23 IMPLEMENTING INTERNET ENABLED VIRTUAL ENTERPRISES USING COLLABORATIVE AGENTS

Weiming Shen* and Douglas H. Norrie
University of Calgary, Canada

This paper presents the development of a Collaborative Agent System Architecture (CASA) and an Infrastructure for Collaborative Agent Systems (ICAS). Initially being proposed as a general architecture for Internet based collaborative agent systems, the proposed architecture and infrastructure are very suitable for implementing Internet enabled virtual enterprises. The main features of CASA and ICAS are described and some domain independent mechanisms and components are presented.

INTRODUCTION

Global competition and rapidly changing customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. Several new concepts have been proposed and developed in the past few years, e.g., Virtual Enterprise, Supply Chain Management, Electronic Commerce, etc. However, there is so far no unified definition of these concepts and a variety of definitions exist from different points of view in the literature. A comprehensive discussion of the Virtual Enterprise concept can be found in (Camarinha-Matos et al., 1997).

We consider Virtual Enterprise and Supply Chain Management are two similar concepts. The supply chain of a manufacturing enterprise can be defined as a world-wide network of suppliers, factories, warehouses, distribution centers and retailers through which raw materials are acquired, transformed into products which are then delivered to customers. So the Supply Chain Management a little more focuses on the *Chain* level and is related to the life cycle of products. The Virtual Enterprise focuses on the collaboration among the related manufacturing factories. While Electronic Commerce includes the ordering and financial transactions involved for suppliers, i.e., it usually focuses on just one link in the supply chain.

Recently, a number of research projects have been developing to define infrastructures and supporting functionalities for Virtual Enterprises. Among others, significant projects include NIIP in the USA for the development of open industry software protocols that will make it possible for manufacturers and their suppliers to

---

* Currently at: NRC-IMTI, London, Ontario, Canada, E-mail: weiming.shen@nrc.ca

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35577-1_37](https://doi.org/10.1007/978-0-387-35577-1_37)

effectively inter-operate (NIIP, 1999), and a number of European ESPRIT projects such as Prodnet II for the development of a reference architecture and supporting infrastructure for virtual enterprises particularly suited for small and medium sized enterprises (Camarinha-Matos et al., 1997). Agent technology has been recently recognized as a promising approach to implementing virtual enterprises and supply chains. Fox et al (1993) proposed organizing the supply chain as a network of cooperating, intelligent agents. A similar proposal has been made by Swaminathan et al (1996) using a multi-agent framework for modeling supply chain dynamics. Parunak and VanderBok (1998) proposed using three species of agents to model the supply chain: *Company agents* representing the different firms that trade with one another in a supply chain; *PPIC agents* modeling the production planning and inventory control algorithms used by company agents; and *Shipping agents* modeling the delay and uncertainty involved in the movement of both material and information between trading partners. MetaMorph II (Shen, Xue et al., 1998) proposed using a hybrid agent-based mediator-centric architecture to integrate partners, suppliers and customers dynamically with the principal enterprise through their respective mediators within a supply chain network. Brugali et al (1998) proposed applying mobile agent technology to implement supply chain networks. A state-of-the-art survey of agent-based manufacturing systems with a detailed discussion of key issues and an extensive annotated bibliography can be found in (Shen and Norrie, 1999).

In response to the rapidly expanding Internet technology and the Java programming language, we have been working through several projects towards the development of infrastructures and methodology for implementing Virtual Enterprises. Based on eight years research work on applications of agent technology in distributed intelligent manufacturing (Norrie and Shen, 1999), our group is now involved with industrial partners to develop and test an Infrastructure for Collaborative Agent Systems (ICAS) in a complex supply chain network. This paper presents the proposed infrastructure with related components and mechanisms, and discusses the benefits of implementing virtual enterprises with the proposed infrastructure. Due to space limitations, case studies could not be included in this paper. Interested readers may refer to (Shen, Norrie et al., 1999; Shen, Ulieru et al., 1999) for the descriptions of case studies on multi-plant production planning and supply chain management.

