Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

Metadatabase Meets Distributed AI

Gilbert Babin, Zakaria Maamar and Brahim Chaib-draa

{babin,maamar,chaib}@ift.ulaval.ca
Département d'informatique, Université Laval
Ste-Foy, Québec, Canada, G1K 7P4

Abstract. Heterogeneous Distributed Database Management Systems (HDDBMS) involve the interoperability of data sources. One approach to achieve this type of integration is to build interfaces between the different databases being integrated. This approach holds, for a particular case, at a specific point in time. In this case however, the database structures need to be adapted. Such adaptation is not advisable since the local systems are usually important for their organizations. Therefore, an integration model that assures flexibility and scalability must be based on some knowledge of the underlying model of the different local databases. One solution is the use of the metadata concept, as a means to describe the logical and physical data characteristics. The metadata concept leads to the development of a Metadatabase system, which is viewed as a knowledge base about the local systems. The Metadatabase work at Rensselaer Polytechnic Institute (Troy, New- York) [11] and Université Laval (Ste-Foy, Québec) [2] has focused on creating such an integration environment and on defining its principal components. These solutions have been developed outside the context of Distributed Artificial Intelligence (DAI) and would certainly benefit from the results in that field of research. In this paper, we explain how the Metadatabase approach can be mapped into or associated with DAI concepts, and how it could benefit from techniques and theories pertaining to the DAI field.

Keywords — heterogeneous distributed database management systems, manufacturing systems, distributed knowledge processing, Metadatabase, distributed artificial intelligence.

Gilbert Babin, 3502 Adrien-Pouliot, Département d'informatique, Université Laval, Ste-Foy, Québec, Canada, G1K 7P4. E-mail :babin@ift.ulaval.ca, Phone : +1 (418) 656-3395, Fax : +1 (418) 656-2324.

Zakaria Maamar, 3502 Adrien-Pouliot, Département d'informatique, Université Laval, Ste-Foy, Québec, Canada, G1K 7P4. E-mail : maamar@ift.ulaval.ca, Phone : +1 (418) 656-2131 (ext. 4799), Fax : +1 (418) 656-2324.

Brahim Chaib-draa, 3502 Adrien-Pouliot, Département d'informatique, Université Laval, Ste-Foy, Québec, Canada, G1K 7P4. E-mail : chaib@ift.ulaval.ca, Phone : +1 (418) 656-2131 (ext. 3226), Fax : +1 (418) 656-2324.

Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

Metadatabase Meets Distributed AI

Gilbert Babin, Zakaria Maamar and Brahim Chaib-draa

{babin,maamar,chaib}@ift.ulaval.ca
Département d'informatique, Université Laval
Ste-Foy, Québec, Canada, G1K 7P4

Abstract. Heterogeneous Distributed Database Management Systems (HDDBMS) involve the interoperability of data sources. One approach to achieve this type of integration is to build interfaces between the different databases being integrated. This approach holds, for a particular case, at a specific point in time. In this case however, the database structures need to be adapted. Such adaptation is not advisable since the local systems are usually important for their organizations. Therefore, an integration model that assures flexibility and scalability must be based on some knowledge of the underlying model of the different local databases. One solution is the use of the metadata concept, as a means to describe the logical and physical data characteristics. The metadata concept leads to the development of a Metadatabase system, which is viewed as a knowledge base about the local systems. The Metadatabase work at Rensselaer Polytechnic Institute (Troy, New- York) [11] and Université Laval (Ste-Foy, Québec) [2] has focused on creating such an integration environment and on defining its principal components. These solutions have been developed outside the context of Distributed Artificial Intelligence (DAI) and would certainly benefit from the results in that field of research. In this paper, we explain how the Metadatabase approach can be mapped into or associated with DAI concepts, and how it could benefit from techniques and theories pertaining to the DAI field.

