collaboration Acquisition

Because they are large scale and open, CISs typically involve more than one workflow. However, since these workflows execute on the same resources, they

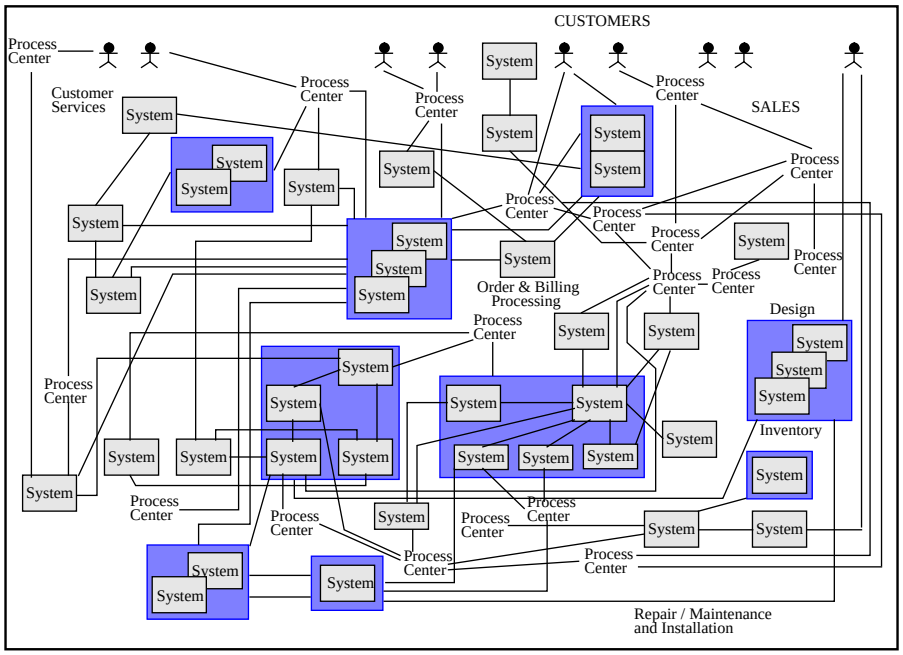

Fig. 4. The Workflow Coordination Required

have a number of interactions. Some of these interactions can be pernicious in that one workflow may cause the failure of another workflow. Some of the interactions, however, are useful. The challenge is to identify the (potential) interactions and to control them appropriately. Figure 4 schematically shows a typical situation in which resources are represented as boxes, and different workflows are sets of lines connecting them. Although the names of the systems have been removed to protect proprietary information, the picture represents the information system of a telecommunication company in the US.

The agent metaphor is useful when thinking about more than one workflow. Agents can be identified not only with the passive resources on which the workflows execute, but also with the workflows themselves—the agents can then correspond either to the humans carrying out a given workflow, or the customer of a workflow. These agents must coordinate their efforts appropriately. For example, in a telecommunications setting, a channel assignment workflow must wait until enough channels have been created by another workflow.

The challenge is to learn the potential ordering constraints of the workflows. More generally, the challenge is to infer the activities or plans of other agents, and learn from repeated interactions with them. Related challenges arise when the information environment is truly open and new agents are added dynamically, or the agents involved do not repeat interactions. In such cases, an agent still needs to learn how to collaborate with classes of agents, and to classify them

appropriately. For example, an agent may infer that agents who request a price quote for valves will often also want a price quote on matching hoses.

The foregoing challenges can be generalized still further to learn about the agents' dispositions to one another. For example, it is important to learn to what extent other agents will cooperate with the given agent. Indeed, if the agents form a team or coalition, they will be able to assist each other and prevent mishaps [Shehory & Kraus, 1996]. It is also useful to have models of the learning abilities of the other agents. A number of studies have shown that coalitions become more effective as the members of the coalition learn about each other. An implication of this is that the team members should act predictably and transparently (possibly by revealing their state) in order to abet the learning. Interestingly, this implication has not yet been researched or incorporated in any systems.

5 Abstractions and Structure

Learning about interactive components, such as roles and organizational structure, their dispositions, responsibilities, and commitments.

