MeRC

McMaster eBusiness Research Centre

A REVIEW OF METHODOLOGIES FOR THE DEVELOPMENT OF INTELLIGENT AGENTS

By

Xiaoqing Li and Ali R. Montazemi

Li.Xiaoqing@uis.edu
montazem@mcmaster.ca

McMaster eBusiness Research Centre (MeRC)
Michael G. DeGroote School of Business

MeRC Working Paper No. 2

May 2003

McMaster University

A REVIEW OF METHODOLOGIES
FOR THE DEVELOPMENT OF INTELLIGENT AGENTS

By
Xiaoqing Li and Ali R. Montazemi

McMaster eBusiness Research Centre (MeRC)
Michael G. DeGroote School of Business
McMaster University
Hamilton, Ontario, L8S 4M4
Canada
Li.Xiaoqing@uis.edu
montazem@mcmaster.ca

This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada Discovery Grant.

Abstract

With the development of computer technology, and especially with the advance of the Internet, information systems are becoming the vehicle for an increasing range of everyday activities. Through the computer, people can trade stocks, check email, chat with friends, play games, etc. More and more untrained people have become computer users. Intelligent agent, a technique from the field of artificial intelligence (AI), is expected to assist end users cope with increasing information overload. An intelligent agent can reduce the complexity of dialogue by understanding the goals of the user and assisting him/her to interact with the system. This paper provides a review of the methodologies used in developing intelligent agent systems.

1. Introduction

Agent-based technologies are considered the most promising means to deploy enterprise-wide and worldwide applications that often must operate across corporations and continents and inter-operate with other heterogeneous systems (Bellifemine et al., 2001). We can identify applications of agent technology in diverse areas such as information retrieval systems to help users to retrieve relevant documents (Shaw et al., 2002); and electronic commerce to help buy and sell (Turowski, 2002). More importantly, with the pricking of the Internet bubble, online retailers are under more pressure than ever to earn their keep, and as a result, many companies are looking at intelligent agents as one of the sophisticated merchandising tools that can recommend products and build customer loyalty and sales (Kwak, 2001).

Generally speaking, an intelligent agent is particularly useful in open and complex systems such as the Internet (Jennings and Wooldridge, 1998). In open systems, the system structure is capable of dynamic change. The availability, type, and reliability of information services are also constantly changing. Information can be ambiguous and possibly erroneous due to the dynamic nature of the information sources, and potential information updating and maintenance problems (Sycara et al., 1996). Therefore, the huge amount of information poses challenges to decision-makers because of the accompanying difficulty in collecting, filtering, evaluating and using it. For example, many institutions (e.g., www.etrade.com) enable investors to purchase common stocks online from their Internet site. However, the onus is on the customer to have perfect knowledge of thousands of common stocks traded in different exchange centers (e.g., The

New York Stock Exchange). This renders the online information market somewhat inefficient and sets the stage for the emergence of information "intermediaries" in the market. We can use software agents to act as "intermediaries" in support of customer requirements.

In complex systems, the most powerful tools for handling complexity in software development are modularity and abstraction. Agents are powerful tools for making systems modular. With a multi-agent system, a designer can partition a complex task into several small and relatively independent subtasks. Each agent then performs a specific subtask. For example, several agents work collaboratively to perform portfolio analysis for the stock trader in the WARREN system (Decker et al., 1998).

Successful application of intelligent agents in support of decision-making processes is contingent on two critical phases. First, one needs to identify the decision-making processes that can best be supported by the agent methodology. The second phase requires the appropriate use of technology in the development of pertinent agent systems. This paper provides an overview of intelligent agent research-and-development environments. Section 2 elaborates on the characteristics of agents. There are almost as many opinions on the definition of agents as there are agents themselves. The diversity of agent definition can be attributed to the range of applications that can use this technology to enhance decision-making processes. In this paper, agent and intelligent agent refer to the same type of application system and are used interchangeably. Agents have to interact with each other as well as with environmental entities (e.g., human decision-makers and databases) to achieve their goals. Three basic agent architectures are described in section 3. Section 4 describes an agent communication language called Knowledge Query and Manipulation Language (KQML). One of the basic problems facing designers of multi-agent systems for open and complex information environments such as the Internet is that of connection: finding the other agents who might have the required information to deal with a decision problem. To this end, section 5 describes the architecture of multi-agent systems and pertinent coordination strategies. In section 6, we discuss the current architecture and existing limitations. Section 7 provides concluding remarks and examines the challenges inherent in the development of agent-based systems in support of decision-making processes.

2. Decision-making and Intelligent Agents

Decision-making is a process of choosing among alternative courses of action for the purpose of attaining a goal or goals (Turban and Aronson, 1998). According to Simon (1977), there are three major phases involved in the decision making process: intelligence, design, and choice. The decision-making process starts with the intelligence phase, where reality is examined and problem is identified and defined. In the design phase, a model that represents the system is constructed. This is done by making assumptions that simplifies reality and by writing down the relationship among all variables. The choice phase includes selection of a proposed solution to the model (not to the problem it represents). Once the proposed solution seems to be reasonable, we are ready for the last phase: implementation. Successful implementation results in solving the real problem.

Various kinds of technologies are developed to support each phase of decision-making process. Decision support systems (DSS) allow people at many different levels to systematically analyze problems before making a decision. In the process, these systems extend the range and capability of the decision-making process, increasing its effectiveness (Gallegos, 1999). Especially, the intelligence phase is a primary target for DSS and for other computer-based information systems that deal with nonstructured problems (Lucas, 1995). The primary requirement of decision support for the intelligence phase is the ability to scan external and internal information sources for opportunities and problems and to interpret what the scanning discovers. Nowadays, the worldwide marketplace provides not only more customers, suppliers and competitors, but also increased complexity for the decision-making process (Sauter, 1999). Internet-based electronic transactions take place actively worldwide and the transaction amount is continuously increasing day by day (Kang and Han, 2002). As an excellent information source, the Internet provides significant opportunities for people to obtain information. Electronic information services are pitched to a wider range of decision-'makers, from CEOs and CIOs on down to the end users themselves (Curle, 1998). At the same time, the Internet also brings about the problem of information overload (Chen et. al, 2002).

Information overload results from the inability of living systems to process excessive amounts of information. The cognitive limitations of humans make it impractical to consider all possible alternatives to a particular problem. Even if we could review all relevant alternatives, we would not be able to assimilate all the information so that we could make an appropriate decision (Marakas, 2003). As the complexity of the task or information load increases, the human information processor tries to reduce cognitive effort by changing to a more effective information-processing strategy. People try to minimize the effects of information overload by employing conscious or even unconscious strategies to reduce information load (Grise and Gallupe, 1999). We tend to "simplify reality" by focusing our energy on finding a solution that meets our preconceived notion of what an acceptable solution looks like. Upon finding such a solution, we immediately adopt it and stop to looking for a better one (Marakas, 2003).