1 Introduction

The computer network domain and the globalization of economy have forced the enterprises to adopt a distributed structure which, in turn, implies a distribution of resources of those enterprises, particularly their information systems. In this case, it becomes more and more important to provide enterprises with integration tools to consolidate the information available throughout the distributed databases. The integration concept is supported by an interoperability process which means the ability of two or more distributed database systems to mutually exchange information, independently of their constraints of distribution and heterogeneity, in order to work together to execute well-defined and delimited tasks jointly. Notice that the integration approaches traditionally proposed assume the interoperability at the application level (i.e. between local systems) via a global schema or a common manipulation language [6,18]. On one hand, designing a global schema implies the combination of different kinds of domains in one model which is generally difficult to obtain and to maintain. On the other hand, finding a common language is not easy because each system has its own standards and needs. One original solution to the integration problem that assures flexibility and scalability is the use of the *metadata* concept. This concept lead to the development of a Metadatabase system, that is, a knowledge base about the logical and physical data characteristics of local systems.

The Metadatabase work at Rensselaer Polytechnic Institute (Troy, New-York) [11] and Université Laval (Ste-Foy, Québec) [2] has focused on creating an integration environment and on defining its principal components, producing a Metadatabase-supported, Rule-Oriented concurrent systems solution to the enterprise information integration and management problem. These solutions have been developed outside the context of Distributed Artificial Intelligence (DAI) and would certainly benefit from the results in that field of research, like using the software agents as modules for the problem resolution [14] and the multiples techniques (interaction, cooperation, negotiation, etc.) as principles for the behavior specification [15].

In this paper, we explain how the Metadatabase approach can be mapped into or associated with DAI concepts, and how it could benefit from techniques and theories pertaining to the DAI field. Section 2, which follows, describes in more details the Metadatabase approach and the internal functions of ROPE shells. In Section 3, we give an overview of the Distributed Artificial Intelligence (DAI) field and its application to the distributed Knowledge Based-Systems (KBS). Section 4 investigates the use of DAI to improve the Metadatabase integration approach. We conclude the paper in Section 5.

2 The Metadatabase Approach

Rensselaer's Metadatabase approach [3,4,12,13] was developed to integrate information systems, more specifically manufacturing systems. Manufacturing systems are heterogeneous and distributed by nature. To produce finished goods, many systems must cooperate. These systems might include an Order Processing System, used to record customers' orders, a Process Planning System, determining the steps to follow to obtain finished goods, and a Shop Floor Control System, dealing with process planning, job assignment, production status, etc. These systems are highly specialized, in the sense that only certain systems can perform specific tasks; for instance, only certain machines can drill a hole into a sheet of metal, for instance. These systems are also autonomous; each one manages its own database, which makes data consistency an issue. The problem then is how to coordinate the tasks performed by each of these systems and integrate the information stored in their local databases.

Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

2.1 The Concurrent Architecture

The Metadatabase approach uses a concurrent architecture (Fig. 1) containing: (1) a central knowledge base and (2) distributed rule processors. The central knowledge base, called a Metadatabase, contains a description of the different (manufacturing) systems and the knowledge describing how they are integrated and semantically interrelated. This knowledge includes (1) database integrity rules, enforcing consistency across the distributed databases, and (2) business rules, automating information flows across these systems.

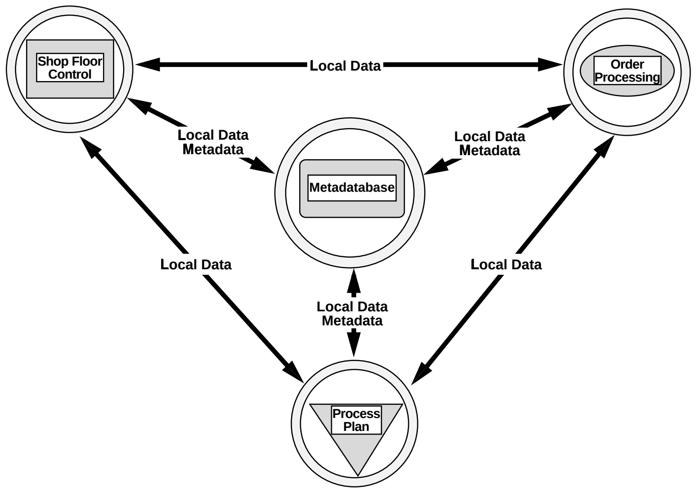

Fig. 1. The Metadatabase Concurrent Architecture

For each of the (manufacturing) systems, we define a rule processor. The rule processor's role is twofold. First, it encapsulates the system, ensuring that the system remains autonomous and independent, while making the system's capabilities available to other systems. Hence, although the shells are identical in structure, they differ in their capabilities, since each one has access to the special capabilities of the system it encapsulates. Second, it enables integration by providing knowledge processing capabilities to the system it encapsulates.