For CISs applied to enterprises or virtual enterprises, a variety of models are typically built. Figure 5 shows some of the common modeling approaches. Of the main ones, entity-relationship (E-R) diagrams describe a conceptual model of the information stored in (a subset of the databases in) the enterprise. Activity decomposition describes the relationship of inclusion among different activities, whereas the control, data, and materiel flows give additional information about it. E-R diagrams correspond to static information as in ontologies; the activity representations correspond to the workflows. It is important to relate the two categories of representations, because the actions in the workflows depend on the concepts they manipulate, and the concepts are defined based on their patterns of usage. A challenge is to classify the concepts and actions in this manner, so that they can be used for building ontologies and coordinating workflows.

In a number of settings, including enterprises, the organizational structure of a CIS is important. By the organizational structure, we mean the set of roles and responsibilities that make up a functioning system [Gasser, 1991; Papazoglou *et al.*, 1992]. There is an intimate relationship between the workflows executing in a CIS, and the organizational roles available in it. Figure 6 shows on the left a simple workflow corresponding to submitting a contract proposal from a company. The *write white paper* task itself may be decomposed into a subworkflow. The bottom left shows a possible subworkflow for travel. The tasks in the workflow impinge upon various databases, and other ongoing processes, such as *budget forecast*. They also relate to the organizational structure of the company, because key steps in the workflow must be performed by people with specific authorities.

Traditionally, the roles are mapped to tasks rigidly. However, in open and dynamic environments, more flexible role-bindings are needed. For example, if the *research director* is on leave, how may the workflow be rerouted? If one

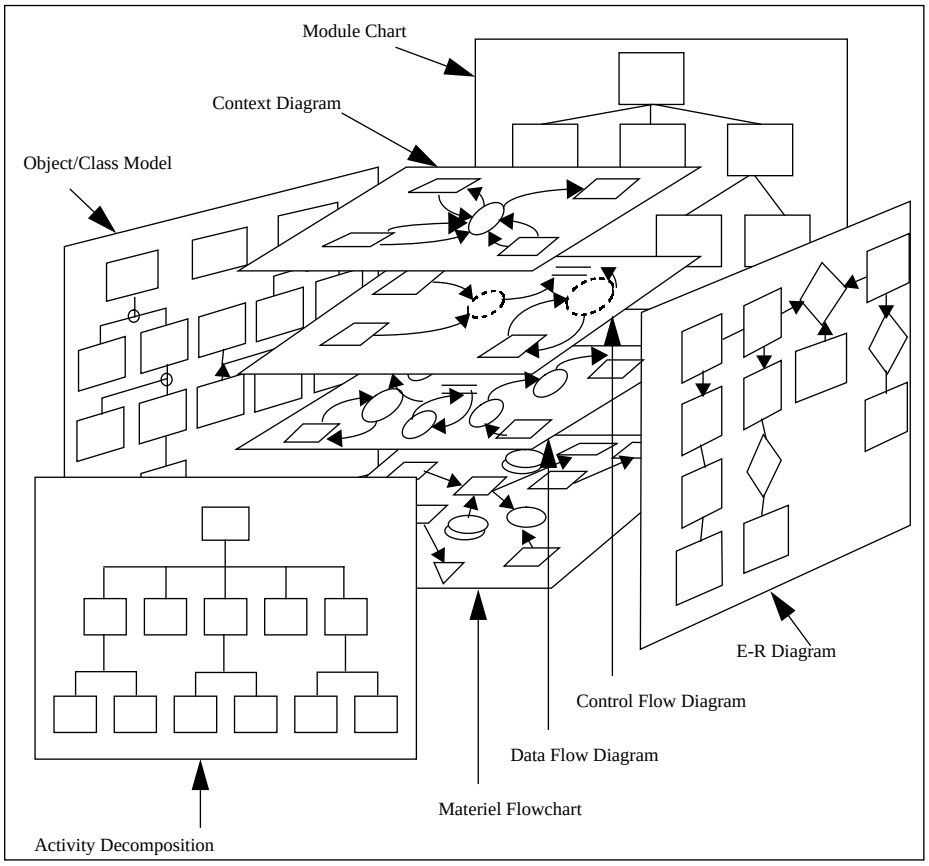

Fig. 5. Different Views of an Enterprise

person fills multiple roles, how may the workflow be scheduled to optimize their time? The challenge is to learn the capabilities and authorities necessary to execute different steps in a workflow, and to learn the interrelationships among the various roles.

6 Conclusions

Learning by agents can minimize or entirely replace communications, which is extremely important in large information environments where communication bandwidth is an expensive commodity.