On the Internet, the staggering amount of information has made it extremely difficult for users to locate and retrieve information that is actually relevant to their task at hand. Given the bewildering array of resources being generated and posted on the WWW, the task of finding exactly what a user wants is rather daunting. Although many search engines currently exist to assist in information retrieval, much of the burden of searching is on the end-user. A typical search results in millions of hits, many of which are outdated, irrelevant, or duplicated (Ram, 2001). One promising approach to managing the information overload problem is to use "intelligent agents" for search and retrieval (Ram, 2001). Agents will interpret user requests and automate manual processes. Agents will allow users to delegate simple tasks. Users will have time to solve complex, abstract problems, while agents use their knowledge of user preferences, standard domain defaults, and networked information sources to make simple decisions and even take action on behalf of the user (Dyer, 1999). For example, an agent might remind or automatically prompt a person to email Joe, find an article on IBM's new chip, or buy Yahoo stock when it drops to 80. In a more technical vein, agents are atomic software entities operating through autonomous actions on behalf of the user without constant human intervention (Ma, 1999).

3. Characteristics of Intelligent Agents

There is currently no general consensus on the definition of an agent (Serugendo, 2001). Different researchers have given different definitions based on their practices and understandings. Here, we will discuss agents mainly from a practical view and investigate the major characteristics of agents.

Intelligent agents work in open and complex information environments (Jennings and Wooldridge, 1998). In complex systems, the most powerful tools for handling complexity in software development are modularity and abstraction. The agent paradigm and multi-agent systems (MAS) are widely recognized as suitable abstractions to deal with complex application environments, especially when the openness and unpredictable dynamics of the environment make traditional approaches less effective (Ricci et al., 2001). In such an information environment, the structure of the system itself is capable of dynamically changing. In order to achieve the goal of the user, the agent performs the following actions (Reticular, 1999): executes autonomously; communicates with other agents or the user; and monitors the state of its execution environment. Its components are not known in advance, can change over time, and may be highly heterogeneous. To be an intelligent agent, Newell argues that software should possess the following capabilities or attributes (Newell, 1988):

- Be able to exploit significant amounts of domain knowledge.

- Be tolerant of errorful, unexpected, or wrong input.
- Be able to use symbols and abstractions.
- Be capable of adaptive, goal-oriented behavior.
- Be able to learn from the environment.
- Be capable of operation in real-time.
- Be able to communicate using natural language.

In fact, not all of the above features are needed for all intelligent agents. Hayes-Roth (1995) views intelligent agents as having the capability to perform three necessary functions:

- To perceive dynamic conditions in the environment.
- To take action to affect conditions in the environment.
- To reason in order to interpret perceptions, solve problems, draw inferences, and determine actions.

Researchers have described the characteristics of, and classified agents in, numerous ways. Nwana (1996) provides a typology that defines four types of agents based on their abilities to cooperate, learn, and act autonomously. Autonomy refers to the principle that agents can operate on their own without the need for human guidance. With cooperation capability, agents can interact with each other and possibly humans via some communication language and coordinate their actions without cooperation. The key attribute of any intelligent being is its ability to learn. Smart agent systems would have to *learn* as they react and/or interact with their external environment. Nwana (1996) terms these smart agents, collaborative agents, collaborative learning agents, and interface agents (Nwana, 1996). Figure 1 depicts how these four types of agents utilize the capabilities described next.

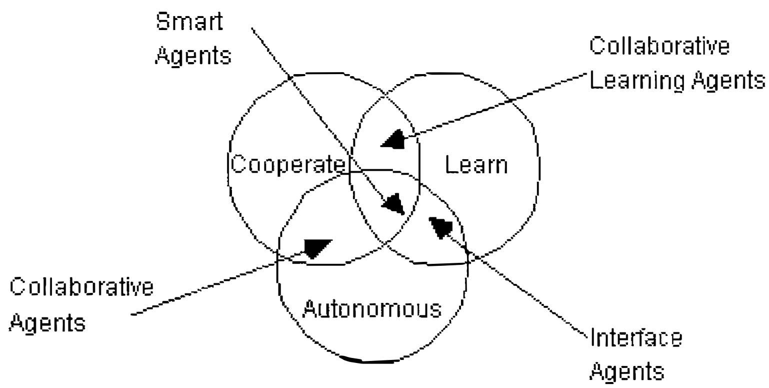

Figure 1: A Typology of Agents (Nwana, 1996)

- **Collaborative Agents**

To deal with complex real world problems, it is desirable to have different types of agents specializing in different types of tasks to collaborate with others to solve a problem (Ram, 2001). Collaborative agents emphasize autonomy and cooperation to perform tasks by communicating, and possibly negotiating, with other agents to reach mutual agreements. For example, the Collaborative Agent Interaction and synchronization (CAIRO) (PenÄa-Mora et. al, 2000) system provides an environment for structured information exchange across the Internet in real-time. The major component of the CAIRO system is the set of distributed artificial intelligence based software agents. The agent removes some level of direct involvement in running a meeting. The CAIRO system allows designers and engineers to work together in virtual teams by supporting multi-media interactions over computer networks. CAIRO aids the concurrent engineering effort by relaxing the physical, temporal and organizational constraints experienced in traditional design meeting environments.

- **Interface Agents**

Interface agents are autonomous and utilize learning to perform tasks for their users. This class of agent is used to implement assistants as well as guides, memory aids, and filters; perform matchmaking and referrals; or buy and sell on behalf of the user (Reticular, 1999; Conway and Koehler, 2000). For example, Letizia (Lieberman, 1997) is a user interface agent that assists a user browsing the World Wide Web. As the user operates a conventional Web browser such as Netscape, the agent tracks user behavior and attempts to anticipate items of interest by doing concurrent, autonomous exploration of links from the user's current position.

- **Collaborative Learning Agents**

A typical example of Collaborative Learning Agents is a robotic soccer system by Stone and Veloso (1998). In this collaborative system, teams of agent players must work together to put the ball in the opposing goal while at the same time defending their own. Learning in this system is divided into two levels. First the agent players learn to acquire some low-level skills that allow them to manipulate the ball. Second, they must learn to work together to achieve the common goal of winning.

Besides the above classification, researchers also classfied agents based on other dimensions such as their mobility (i.e., by their ability to move around telecommunication networks. This yields the classes of static or mobile agents). Mobile agents are computational processes capable of moving over a network (e.g., a wide area network such as the Internet or World Wide Web); interacting with foreign hosts; gathering information on behalf of the user; and returning to the user after performing their assigned duties. Mobile agents are increasingly used in various Internet-based applications such as electronic commerce, network management, and information retrieval (Kim et al., 2001). For example, TabiCan (www.tabican.ne.jp), one application of IBM's Aglet (www.trl.ibm.co.jp/aglets), offers several merchant agents for companies selling tickets online. When a user accesses TabiCan, a consumer agent is created and interacts with the merchant agent to find travel information.

The most popular uses for intelligent agents are finding, analyzing and retrieving large amounts of information (Krishnan et al., 2001; Rhodes and Maes, 2000; Tu and Hsiang, 2000). Information agents are tools to help manage the tremendous amount of information available through communication networks. Information agents access the network looking for particular kinds of information, filter it, and return it to their users. For example, WARREN (Decker et el., 1996) has six information agents: two stock ticker agents using different WWW sources; a news agent for Clarinet and Dow-Jones news articles; and an agent that can extract current and historical sales and earnings-per-share data from the SEC Edgar electronic annual reports.