The Metadatabase and the rule processors form an integrated cell. The Metadatabase manages the knowledge, providing a global coherent view of the whole system, and distributes it to the local rule processors, hence achieving operational integration. In a Wide-Area Network (WAN), we can rely on a number of such integrated cells to provide a more robust framework (Fig. 2).

2.2 The Rule-Oriented Programming Environment

The ROPE (Rule-Oriented Programming Environment) [1] was developed as one implementation of the Metadatabase concurrent architecture. It creates a

Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

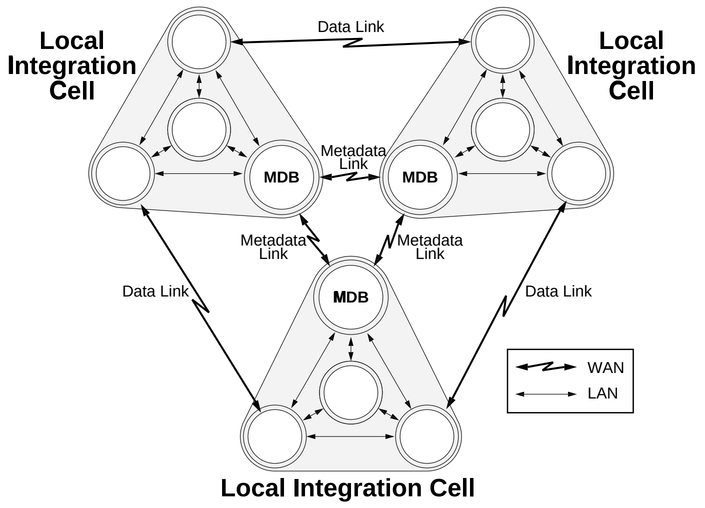

Fig. 2. Distributed Metadatabase Architecture

distributed rule processing environment, where fact bases and inference engines are distributed. The collaboration of rule processors is minimized by utilizing most of the information contained in the Metadatabase.

The ROPE approach defines the shell technology needed for the concurrent architecture evolution. Specifically, it defines: (1) how rules are stored in the local shells, (2) how rules are processed, distributed, and managed, (3) how the shell interacts with its corresponding system, (4) how the different shells interact with each other, and finally (5) how the shells are structured. The main advantage of the ROPE approach is that shells are invisible to the users. They are able to control the external behaviors of the local systems, without user intervention.

The ROPE shells have the following functions (See [1] for more details):

1. Global Query Processing: a global query is a data retrieval query that needs the participation of one or many local systems from the integration environment.
2. Global Update Processing: a global update processing represents a set of transactions (insertion, deletion or modification) that act on the data of the local systems. As a result, every local behavior change that has an impact (as updating data which pertain to another local system) on the behavior of the integration environment, has to be immediately transmitted to the rest of the appropriate systems.
3. Adaptability and Flexibility: the adaptability of the concurrent architecture refers to the ability of shells to change their behavior. This change is based on the knowledge stored in the Metadatabase.

Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

2.3 Knowledge Decomposition Process

The Metadatabase is used to decompose the integration rules and to supply the shells' rulebase with these decomposed rules, at build-time, or whenever a change occurs in the knowledge stored in the Metadatabase (for adaptability and flexibility). These rulebases are then used by the shells to perform the integration tasks, at run-time.

If all the rules were processed centrally, it would result in processing bottlenecks. To avoid this problem, and to take into account the specificities of the different shells, the rules are decomposed and distributed. The rule decomposition process is also constrained by the fact that the different rule processors (shells) have different capabilities. This is in fact the case in manufacturing systems, where a single machine can perform a specific operation (drill, sand, etc.). It also implies that the different rule processors must collaborate to a certain extent in order to execute the rule; i.e., specific actions must be executed from the appropriate location.