In looking at the examples of CISs, we find that certain problems show up in different guises. The unifying themes of CIS are the following. One, we wish to obtain the effect of logical homogeneity and centralization despite physical distribution and heterogeneity. Two, we wish to support the logical openness of

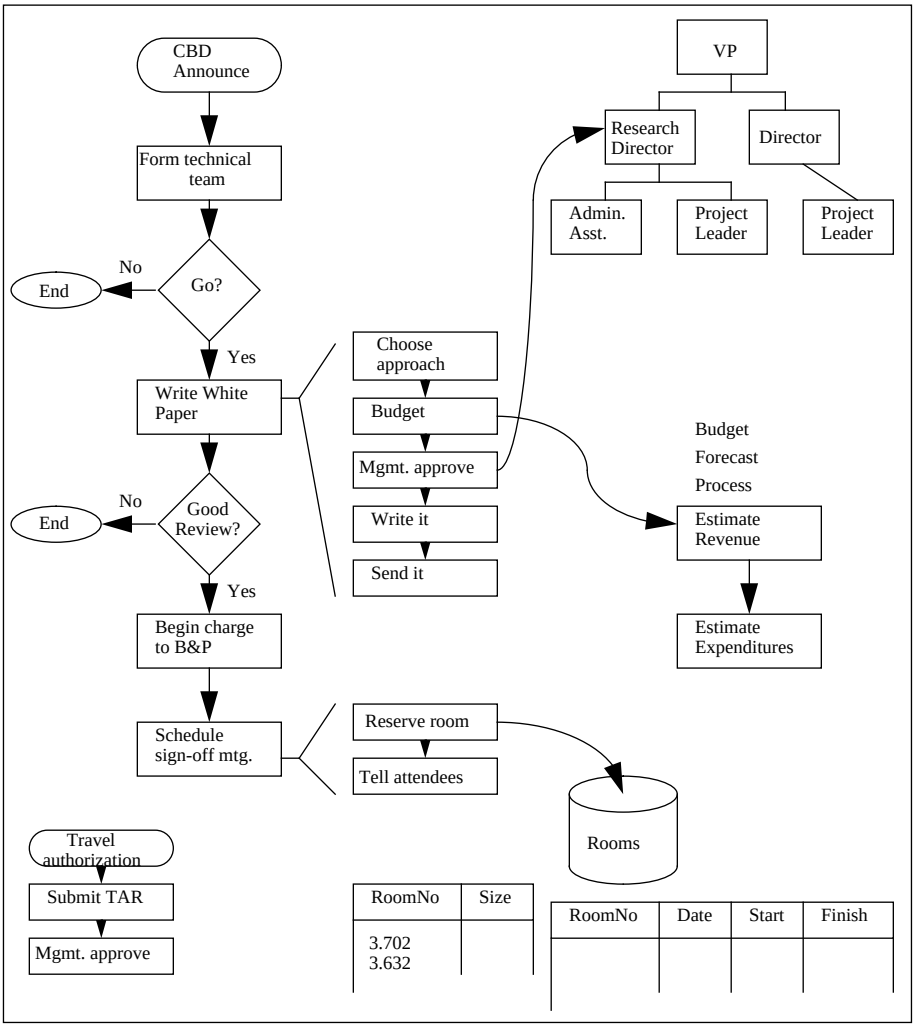

Fig. 6. Relating Views to Obtain Flexible Behavior

CISs. Openness translates into a number of interesting systemic challenges, relating to how a CIS may initialize and stabilize when some agents come together, are added, or leave. These lead to the following challenges for machine learning:

- learning about each other
- learning about society and the environment
- learning from repeat interactions with changing agent instances
- learning biased by social structure
- forgetting by a group about its former members.

A number of learning techniques exist [Russell & Norvig, 1995]. We give some

suggestions about how different categories of learning might relate to problems in CIS. These categories are, of course, not mutually exclusive:

- Clustering techniques can help extract concepts from vast amounts of data (e.g., by classifying data that was carelessly mixed up)
- Passive learning appears appropriate for a new agent that joins a group (e.g., watching)
- Active learning can help a group learn about its new members, (e.g., interviewing them to evaluate their opinions)
- Unsupervised learning can be an unintrusive approach for acquiring workflows and learning the constraints on role-bindings (e.g., looking over the shoulder of staff members performing different tasks)
- Supervised learning applies for relating the more subtle interactions among workflows (e.g., being told business rules)
- Reinforcement learning applies in environments with autonomously built agents (e.g., adaptively acting and interacting).