AN INFRASTRUCTURE FOR INTERNET ENABLED COLLABORATIVE AGENT SYSTEMS

Motivations and Objectives

Eight years research experience related to applications of agent technology in developing agent-based intelligent manufacturing systems give us an insight into requirements and key issues of agent-based manufacturing systems. An overview of related projects at the University of Calgary with a summary of several interesting techniques and mechanisms developed during these projects and a discussion of key issues can be found in (Norrie and Shen, 1999). Based on our previous research results, a new general purpose Collaborative Agent System Architecture (CASA)

and an Infrastructure for Collaborative Agent Systems (ICAS) are proposed and being developed for the following reasons:

- a generic domain independent architecture is more reusable than a domain specific architecture;
- several interesting techniques and mechanisms have been developed and proved during our previous projects (Norrie and Shen, 1999), and they may become more interesting and useful after being developed into domain independent modules and mechanisms;
- emergent Internet technology and the Java programming language allow us to develop agent-based systems in a different way than previously.

Figure 1 shows an Infrastructure for Collaborative Agent Systems (ICAS). It also indicates our basic ideas and motivations to develop a generic collaborative agent system architecture.

Figure 1 - Infrastructure for Collaborative Agent Systems (ICAS)

Basic Communication and Cooperation Services

A Generic Collaborative Agent System Architecture

The generic Collaborative Agent System Architecture (CASA) is composed of collaborative elements such as agents, local area coordinators (LACs), yellow pages and cooperation domain servers. The objective of the CASA work is to support collaborative software agents by providing easy-to-use domain independent communication and cooperation services over the Internet. These services include conversation messaging services, lookup and search services, and remote call services etc. The provision of these services should significantly reduce the complexity of developing systems of collaborative agents.

Cooperation Domain Servers

The cooperation domain server listens to a specified port for new connections. Other agents may send requests to this port to create a new cooperation domain, or join an active or inactive cooperation domain. The server may allow this request by creating a new cooperation domain server (in the case of a new cooperation domain or a join to an inactive cooperation domain) and replying to the agent with the port number that the cooperation domain server will use. In addition, the server offers data storage services to the cooperation domain – storing and retrieving named stream data on request.

The cooperation domain server is the receiver of all messages sent by the agents in the cooperation domain. Although the exact content of the messages may be in any language (whether known or unknown to the server), all messages are in an extended KQML format (Finin and Labrou, 1997). Thus, the cooperation domain server can determine the ultimate destination and some of the high level semantics of the message. The cooperation domain server takes care of forwarding the message transparently to its destination, whether it is an individual agent, a list of agents, all agents of a particular service type (role), or all agents in the cooperation domain. In addition, the cooperation domain server may record all transactions to enable history playback and "group undo" operations. A collaborative agent system may have one or more cooperation domain servers.

Yellow Page Agents

Yellow Page agents (also called Yellow Pages, or YPs) are responsible to accept messages for registering services and to record this information in a local database. Other agents may later query the yellow pages to determine what agents offer a specified service and may retrieve descriptions of those agents' locations.

Area and Local Area Coordinators

An area is a convenient quasi-physical division of the network that can be controlled by a local area coordinator (LAC). The area may be a single computer, some arbitrary division of a single computer, or a cluster of computers. A local area coordinator acts both as a representative of the area to the outside world, and a manager for the local agents within the area. Any agent may send a message to the LAC to request that the LAC ask one of its local agents to join a particular cooperation domain. The LAC will invoke the agent if necessary, and pass along the request to the agent.

The LAC also facilitates local agent's communication to the outside world by providing an interface service to the outside world. Instead of each agent having to keep a model of network locations of servers and yellow pages, the LAC provides a central cache of such data. Local agents may query the LAC to obtain yellow page services; the LAC may, in turn, pass these requests to one or more known yellow pages and broker the responses. This simplifies network modeling requirements for individual agents.

Communication in CASA

In CASA, agents do not communicate with one another in an ad-hoc, point-to-point manner. Instead, agents working together form cooperation domains. Each agent in a cooperation domain routes all its outgoing messages through the cooperation domain server, which can direct it to a specific agent (imitating point-to-point communication), to several agents (imitating multicast communication), or to all agents in the cooperation domain (imitating broadcast communication). All incoming messages are received from the cooperation domain server as well (the original sender's identity is contained in the message header). The benefit of all messages going through the cooperation domain server is that the server can then provide several services (as described previously) and agents can address roles and services, even if they don't know the identity of the agent or agents performing these

roles or services. This type of communication is implemented by the Collaborative Agent Communication Language (CACL) that is developed by the same group. Initially, CACL allows for three simple actions: *request*, *reply* and *inform*.