The approach proposed in [2] decomposes the rule in such a way that (1) the different rule processors do not need to share the rule execution control, and (2) there is no need for a central process controller. Process control, however, is passed sequentially from one rule processor to the other, using a simple message protocol. A rule is decomposed into subrules; these subrules are then distributed to the shells. The decomposition algorithm uses the fact that a rule can be broken down into five stages of execution: (1) rule triggering, (2) data retrieval, (3) condition evaluation and actions execution, (4) result storage, and (5) rule chaining. The idea is to serialize the execution of the rule using these five stages and to assure that the condition and actions of each subrule produced are localized in the same rule processor.

Once the rule is triggered, a global query is launched by the shell where the triggering event occurred. This global query was preprocessed by the Metadatabase, prior to rule distribution. The result from the global query is a fact table that will be used by the rule processors. Each subrule will be executed in sequence at the appropriate shell, using that fact base. When every subrule is executed, the data contained in the fact base will be used to update the (distributed) databases from which the data was retrieved. Rule chaining occurs then, all possibly chained rules being launched in parallel. The decomposition process will minimize the total number of subrules to reduce the network load and the total execution time.

3 An Overview of DAI

Generally, the DAI field aims to construct systems of intelligent entities that interact productively with one another. More precisely, DAI is concerned with studying a broad range of issues related to the distribution and coordination of knowledge and actions in environments involving multiple entities. These entities, called *agents*, can be viewed collectively as a society. The agents work

Gilbert Babin, Zakaria Maamar, and Brahim Chaib-draa. "Metadatabase Meets Distributed AI." in International Workshop on Cooperative Information Agents '97 (CIA'97). Kiel, Germany. Peter Kandzia and Matthias Klusch (eds.). Lecture Notes in Artificial Intelligence. No. 1202. Springer Verlag. February 1997. pp. 138-147.

together to achieve their own goals, as well as the goals of the society considered as a whole.

A major distinction in the DAI field is between research in distributed problem solving (DPS), and research in multiagent systems (MAS) [7]. Early DPS work concentrated on applying the power of networked systems to a problem exemplified by the three–phase nomenclature. In the first phase, the problem is decomposed into subproblems. The decomposition process may involve a hierarchy of partitionings. The second phase involves solution of the kernel subproblems by agents that communicate and cooperate as necessary. Finally, the subproblems' results are integrated to produce an overall solution. DPS work also addressed the robustness available from multiple sources of expertise, multiple views and multiple capabilities. Generally, multiple views referred to distributed applications (air-traffic control, urban-traffic control, etc.). In summary, in all of DPS work, the emphasis was on the *problem* and how to get multiple intelligent entities (programmed computers) to work together to solve it in a efficient manner.

In MAS, the agents are autonomous, potentially preexisting, and typically heterogeneous. Research here is concerned with coordinating intelligent behaviors among a collection of autonomous agents, that is, how these agents can coordinate their knowledge, goals, skills, and plans jointly to take action and to solve problems. In this type of environment, the agents may be working toward a single global goal, or toward separate individuals goals that interact. Like solvers in DPS, agents in MAS might share knowledge about tasks and partial works. Conversely, to the DPS approach however, they must also reason about the process of coordination among the agents. Coordination is central to multiagent systems: without it, any benefits of interaction vanish and the behavior of the group of agents can become chaotic.

4 Towards an Approach Based on DAI for Interoperability

DAI and distributed database systems (DDBS) concentrate on different objectives and hence use different techniques for achieving their purposes. Thus, DAI concentrates on DPS and on interaction between autonomous intelligent agents whereas DDBS basically deals with the problem of management and integration of a collection of distributed data. However, from both a conceptual as well as a functional point of view, there are some similarities between our Metadatabase model and DAI approach.

4.1 The Metadatabase architecture is a Distributed Problem Solving System

The DPS approach suggests that a certain category of problems can be resolved by distributing them on a collection of nodes. This suggestion is motivated by the following points: the problems have a distributed and heterogeneous nature, the network imposes a distributed vision and software engineering is oriented towards a design based on autonomous and interactive entities, i.e. software agents. These elements are supported by the Metadatabase technology. Each ROPE shell has the required knowledge to function locally. But, in order to accomplish a global operation, all the ROPE shells need to cooperate with each other. This kind of operation generally implies the management of knowledge disparities, access privileges, distributed sources, resource conflict, etc. The DAI principles used in these situations seem adequate.