4. Agent Architecture

Agent architectures are essentially design methodologies: they are technological frameworks and scaffolding for developing agents (Bryson and Stein, 2001). The architecture of an agent describes its modules and capabilities. Usually three types of architectures are distinguished according to the agent paradigm as follows (Müller, 1997; Botti et al., 1999):

- Reactive agents

  Agents that are built according to the behavior-based paradigm, that have no or only a very simple internal representation of the world, and that provide a tight coupling of perception and action.

- Deliberative agents or belief-desire-intention (BDI) systems

  Agents in the symbolic artificial intelligence tradition that have a symbolic representation of the world in terms of categories such as beliefs, goals, or intentions, and that possess logical inference mechanisms to make decisions based on their world model.

- Hybrid agents

  Agents that are built from two or more subsystems. One is deliberative (i.e., containing a symbolic world model) and the other is reactive.

Each BDI agent has a sophisticated reasoning architecture that consists of different modules that operate asynchronously. Reactive agents do not have representations of their environment and act using a stimulus-response type of behavior; they respond to the present state of the environment in which they are situated. Reactive systems are mainly used in rapidly changing environments. Nonetheless hybrid agent systems can be used for most application problems since neither a purely deliberate nor purely reactive architecture is appropriate (Sycara, 1998).

5. Agent Communication Language

The central idea underlying software agents is that of delegation. To delegate is to entrust a representative to act on one's behalf (Norman and Reed, 2001). The user delegates a task to the agent and the agent autonomously performs that task on behalf of the user. For delegation to be successful, there must be a relationship between the agent delegating the goal or task and the agent to whom it is delegated (Norman and Reed, 2001). The act of delegating can be carried out by the performance of communication. In multiagent systems, if agents are not designed with an embedded knowledge about the beliefs, intentions, abilities and perspectives of other agents, they need to exchange information to improve their social activities (Dragoni et al., 2001). Knowledge Query and Manipulation language (KQML) is designed to support interactions among intelligent software agents. KQML offers an abstraction of an information agent (provider or consumer) at a higher level than is typical in other areas of computer science. KQML assumes a model of an agent as a knowledge-based system (KBS) (Finin et al., 1994b). The KBS model easily subsumes a broad range of commonly used information agent models, including database management and hypertext systems, server-oriented software (e.g., finger demons, mail servers, HTML servers), and simulations. Figure 2 summarizes the possible components of an agent; they are grouped into representation components, communication components, and components that are not directly related to shared understanding (Finin et al., 1997).

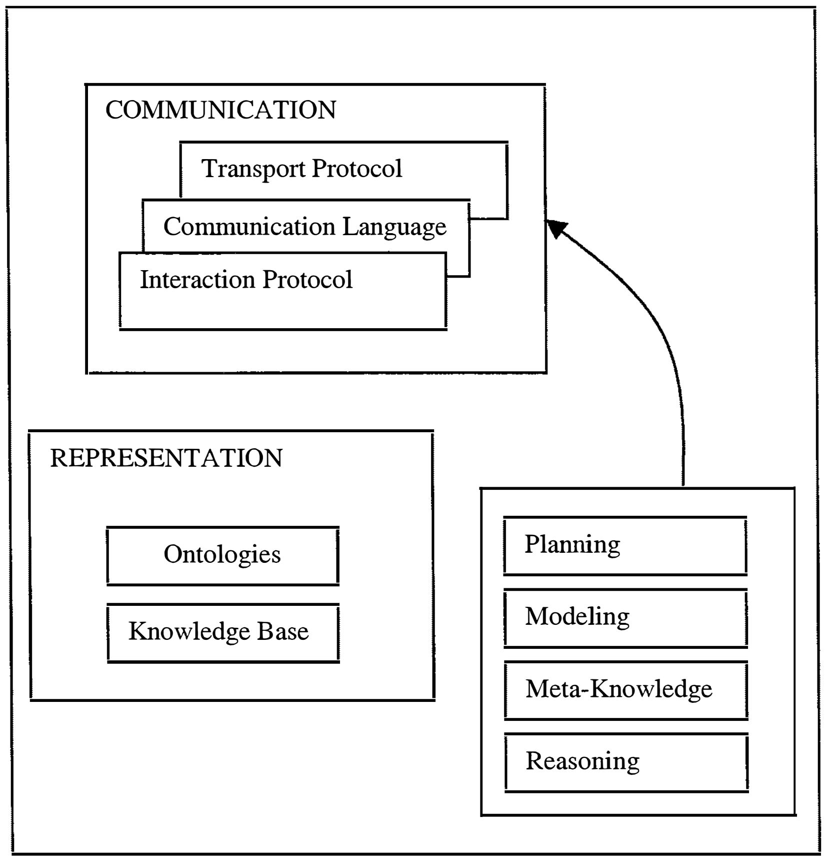

Figure 2: An abstract model for interoperating software agents with three classes of components: representation components, communication components, and components not directly related to shared understanding. (Finin et al., 1997)

KQML is most useful for communication among agent-based programs, in the sense that the programs are autonomous and asynchronous. Autonomy means that agents may have different, and even conflicting, agendas. Thus, the meaning of a KQML message is defined in terms of constraints on the message sender rather than on the message receiver. This allows the message receiver to choose a course of action that is compatible with other aspects of its function.

KQML language can be viewed as being divided into three layers: the content layer, the message layer and the communication layer. See Figure 3.

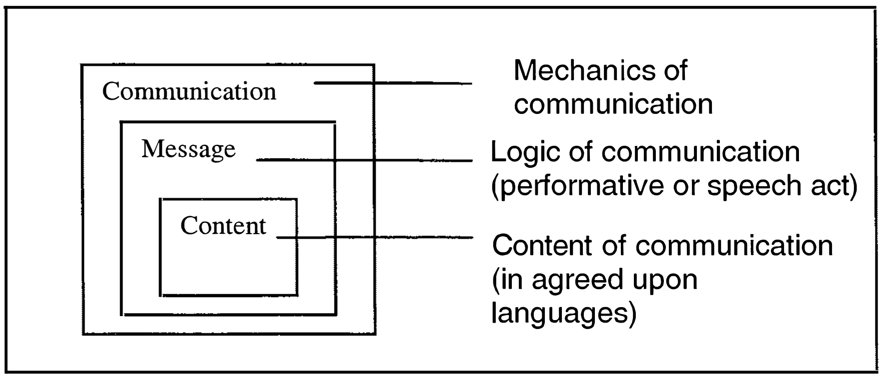

Figure 3: The three layers of the KQML communication language (Reticular, 1999)

- The content layer is the actual content of the message in the program's own representation language. KQML can carry any representation language, including languages expressed as ASCII strings and those expressed using a binary notation. All of the KQML implementations ignore the content portion of a message except to the extent that they need to determine its boundaries.
- The communication layer encodes a set of features to the message which describe the lower level communication parameters, such as the identity of the sender and recipient, and a unique identifier associated with the communication.