Domain Independent Components

As mentioned previously, several interesting techniques and mechanisms have been developed and proved during our previous projects (Norrie and Shen, 1999), and they become more interesting and reusable after being developed into domain independent modules. We are now working on defining and developing such components such as collaborative interface agents, high-level collaboration agents, knowledge management (KM) servers (or agents) and ontology server(s) etc. As shown in Figure 1, these components utilize the basic communication and cooperation services provided by CASA.

All these components will become domain specific when implemented for a specific application and filled with domain specific information and knowledge. Detailed descriptions of these components are being written in separate reports and papers by other group members, and are also out of scope of this paper. However, we will give a brief introduction of these components here.

Collaborative Interface Agents

We consider that (domain independent) collaborative interface agents may have the following characteristics: communicative, semi-autonomous, collaborative, reactive, pro-active, adaptive, self-aware, and mobile. However, the implementation of interface agents in a real application does not need all these features. The development of collaborative interface agents is one of current research projects in our group, and the results will be reported separately.

High-Level Collaboration Agents

Although several CASA components such as YPs and LACs also provide basic collaboration (cooperation) services, some complex large collaborative agent systems often need more high-level and domain specific collaboration services which cannot be covered by YPs and LACs. In order to meet such requirements, the high-level collaboration agent is proposed as one of important components for ICAS. Such collaboration agents may be implemented by mediators as proposed and developed in our previous projects (Maturana and Norrie, 1996). In most cases, they are static, but they can also be implemented using dynamic mediators.

Knowledge Management Agents

Knowledge management is one of the most important issues in developing multi-agent systems. Typically, there are two approaches to knowledge management in multi-agent systems: knowledge is distributed among agents; some knowledge management agents are used to centralize knowledge management. Our approach to this issue is to combine above mentioned two approaches, i.e., in addition to distribute some knowledge among agents, several knowledge management agents are developed for specific problems. A knowledge management agent is usually associated with one or more databases and knowledge bases. The key issues to

develop knowledge management agents are to develop efficient mechanisms for knowledge acquisition, representation, learning and reasoning. Note that some simple knowledge management agents may be implemented by Yellow Page agents in CASA.

Ontology Server(s)

As one of the important components for our proposed infrastructure, ontology server(s) will be developed using *Ontolingua* (Gruber, 1993). The ontology server structure and related mechanisms are initially domain independent. It becomes domain specific when it filled with domain specific ontologies for a specific application, e.g., manufacturing production planning. Of course, it is not difficult to define the structure and mechanisms for an ontology server, but it is extremely difficult to develop and complete an efficient ontology server for an application domain. However, it has not been developed during our prototype implementation.

Organization, Coordination and Adaptation

The organization, coordination, adaptation and some other domain independent mechanisms as shown in Figure 1 have been separately proposed and developed during our previous research projects. These mechanisms include:

- Agent-Based Mediator-Centric Organization
- Task Decomposition
- Virtual Clustering
- Partial Agent Cloning
- Adaptation and Learning

Agent-Based Mediator-Centric Organization

Mediators are intended to encapsulate various manufacturing behaviors to facilitate the coordination of heterogeneous intelligent agents. A generic model for the design of mediators, based on the specification of various meta-level activities, was proposed and implemented during the MetaMorph I project (Maturana and Norrie, 1996). Such a generic model can enable different types of mediators to be created for various activities in the factory. These meta-level activities are high-level abstractions of behaviors that follow common patterns typical of different areas of expertise within the manufacturing system. A product's life cycle transits through these patterns at different stages of planning, scheduling, and control activity.

The generic model for mediators includes the following seven meta-level activities: Enterprise, Product Specification and Design, Virtual Organizations, Planning and Scheduling, Execution, Communication, and Learning. Each instantiated mediator includes some or all of these activities to a varying extent. Prototyping with this generic model allows the creation of diverse types of mediators. For example, a mediator may be specialized for organizational issues (enterprise mediator) or for shop-floor production coordination (execution mediator). Although each of these mediator types will have different manufacturing knowledge, both conform to a similar generic specification. More information about this generic model can be found in (Maturana and Norrie, 1996).