4.2 The Metadatabase Architecture is a Multiagent System

The primary objective of the Metadatabase architecture is to build a cooperative environment. As mentioned in Section 1, the Metadatabase allows a set of distributed and heterogeneous local systems to interact, without changing their internal structure or behavior. It should be noted that application systems need not be modified to interoperate. The ROPE shells support all the operations from and to the different systems. Combining a local system with its representative ROPE shell gives an integrated component. The same approach can be found in [5]. The author considers a Cooperative KBS as a collection of autonomous KBS, referred to as agents, which are able to interact with each other. The same idea is found in the Metadatabase approach, in which the ROPE shell can be seen as a reactive agent. Data and rules are contained in the local system and the ROPE shell, respectively. Rules (integrity and business) determine the accomplishment of (1) the consistency across systems, (2) the reliability of the overall environment, and (3) the integration of sub-solutions. Furthermore, each shell has specific capabilities which are the actions that it can perform within the local system. These capabilities are unique, since we assumed, for sake of simplicity, that every local system has unique functions (e.g., no two machines can perform a drilling operation).

It is also possible to consider the Metadatabase architecture as a multiagent system. Each shell-application pair has specific functionalities that describe how to support its responsibilities, how to interact with the external world and particularly how to cooperate to support the work of other pairs. These functionalities are globally controlled by the Metadatabase (Fig 1). The management mechanisms of the central knowledge can be compared to a coordination protocol which controls the distributed knowledge, manages the knowledge of the domain application via the metadata, supports *ad hoc* requests, etc.

The coordination aspect in a multiagent system implies two elements [8]: mechanisms needed to establish reliable communication with the right agent, and mutual and useful understanding of the communication contents. For such problems, some authors suggest the introduction of an additional kind of agents, communication facilitators [9] and communication mediators[17], respectively. Such facilitators and mediators communicate among themselves using well-defined protocols which are independent from the language or languages which the agent uses to exchange knowledge; see [10] for other examples. The different functionalities being associated to these new kind of agents are included in the

Metadatabase architecture. The communication aspect is handled by the ROPE shell, since it uses its own knowledge to know how to interact with another local system via its associated ROPE shell. In addition, the semantic problem of the exchanged knowledge can be supported by the content of the Metadatabase schema. It uses the metadata concept to have a extensive description of the local information characteristics (as, formats, structures, values, etc.). Furthermore, the Metadatabase (and the ROPE shells) uses a special type of rules (equivalence rules) to ensure a coherent transformation of knowledge during the interaction between local systems.

4.3 Improving the Metadatabase Approach Using DAI Concepts

In their current state, the shells only react to events occurring in their local environment: changes in the local databases, reception of a message, etc. In fact, inter-shell communications are at their simplest expression: a shell receives execution orders from other shells. Furthermore, the relative order in which these execution commands are performed is predetermined during the rule decomposition process. It is clear that, since only a specific shell has a specific capability, only that shell can perform that specific operation. In that context, using negotiation techniques is useless and even cumbersome.

However, this assumption does not reflect the reality of manufacturing systems. Often times, a same functionality will be performed by multiple instances of the same machine. In this perspective, the decomposition process could determine the subrules needed, but not the processor which will execute it. In that new context, a shell must be able to select one or many processors that will be able to execute the next subrule, most likely based on the workload of the different shells having the appropriate capabilities. We can now see how negotiation and inter-agent cooperation techniques could be used to improve the functionalities of the shell. For this type of approach to be feasible, the shells must have a minimal notion of self-knowledge. For instance, they should be able to answer the following questions:

- What are my functional capabilities (types of functions I can accomplish)?
- What is my current workload?
- What are my processing capabilities (amount of work I can accomplish)?
- etc.

In this case, shells can be considered as autonomous cognitive agents that have the capacity to coordinate their activities through contracts to accomplish specific tasks [16]. A shell, acting as *manager*, decomposes rules into subrules to be accomplished by other *potential contractor* shells. For each subrule the manager announces a task to the other shells. These shells receive and evaluate the announcement, and those with appropriate resources, workload and capabilities reply to the manager with *bids* that indicate their ability to achieve the announced task. The manager evaluates the bids it has received and awards the task to the most suitable shell, called the contractor. Finally, manager and contractor exchange information together during the accomplishment of the task.