- The message layer forms the core of the language. It determines the kinds of interactions one can have with a KQML-speaking agent. The primary function of the message layer is to identify the protocol to be used to deliver the message and to supply a speech act, or performative, which the sender attaches to the content. The performative signifies that the content is an assertion, a query, a command, or any other function in a set of known performatives. Because the content is opaque to KQML, this layer also includes optional features which describe the content: its language; the ontology it assumes; and some type of more general description, such as a descriptor naming a topic within the ontology. These features make it possible for KQML implementations to analyze, route, and properly deliver messages even though their content is inaccessible.

A KQML message consists of a performative, its associated arguments which include the real content of the message, and a set of optional arguments. The main focus of KQML is on its extensible set of performatives, which defines the permissible operations that agents may attempt on each other's knowledge and goal states at run time. The performatives comprise substrata on which to develop higher-level models of inter-agent interaction such as contract net and negotiation.

The contribution that KQML makes to Distributed AI (DAI) research is to offer a standard language and protocol that intelligent agents can use to communicate among themselves as well as with other information servers and clients. Permitting agents to use whatever content language they prefer makes KQML appropriate for most DAI research.

6. Architecture of Multi-Agent Systems

Multi-Agent systems are groups of agents that work together as a single system to integrate their functionality. They consist of a group or groups of agents that interoperate, cooperating to execute large complex tasks (Nodine et al., 2001). In the open and dynamic environment, each agent needs to collaborate with other agents. Therefore a fundamental agent requirement is the ability to coordinate its own actions with those of other agents (Durfee, 2001). Coordination entails managing dependencies between activities (Schumacher, 2001). There are many kinds of research on the coordination problem related to organizations or even virtual organizations. For example, Fernandez and Wijegunaratne (1998) studied the cooperation approach in distributed applications, Bernus and Uppington (1998) demonstrated the coordination in a virtual enterprise, and Flores et al. (2001) developed the architecture for multi-agent coordination and cooperation. But of all these multi-agent system architectures, the major basic structures are two: centralized and decentralized (Sikora and Shaw, 1998). Next, we explain these two main kinds of control structures used in multi-agent systems coordination.

6.1 Control Structures and Coordination Mechanisms

One of the basic problems facing designers of open multi-agent systems for the Internet is the connection problem (Davis and Smith, 1983)--finding the other agents who might have desired preferences and capabilities. Preference is (meta) knowledge about what types of information have utility for a requester, both in form (e.g., John follows the price of IBM) and in other characteristics (e.g., John wants only free information; John wants stock quotes at least every 35 minutes). Capability refers to (meta) knowledge about what types of requests can be serviced by a provider (e.g., Mary can provide the current price of any NASDAQ stock, delayed 15 minutes, for free, at a rate of 10 quotes per minute). There are basically two kinds of control structures in multi-agent systems (Mařík et al., 1999; Sikora and Shaw, 1998): centralized control and decentralized control. In centralized control, there is a central coordinator to whom everyone communicates solutions. The coordinator, therefore, handles the interdependencies among the agents. Usually, either they reply on some quantitative measures of utility, or their replies are based on a qualitative notion of interrelation (Ossowski, 1999). Decentralized control is the most common form of control structure in distributed systems. There is no coordinator in decentralized control. A solution to a coordination problem constitutes equilibrium, a compromise that assures somehow "maximal" attainment of the different interests of all involved individuals (Ossowski, 1999).

6.1.1 Centralized Control

In centralized control, there is a central coordinator called middle agent that handles interdependences among agents (See Figure 4) (Finin et al., 1994).

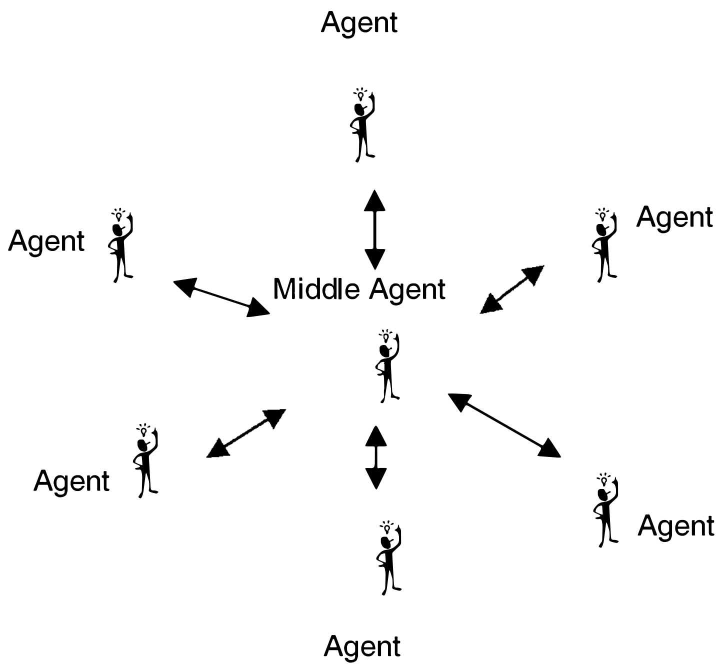

Figure 4: Agents Communicating through a Middle Agent

There are several types of middle agents, including matchmaker agents, blackboard agents, broker agents and yellow pages agents (Sycara et al., 1997).

- A broker agent protects the privacy of both preferences and capabilities, and routes both requests and replies appropriately.
- A matchmaker/yellow-pages agent stores capability advertisements that can be queried by requesters. The requesters then choose and contact directly any provider they wish.

- A blackboard agent keeps track of requests. Requesters post their problems and providers can then query the blackboard agent for events they are capable of handling.

For example, Retsina (Sycara et al., 1996) uses a distributed collection of software agents that cooperate asynchronously to perform goal-directed information retrieval and integration for supporting a variety of decision-making tasks. Each user in the Retsina framework is associated with a set of agents that collaborate to support the user in various tasks and act on his or her behalf. The agents are distributed and run across different machines. They have access to models of the task and information-gathering needs associated with different steps of the task. Based on this knowledge, the agents decide a) how to decompose and delegate tasks; b) what information is needed at each decision point, and c) when to initiate collaborative searches with other agents to get, fuse, and evaluate the information. Retsina uses three types of agents: interface, task, and information as follows.

- Interface agents interact with the user by receiving user specifications and delivering results. They acquire, model, and utilize user preferences to guide system coordination in support of the user's tasks. The main functions of an interface agent include: (1) collecting relevant information from the user to initiate a task; (2) presenting relevant information including results and explanations; (3) asking the user for additional information during problem solving; and (4) asking for user confirmation, when necessary.

- Task agents support decision making by formulating problem-solving plans and carrying them out through query and exchange of information with other software agents.
- Information agents provide access to a heterogeneous collection of information sources. These agents have models of the associated information resources, and strategies for source selection, information access, and conflict resolution and information fusion.

Agents are distributed across different machines in Retsina that use a matchmaker structure. Agents that can provide services advertise their capabilities to the matchmaker. An agent queries the matchmaker when looking to find another agent with a specific capability-- one that can supply particular information or achieve a problem-solving goal. The matchmaker either returns appropriate lists of agents matching the query description, or returns "null" if it finds no match.