This paper described some of the key ideas in CIS, and pointed out some of the places where machine learning could contribute. We believe the relationship between the two areas is synergistic. Cooperative information systems need machine learning to realize their promise of adaptivity and flexibility. Machine learning can benefit from CISs as a rich application area with open problems that are widely recognized as crucial, and promise to yield significant scientific advances in machine learning.

References

[Arens *et al.*, 1993] Arens, Yigal; Chee, Chin Y.; Hsu, Chun-Nan; and Knoblock, Craig A.; 1993. Retrieving and integrating data from multiple information sources. *International Journal of Intelligent and Cooperative Information Systems* 2(3):127–158.

[Attie *et al.*, 1993] Attie, Paul C.; Singh, Munindar P.; Sheth, Amit P.; and Rusinkiewicz, Marek; 1993. Specifying and enforcing intertask dependencies. In *Proceedings of the 19th VLDB Conference*. 134–145.

[Buchmann *et al.*, 1992] Buchmann, Alejandro; Özsu, M. Tamer; Hornick, Mark; Georgakopoulos, Dimitrios; and Manola, Frank A.; 1992. A transaction model for active distributed object systems. In *[Elmagarmid, 1992]*. Chapter 5, 123–158.

[Bukhres & Elmagarmid, 1996] Bukhres, Omran A. and Elmagarmid, Ahmed K., editors. *Object-Oriented Multidatabase Systems: A Solution for Advanced Applications*. Prentice Hall.

[Chrysanthis & Ramamritham, 1994] Chrysanthis, Panos K. and Ramamritham, Krithi; 1994. Synthesis of extended transaction models using ACTA. *ACM Transactions on Database Systems* 19(3):450–491.

[Elmagarmid, 1992] Elmagarmid, Ahmed K., editor. *Database Transaction Models for Advanced Applications*. Morgan Kaufmann.

[Elmasri & Navathe, 1994] Elmasri, Ramez and Navathe, Shamkant; 1994. *Fundamental of Database Systems*. Benjamin Cummings, Redwood City, California, second edition.

[Fayyad, 1996] Fayyad, Usama, editor. *Special Issue on Data Mining*, volume 39(11) of *Communications of the ACM*.

[Gasser, 1991] Gasser, Les; 1991. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence* 47:107–138.

[Gray & Reuter, 1993] Gray, Jim and Reuter, Andreas; 1993. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann.

[Huhns & Stephens, 1989] Huhns, Michael N. and Stephens, Larry M.; 1989. Plausible inferencing using extended composition. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1420–1425.

[Huhns *et al.*, 1994] Huhns, Michael N.; Singh, Munindar P.; Ksiezyk, Tomasz; and Jacobs, Nigel; 1994. Global information management via local autonomous agents. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*.

[Neches *et al.*, 1991] Neches, Robert; Fikes, Richard; Finin, Tim; Gruber, Tom; Patil, Ramesh; Senator, Ted; and Swartout, William R.; 1991. Enabling technology for knowledge sharing. *AI Magazine* 12(3):36–56.

[Papazoglou *et al.*, 1992] Papazoglou, Mike P.; Laufmann, Steven C.; and Sellis, Timothy K.; 1992. An organizational framework for cooperating intelligent information systems. *International Journal on Intelligent and Cooperative Information Systems* 1(1):169–202.

[Russell & Norvig, 1995] Russell, Stuart J. and Norvig, Peter; 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

[Shehory & Kraus, 1996] Shehory, Onn and Kraus, Sarit; 1996. Formation of overlapping coalitions for precedence-ordered task execution among autonomous agents. In *Proceedings of the International Conference on Multiagent Systems*. 330–337.

[Shen & Leng, 1996] Shen, Wei-Min and Leng, Bing; 1996. A metapattern-based automated discovery loop for integrated data mining—unsupervised learning of relational patterns. *IEEE Transactions on Knowledge and Data Engineering* 8(6):898–910.

[Singh, 1996] Singh, Munindar P.; 1996. Synthesizing distributed constrained events from transactional workflow specifications. In *Proceedings of the 12th International Conference on Data Engineering (ICDE)*.

[Wiederhold, 1992] Wiederhold, Gio; 1992. Mediators in the architecture of future information systems. *IEEE Computer* 25(3):38–49.

[Zhang *et al.*, 1996] Zhang, T.; Ramakrishnan, R.; and Livny, M.; 1996. BIRCH: An efficient data clustering mehtod for very large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*.