The cooperation between Metadatabase model and DAI approach brings to us many issues for our future work, particularly the following aspects:

- Enabling local shells to represent and reason about locally stored data and knowledge as well as information regarding other agents;
- Forming coalition of shells (or cells) according to shared characteristics, such as domain of expertise;
- Enabling local cells to negotiate and interact efficiently;
- Reconciling the disparate views (by the negotiation) between shells and between local cells.

5 Conclusion

In this paper, we presented the Metadatabase approach as a solution to the problem of systems integration in heterogeneous and distributed environments, more specifically for manufacturing systems. We also showed how the Metadatabase approach can be assimilated to a Distributed Artificial Intelligence (DAI) system. Finally, we proposed ways to improve the Metadatabase approach using the DAI perspective.

Acknowledgments

Part of this research was funded by the Natural Sciences and Engineering Research Council of Canada (grants #OGP0155899 and #OGP0121634), by Université Laval (Startup grant), and by Samsung Electronics Co.

References

1. G. Babin. *Adaptiveness in Information Systems Integration.* PhD thesis, Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, N.Y., August 1993.
2. G. Babin and C. Hsu. Decomposition of knowledge for concurrent processing. *IEEE Transactions on Knowledge and Data Engineering*, 1995. Forthcoming.
3. M. Bouziane. *Metadata Modeling and Management.* PhD thesis, Computer Sciences, Rensselaer Polytechnic Institute, Troy, N.Y., June 1991.
4. W. Cheung. *The Model-Assisted Global Query System.* PhD thesis, Computer Sciences, Rensselaer Polytechnic Institute, Troy, N.Y., November 1991.
5. S.M. Deen. A general framework for coherence in ckbs. *Intelligent Information Systems*, 1(3), 1 Septembre 1992.
6. D.M. Dilts and W. Hua. Using knowledge-based technology to integrate cim databases. *IEEE Expert*, 3(2):237–245, 1991.
7. E. Durfee and J. S. Rosenschein. Distributed problem solving and multi-agent systems: comparisons and examples. In *Proc. of 13th Int. DAI Workshop*, Seattle, USA, 1994.
8. T. Finin, R. Fritzson, and D. McKay. A language and protocol to support intelligent agent interoperability. has appeared in the Proceedings of the CE & CALS Washington '92 Conference, june 1992.

9. M.R. Genesereth. An agent-based approach to software interoperability. In *Proceedings of the DARPA Sotware Technology Conference*, 1992.
10. M.R. Genesereth and A.P. Ketchpel. Software agents. *Communication of the ACM*, 37(7):48–53, July 1994.
11. C. Hsu. *Enterprise Integration and Modeling — the Metadatabase Approach*. Kluwer Academic Publisher, Boston, Mass., USA, 1996.
12. C. Hsu, G. Babin, M. Bouziane, W. Cheung, L. Rattner, and L. Yee. Metadatabase modeling for enterprise information integration. *Journal of Systems Integration*, 2(1):5–39, January 1992.
13. C. Hsu, M. Bouziane, L. Rattner, and L. Yee. Information resources management in heterogeneous, distributed environments: A metadatabase approach. *IEEE Transactions on Software Engineering*, 17(6):604–625, June 1991.
14. S.-N. Hyacinth. Software agents: An overview. *Knowledge Engineering Review*, 11(3):1–40, Sept 1996.
15. B. Moulin and B. Chaib-Draa. An overview of distributed artificial intelligence. In *Foundations of Distributed Artificial Intelligence*, chapter Formulative Readings, pages 3–55. G.M.P. O'Hare and N.R. Jennings, 1996.
16. R. G Smith. The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Trans. on Computer*, 29(12), 1980.
17. G. Wiederhold. The architecture of future information systems. Stanford University Computer Science Dept., 1989.
18. W. Wu and D. M. Dilts. Integrating diverse cim data bases: The role of natural language interface. *IEEE Trans. on Syst., Man and Cyb.*, 22(6):1331–1347, 1992.