6.1.2 Decentralized Control

The majority of MAS work deals with systems in which agents are peers of each other (Turner and Jennings, 2001) with a common form of decentralized control (Sikora and Shaw, 1998). See Figure 5. The agents have to interact among themselves, exchanging information and coordinating their interdependencies without the help of a middle agent. In practice, however, due to the communication costs and information overload, each agent is allowed to communicate only with a small subset of other agents

(Sikora and Shaw, 1998). The information about capabilities and behavior of other agents is stored in each individual agent. We call the stored information about other agents an “acquaintance list.” In the acquaintance model, the individual agent contains information on the current capabilities of peers of the agent (Mařík et al., 1999).

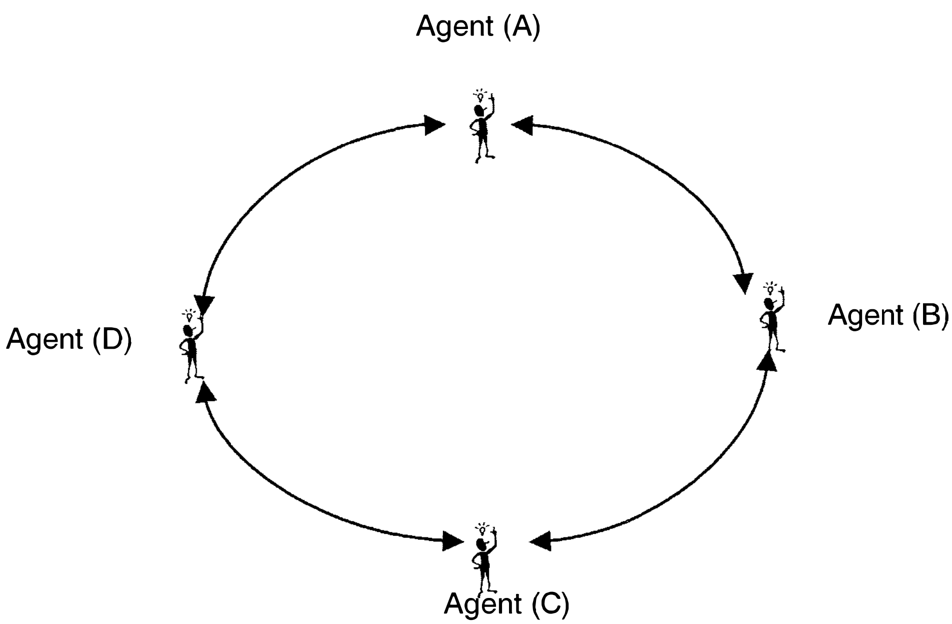

Figure 5: Agents Communicate with Each Other through Peer-to-Peer Mode

Mařík et al. (1999) suggested the tri-base acquaintances model. It can be viewed as a specific knowledge-based system that is able to combine permanent and temporary knowledge with facts stored in distributed databases. This approach suggests organizing the relevant information about cooperating agents into three separate information bases in the agent’s wrapper as follows:

- *The Co-operator Base* stores static information on the peer agents, such as their IP addresses, communication means and predefined responsibilities. Moreover, it specifies subscribed agents – agents that are subscribed to report their status change.
- *The Task Base* contains knowledge concerning possible task decomposition with respect to problem solving processes. It is further split into two separate sections. The problem section contains general knowledge on possible task decomposition and contingent time precedence and prerequisites. The plan section stores deduced plans on how to solve particular tasks through co-ordination of sub-contracted helping agents.
- The *State Base* reflects the actual peer agents' states that may evolve rapidly in time. The agent section of the state base reflects the internal states of the peer agents like their current load, attainability and trust, as well as capabilities (e.g., speed and price of processing) and schedule of the considered agent. The task section describes the current states of the solution of the tasks that have been contracted and are coordinated by the agent. The peers are expected to report the solution progress.

The state base reflects the actual peer agents' states that may evolve rapidly in time. The peers are expected to report the solution process to the related agent. This structure makes it possible for the task base to have up-to-date information on the current capabilities of the peers. This facilitates directing the co-operation requests to the most suitable agent in the community. Therefore, the communication traffic is significantly reduced and the system response is very fast since non-addressable task announcements are avoided.

6.2 Coordination Mechanisms

Co-operative multi-agent systems offer a novel approach to handle complex integration tasks. All participants in a coordination process have interdependencies. Coordination entails managing dependencies between activities (Schumacher, 2001). Ossowski (1999) has classified agent coordination into three groups: multi-agent planning, negotiation, and organization.

- Multi-agent planning

  With multi-agent planning, agents form plans that specify all their future actions and interactions with respect to a particular objective: all agents involved in a multi-agent plan commit to behave in accordance with it. This plan describes all actions that are required to achieve the respective goals of agents. The planning can be centralized or decentralized.

- Negotiation

  Negotiation is seen as a method for coordination and conflict resolution (Sycara, 1998), and a process by which two or more parties make a joint decision (Zhang et al., 2001). The parties first verbalize and then move toward an agreement through a process of concession formation; or they search for new alternatives (Mueller, 1996). Negotiation processes dynamically generate an agreement, which usually lasts shorter than a priori commitments that organizational structures imply. Still, agreements can be re-negotiated. The most promising application areas for agent negotiation include retail e-commerce, electricity markets, bandwidth allocation, manufacturing planning and scheduling in subcontracting networks, distributed vehicle routing among independent dispatch centers, and electronic trading of financial instruments (Sandholm, 1999).

- Organization

  Organization is usually seen as a metaphor for a set of long-term structural relationships between roles. A role determines the expectations about the agent's individual behavior by describing the agent's responsibilities, capabilities and authority inside the MAS. When an agent agrees to play certain roles within an organization, they commit to comply with the behavior that these roles and their relationship imply.

Agents can improve coherence by planning their actions. Planning considers the constraints that the other agents' activities place on an agent's choice of actions, the constraints that an agent's commitments to others place on its own choice of actions, and the unpredictable evolution of the world caused by other unmolded agents. One direction of research in cooperative multiagent planning has been focused on modeling teamwork explicitly (Sycara, 1998). The joint-intentions framework (Cohen & Levesque, 1991) focuses on characterizing a team's mental state, called joint intention. A team jointly intends a team action if team members are jointly committed to completing that team action, while mutually believing that they are doing it. The model of SHAREDPLAN (Grosz and Sidner, 1990) is not based on a joint mental attitude but rather on a new mental attitude intending that an action be done. However, an individual agent's intention is directed towards its collaborator's actions or towards a group's joint action. Intention is defined using a set of axioms that guide a teammate to take action or enter into communication that enables or facilitates its teammates to perform assigned tasks (Tambe, 1997).

Negotiation is a coordination mechanism used in the distributed environment. Negotiation means a discussion in which the interested parties exchange information and come to an agreement (Davis and Smith, 1983). For example, CAP II is an office automation agent for time management. The agent works in the background as a personal digital assistant. Since much of the office work is performed with the cooperation of different people, the agent also models the workflow, including simple sequence work and complex negotiation work. CAP II performs the negotiation of meetings between the attendees by sending email messages back and forth. Agents operate strictly locally in a purely reactive manner without any planning. Each agent uses its owner's calendar but does not have access to the calendars of other participants. All synchronization works through communication and negotiation. Planning (i.e., deliberative behavior) is not necessary. Once the user has indicated the desire for a meeting, the CAP II agent generates a proposal and sends it to all invited attendees. If the recipients possess a CAP II system, their agent may negotiate until they find a commonly accepted date. Replies are called "bids." A CAP II agent accepts Yes, Not-then, Maybe, or No bids. If an agreement has been found, an additional "handshake" procedure of sending confirmation and validation message follows. In case those responses are missing, a time-out handler generates and sends remainder messages or, if unsuccessful, notifies its user so that he can contact the attendee directly (Bocionek, 1995).

An organization provides a framework for agent interaction through the definition of roles, behavior expectations, and authority relations. An organization gives each agent a high-level view of how the group solves problems and indicates the connectivity information to the agents so that they can distribute subproblems to competent agents. Examples of organizations include the following (Sycara, 1998):

Hierarchy: The authority of decision making and control is concentrated in a single problem solver (or specialized group) at each level in the hierarchy. Interaction is through vertical communication from superior to subordinate agent; and vice versa. Superior agents exercise control over resources and decision making.

Community of experts: This organization is flat, where each problem solver is a specialist in some particular area. Agents coordinate through mutual adjustment of their solutions so that overall coherence can be achieved.

Market: Control is distributed to the agents that compete for tasks or resources through bidding and contractual mechanisms.

Scientific community: Solutions to problems are locally constructed, then they are communicated to other problem solvers that can test, challenge, and refine the solution.

7. Discussion

In multi agent systems, agents communicate and cooperate with each other to solve problems. There are basically two kinds of control structures in multi-agent systems (Sikora and Shaw, 1998; Mařík et al., 1999): centralized control and decentralized control. In the centralized control structure, there is a central coordinator to whom all agents communicate their solutions. The service provider agents advertise their capabilities to the middle agent, and the middle agent takes the responsibility to dispatch the task to the right agent when it receives a service request. This control structure is based on the advertised agents' capabilities. As in the multi-agent system developed by Pouchard and Walker (2001), different agents in the system are distinguished according to their roles and responsibilities. When all the agents' roles and capabilities are similar or difficult to differentiate, the central control structure won't work. For example, in music fan societies, all music fans have an interest in and knowledge of music, and their interest and knowledge are changing with time – thus making it difficult for a middle agent to keep track of all possible music that would match with the changing interest of each user.

MASs are best suited for use in open systems with a large and dynamic number of agents (Turner and Jennings, 2001). Pouchard and Walker (2001) contend that the central control agent (CA) may create a bottleneck since the CA controls all information exchange for all other agents when the number of users increases. It is believed that the CA system can scale up to 100 users. Therefore, getting the right team of agents and controlling them is of prime interest in the decentralized control structure for a large number of users (Dignum et al., 2001). There is no middle agent in a decentralized control structure. Therefore, agents use an acquaintance list to communicate only with a small subset of agents (Sikora and Shaw, 1998). In the acquaintance model, the individual agent contains information on the current capabilities of peers of the agent. For example, PoliTeam is a groupware support system that makes use of intelligent agent technology and cased-based reasoning technique towards information sharing among team members (Bordetsky and Mark, 2000). In this system, feedback control relationships are captured into a multilayered model of organizational memory and transferred to users by agent-facilitators. The approach is based on a system dynamic approach to organizational learning when the group members constitute a small finite set with similar information needs. The question arises as to how we can extend the functionality of a system such as PoliTeam to share information among a very large number of decision makers who are unaware of each other's existence and/or information needs. Response to this question is the objective of our ongoing research.

References:

Awad, E. M., *Building Expert Systems: Principles, Procedures, and Applications*, West Publishing Co., Los Angles, CA, 1996.

Bellifemine, F., Poggi, A., and Rimassa, G., "Developing Multi-agent Systems with Jade," *Intelligent Agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986*, Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 89-103.

Bernus, P., and Uppington, G., "Co-ordination of Managemnet Activities-Mapping Organizational Structure to the Decision Structure," *Coordination Technology for Collaborative Application - Organizations, Processes, and Agents, W.Conen and G.Neumann (Eds), LNCS 1364*, Springer-Verlag, Berlin, 1998, pp. 25-38.

Bocionek, S.R., "Agent systems that negotiate and learn," *International Journal of Human-Computer Studies*, Volume 42, 1995, pp. 265-288.

Bordetsky, A., and Mark, G., "Memory-Based Feedback Controls to Support Groupware Coordination," *Information Systems Research, 2000 INFORMS*, Volume 11, Number 4, December 2000, pp. 366-385.

Bryson, J., and Stein, L. A., "Architectures and Idioms: Making Progress in Agent Design," *Intelligent Agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986*, Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 73-78.

Botti, V. J., Carrascosa, C., Julián, V., and Soler, J., "Modeling Agents in Hard Real-Time Environments," *MultiAgent System Engineering, 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAAMAW '99*, Valencia, Spain, June 30 - July 2, 1999, Proceedings, LNCS 1647, Springer 1999, pp. 63-76.

Chen, H., Chau, M., and Zeng, D., "CI Spider: a tool for competitive intelligence on the Web," *Decision Support Systems*, Volume 34, Issue 1, December 2002, pp. 1-17.

Cohen, P. R., and Levesque, H. J., "Confirmation and Joint Action," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991, Sydney, pp. 951-957.

Conway, D. G., and Koehler, G. J., "Interface Agents: Caveat Mercator in Electronic Commerce," *Decision Support Systems,* Volume 27, Number 4, Jan 2000, pp. 355-366.

Cothrel, J., and Williams, R. L., "On-line Communities: Helping Them Form and Grow," *Journal of Knowledge Management*, Volume 3, Number 1, 1999, pp. 54-60.

Curle, D., "Filtered news services: Solutions in search of your problem?" *Online*, Volume 22, Number 2, Mar/Apr 1998, pp.14-24.

Davis, R., and Smith, R.G., "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence*, Volume 20, Number 1, January 1983, pp. 63-109.

Decker, K., Sycara. K., and Zeng, D., "Intelligent Agents in Portfolio Management," *Agent Technology: Foundations, Applications, and Markets*, Jennings, N. R., and Wooldridge, M.J. (Eds.), Springer-Verlag, Heidelberg, 1998, pp. 267-281.

Dignum, F., Dunin-Keplicz, B., and Verbrugge, R., "Agent Theory for Team Formation by Dialogue," *Intelligent Agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986*, Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 150-166.

Dragoni, A. F., Giorgini, P., and Serafini, L., "Updating Metal States from Communication," *Intelligent agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986*, Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 15-30.

Durfee, E. H., "Distributed Problem Solving and Planning," *Multi-Agent Systems and Applications, LNAI 2086*, Luck, M., Marik, M., Stepankova, O. and Trappl, R. (Eds.), Springer-Verlag, Berlin, 2001, pp. 118-149.

Fernandez, G., and Wijegunaratne, I., "A Cooperation Approach to Distributed Application Engineering," *Coordination Technology for Collaborative Applications - Organizations, Processes, and Agents, Lecture Notes in Computer Science 1364*, Conen, W., Neuman, G., Goods, G., Hartmanis J. and Leeuwen, J. V. (Eds.), Springer-Verlag, Berlin, 1998, pp. 39-48.

Finin, T., Fritzson, R., McKay, D., and McEntire R., "KQML as an Agent Communication Language," *The Proceedings of the Third International Conference on Information and Knowledge management (CIKM'94),* ACM Press, November 1994, pp. 456-463.

Finin, T., Fritzson, R., McKay, D., and McEntire, R., "KQML - A Language and Protocol for Knowledge and Information Exchange," *Technical Report No. CS-94-02,* University of Maryland, Department of Computer Science, 1994b.

Finin, T., Labrou, Y., and Mayfield, J., "KQML as an Agent Communication Language," *Software Agents*, Bradshaw, J.M. (Ed.), The MIT Press, Cambridge, MA, 1997, pp. 291-316.

Flores, R. A., Kremer, R.C., and Norrie, D. H., "An Architecture for Modeling Internet-Based Collaborative Agent Systems," *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, LNAI 1887, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Barcelona, Spain, June 2000, Revised Papers,* Wagner, T. and Rana, O. (Eds.), Springer-Verlag, Berlin, 2001, pp. 56-63.

Gallegos, F., "Decision support systems: an overview," *Information Strategy*, Volume 15, Number 2, Winter 1999, pp. 42-45.

Grosz, B. J., and Sidner, C. L., "Plans for Discourse," *Intentions in Communication*, in *Cohen, P. R., Morgan, J., & Pollack, M., (Eds.),* MIT Press, Cambridge, MA, 1990, pp. 417-445.

Hayes-Roth, B., "A Blackboard Architecture for Control," *Artificial Intelligence*, Volume 26, Number 3, 1985, pp. 251-321.

Hayes-Roth, B., "An Architecture for Adaptive Intelligent Systems," *Artificial Intelligence*, Volume 72, Number 12, 1995, pp. 329-365.

Hayes, C., Cunningham, P., and Smyth, B., "A Case-Based Reasoning View of Automated Collaborative," *Case-Based Reasoning Research and Development, LNAI 2080, 4th International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, BC, Canada, July/August 2001,* Aha, D. W. and Watson, I. (Eds.), Springer-Verlag, Berlin, 2001, pp. 234-248.

Jennings, N.R., and Wooldridge, M., *Agent Technology: Foundations, Applications and Markets*, Springer-Verlag, Berlin, 1998.

Kang, N. and Han, S., "Agent-based e-marketplace system for more fair and efficient transaction," *Decision Support Systems*, Volume 34, Issue 2, 2003, pp.157-165.

Kim, I., Hong, J., Bae, D., Han, I., and Youn, C., "Scalable Mobile Agents Supporting Dynamic Composition of Functionality," *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, LNAI 1887, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Barcelona, Spain, June 2000,* *Revised Papers,* Wagner, T., and Rana, O. (Eds.), Springer-Verlag, Berlin, 2001, pp. 199-213.

Krishnan, R., Li, X. and Steier, D., "On Heterogeneous Database Retrieval: A Cognitively Guided Approach, " *Information Systems Research,* Volume 12, Number 3, Sept. 2001, pp. 286-301.

Kwak, M., "Web Sites Learn to Make Smarter Suggestions," *MIT Sloan Management Review*, Volume 42, Number 4, Summer 2001, pp. 17.

Lewis, M., "Designing for Human-Agent Interaction," *AI Magazine*, Volume 19, Number 2, Summer 1998, pp. 67-78.

Lieberman, H., "Autonomous Interface Agents," *Proceedings of the ACM Conference on Computers and Human Interface*, CHI-97, Atlanta, Georgia, March 1997, pp. 67-74.

Lieberman, H., and Selker, T., "Out of Context: Computer Systems that Adapt to, and Learn from, Context," *IBM Systems Journal,* Volume 39, Number 3/4, 2000, pp. 617-632.

Lucas, H.C., *Information Systems Concepts for Management*, New York, McGraw-Hill, 1995.

Mařík, V., Pěchouček, M., Lažanský, J. and Roche, C., "PVS'98 Agents: Structures, Models and Production Planning Application," *Robotics and Autonomous Systems*, Volume 27, 1999, pp 29-43.

Maes, P., "Agents that Reduce Work and Information Overload," *Communications of the ACM*, Volume 37, Number 7, July 1994, pp. 30-40.

Maes, P., "Smart Commerce: The Future of Intelligent Agents in Cyberspace," *Journal of Interactive Marketing*, Volume 13, Number 3, Summer 1999, pp. 66-76.

Maes, P., Guttman R.H., and Moukas A.G., "Agents that Buy and Sell: Transforming Commerce as We Know It," *Communications of the ACM*, Volume 42, Number 3, March 1999, pp. 81-87.

Müller, J. P., "Control Architectures for Autonomous and Interacting agents: A Survey," *Intelligent Agent Systems: Theoretical and Practical Issues: Based on a Workshop Held at Pricai'96, Cairns, Australia, August 26-30, 1996, Lecture Notes in Artificial Intelligence 1209*, August, 1996, Cavedon, L., Rao, A., and Wobcke, W. (Eds.), Springer-Verlag, Berlin, April 1997, pp. 1-26.

Mueller, H. J., "Negotiation Principles," *in Foundations of Distributed Artificial Intelligence,* O'Hare, G. M. P. and Jennings, N. R. (Eds.), John Wiley & Sons, Inc, 1996.

Newell, A., "Putting It All Together," *Complex Information Processing: The Impact of Herbert Simon,* Klahr, D. & Kotovsky, K. (Eds.), Hillsdale, NJ: Lawrence Erlbaum, 1988.

Nodine, M., Chandrasekara, D., and Unruh, A., "Task Coordination Paradigms for Information Agents," *Intelligent Agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986*, Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 167-181.

Norman, T. J., and Reed, C., "Delegation and Responsibility," *Intelligent agents VII, Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 2000 Proceedings, LNAI 1986,* Castelfranchi, C. and Lespérance, Y. (Eds.), Springer-Verlag, Berlin, 2001, pp. 136-149.

Nwana, H. S., "Software Agents: An Overview," *Knowledge Engineering Review*, Volume 11, Number 2, 1996, pp. 1-40.

O'Dell, C., Grayson, C. J., and Essaides, N., *If Only We Knew What We Know: The Transfer of Internal Knowledge and Best Practice*, New York, The Free Press, 1998.

O Donnell, D., and O Regan, P. C. B., "Intellectual capital: a Habermasian introduction", *Journal of Intellectual Capital*, Volume 1, Number 2, 2000, pp. 187-200.

Ossowski, S., *Co-Ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents, Lecture Notes in Artificial Intelligence 1535*, Springer-Verlag, Berlin, 1999.

PenÄa-Mora, F., Hussein, K., Vadhavkar, S. and Benjamin, K., "CAIRO: a concurrent engineering meeting environment for virtual design teams," *Artificial Intelligence in Engineering* , Volume 14, Issue 3, July 2000, pp. 203-219.

Pilkington, R.M., *Intelligent Help: Communicating with Knowledge-Based Systems*, Oxford University Press, Oxford, U.K., 1992.

Pouchard, L., and Walker D. W., "A Community of Agents for User Support in a Problem-Solving Environment," *Infrastructure for Agents, Multi-Agent Systems,* *and Scalable Multi-Agent Systems, LNAI 1887, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Barcelona, Spain, June 2000, Revised Papers,* Wagner, T., and Rana, O. (Eds.), Springer-Verlag, Berlin, 2001, pp. 192-198.

Ram, S., "Intelligent agents and the World Wide Web: Fact or fiction?" *Journal of Database Management,* Volume 12, Number 1, January-March 2001, pp. 46-47.

Reticular, *User Guide*, Version 1.1, Rev. 4, www.reticular.com, 1999.

Rhodes, B J., and Maes, P., "Just-in-time information retrieval agents," *IBM Systems Journal,* Volume 39, Number ¾, 2000, pp. 685-704.

Ricci, A., Denti, E., and Omicini, A., "Agent Coordination Infrastructures for Virtual Enterprises and Workflow Management," *Cooperative Information Agents V, LNAI 2182, 5th International Workshop, CIA 2001 Modena, Italy, September 2001 Proceedings, Klusch, M., and Zambonelli, F. (Eds.),* Springer-Verlag, Berlin, 2001, pp. 235-246.

Roxanne, H., S., and Turoff, M., "What Makes Learning Networks Effective?" *Communications of the ACM*, Volume 45, Number 4, April 2002, pp. 56-59.

Ruggles, R., "The State of the Notion: Knowledge Management in Practice," *California Management Review*, Volume 40, Number 3, 1998, pp. 80-89.

Sandholm, T., "Automated Negotiation", *Communications of the ACM*, Volume 42, Number 3, March 1999, pp. 84-85.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Item-Based Collaborative Filtering Recommendation Algorithms," *Proceedings of the Tenth International World Wide Web Conference*, WWW 10, Hong Kong, P.R. China, May 1-5, ACM, 2001, pp. 285-295.

Sauter, V., *Decision Support Systems*, New York, John Wiley & Sons, 1999.

Schumacher, M., *Objective Coordination in Multi-Agent System Engineering: Design and Implementation*, *Lecture Notes in Artificial Intelligence 2039*, Springer-Verlag, 2001.

Serugendo, G. D. M., "A Formal Development and Validation Methodology Applied to Agent-Based Systems," *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, LNAI 1887, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Barcelona, Spain, June 2000,* *Revised Papers,* Wagner, T., and Rana, O. (Eds.), Springer-Verlag, Berlin, 2001, pp. 214-225.

Shaw, N. G, Mian, A., and Yadav, S. B, "A Comprehensive Agent-Based Architecture for Intelligent Information Retrieval in a Distributed Heterogeneous Environment," *Decision Support Systems,* Volume 32, Number 4, March 2002, pp. 401-415.

Sikora, R., and Shaw, M. J, "A Multi-Agent Framework for the Coordination and Integration of Information Systems," *Management Science,* Volume 44, Number 11 (Part 2), November 1998, pp. S65-S78.

Smith, M., "Tools for Navigating Large Social Cyberspace," *Communications of the ACM*, Volume 45, Number 4, April 2002, pp. 51-55.

Simon, H., *The New Science of Management Decision*, Englewood Cliffs, NJ, Prentice Hall, 1977.

Simon, H.A., and Hayes, J.R., "Understanding complex task instructions," *Cognition and Instruction (Chap. 14),* Klahr, D. (Ed.), Hillsdale, NJ, Erlbaum, 1976.

Stoddart, L., "Managing Intranets to Encourage Knowledge Sharing: Opportunities and Constraints," *Online Information Review*, Volume 25, Number 1, 2001, pp. 19-28.

Stone, P., and Veloso, M., *RoboCup-97: Robot Soccer World Cup I, (Lecture Notes in Computer Science, 1395)*, Kitano, H., Siekmann, J., and Carbonell, J.G. (Eds.), Springer-Verlag, Berlin, 1998.

Sutton, R.S., and Barto, A.G., *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, 1998.

Swan, J., Newell, S., Scarbrough, H., and Hislop, D., "Knowledge Management and Innovation: Networks and Networking," *Journal of Knowledge Management*, Volume 3, Number 4, 1999, pp. 262-275.

Sycara, K., "Multiagent Systems," *AI Magazine*, Volume 19, Number 2, 1998, pp. 79-92.

Sycara, K., Decker, K., and Williamson, M., "Middle-Agents for the Internet," *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97*, Nagoya, Japan, August 23-29, Volume 1, Morgan Kaufmann, 1997, pp. 578-583.

Sycara, K., Decker, K., Pannu, A., Williamson, M., and Zeng, D., "Distributed Intelligent Agents," *IEEE Expert*, December 1996, pp. 36-46.

Tambe, M., "Toward Flexible Teamwork," *Journal of Artificial Intelligence Research,* Volume 7, Number 1, 1997, pp. 83-124.

Tu, H. C., and Hsiang, J., "An Architecture and Category Knowledge for Intelligent Information Retrieval Agents," *Decision Support Systems*, Volume 28, Number 3, May 2000, pp. 255-268.

Turban, E., and Aronson, J.E., *Decision Support Systems and Intelligent Systems*, 5th ed. Prentice-Hall, NJ, 1998.

Turner, P. J., and Jennings, N. R., "Improving the Scalability of Multi-Agent Systems," *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, LNAI 1887, International Workshop on Infrastructure for Scalable Multi-Agent Systems, Barcelona, Spain, June 2000, Revised Papers,* Wagner, T., and Rana, O. (Eds.), Springer-Verlag, Berlin, 2001, pp. 246-262.

Turowski, K., "Agent-Based e-Commerce in Case of Mass Customization," *International Journal of Production Economics,* Volume 75, Number 1-2, January 10, 2002, pp. 69-81.

Wooldridge, M., "Verifying that Agents Implement a Communication Language," *AAAI-99 Sixth National Conference on Artificial Intelligence*, Orlando, Florida, July 18-22, 1999, pp. 52-57.

Zhang, X.Q., Lesser, V., and Podorozhny, P., "New Results on Cooperative, MultiStep Negotiation Over a Multi-Dimensional Utility Function," *AAAI Fall 2001 Symposium on "Negotiation Methods for Autonomous,"* AAAI Press, January 2001, pp. 1-10.

MeRC

McMaster eBusiness Research Centre

McMaster eBusiness Research Centre (MeRC)
Michael G. DeGroote School of Business
McMaster University
1280 Main St. W. MGD A201
Hamilton, ON
L8S 4M4

Tel: 905-525-9140 ext. 27027
Fax: 905-528-0556
Email: ebusiness@mcmaster.ca
Web: http://merc.mcmaster.ca