Task Decomposition Mechanism

The task decomposition mechanism developed during the MetaMorph I & II projects is one of the core mechanisms needed in agent-based manufacturing systems. Through this mechanism, high-level tasks are decomposed initially by mediators acting at the corresponding information level. Each subtask is subsequently distributed to agent clusters to determine the best solution plan. Mediators can learn dynamically from the agent interactions and identify clusters that can be used for distributed resolution of tasks. Subtasks are then further decomposed, and allocated through negotiation among resource agents. See (Maturana and Norrie, 1996) and (Shen, Xue et al., 1998) for more details.

Virtual Clustering Mechanism

The virtual clustering mechanism was proposed and developed during the MetaMorph I project. To work cooperatively, agents may form clusters that bond dissimilar agents into harmonious decision groups. Multistage negotiation and coordination protocols that can efficiently maintain the stability of these clusters are required. Each agent has its individual representation of the external world, goals, and constraints, so diverse heterogeneous beliefs interact within a cluster through distributed cooperation models.

Through the virtual clustering mechanism, agents can be dynamically contracted to participate in a problem-solving group (cluster). In the situation where the agents in the problem-solving group (cluster) are only able to partially complete the task, the agents will seek for help outside their cluster. This results in sub-clusters being formed for sub-tasks. This process is repeated, with sub-clusters being formed and then sub-sub-clusters etc as needed, within a dynamically inter-linked structure. As the respective tasks and subtasks are solved, the related clusters and links are dissolved. However, mediators will store the most relevant links with associated task information for future re-use. This clustering process, as implemented, provides scalability and aggregation properties to the system.

Partial Agent Cloning Mechanism

The agent (partial) cloning mechanism was proposed and implemented during the MetaMorph project. Through this mechanism, resource agents can be partially cloned as needed for concurrent information processing. These partial clone agents can then participate in virtual coordination clusters where agents negotiate with each other to find the best solution for a production task. A detailed description of this mechanism can be found in (Maturana and Norrie, 1996).

In a simulation, resource agents can be objects with goals and associated motivations, located in the same computer. Such clone agents are, in fact, clone objects. In a real implementation, the cloning mechanism can be used to partially 'clone' resource agents from remote resources (e.g., CNC and other manufacturing machines) to the computer where the resource mediators reside, so as to reduce communication time and consequently to reduce scheduling and rescheduling time.

Adaptation and Learning

Two fundamental learning mechanisms have been implemented in MetaMorph to enhance the system's performance and responsiveness, with mediators playing an essential role in both mechanisms. First, a mechanism that allows mediators to learn

from history is developed at the resource mediator level to capture significant multi-agent interactions and behaviors. Second, a mechanism for propagating the system's behavior into the future is implemented to help mediators 'to learn from the future'.

A 'learning from history' mechanism based on distributed case-based learning approach was developed for capturing agents' behavioral patterns at the resource mediator level and storing these in its knowledge base. Such knowledge is then reused for later manufacturing requests, through an extended case-based reasoning mechanism. The main purpose of 'learning from the future' is to modify promissory schedules at the resource agent level for otherwise unforeseen perturbations and changes in production priorities on the shop floor. The forecasting process simulates the behavior of the virtual model which emulates the shop-floor activities. By partially projecting 'unpredictable behaviors' and agent interactions, the agent-based manufacturing system is able to correct its real-world model and provide more accurate plans. A detailed description of these learning and reasoning mechanisms in MetaMorph have been presented separately (Shen, Maturana et al., 1998).

BENEFITS OF IMPLEMENTING INTERNET ENABLED VE WITH ICAS

Our previous research experience has shown that agent-based approaches potentially offer many advantages for the development of distributed intelligent manufacturing systems and the implementation of virtual enterprises: modularity; reconfigurability; scalability; upgradeability; and robustness (including fault recovery) (Norrie and Shen, 1999). Although the Infrastructure for Collaborative Agent Systems (ICAS) is proposed as a general approach for developing Internet enabled collaborative agent systems, most components and mechanisms proposed and developed under this infrastructure are very useful and suitable to implement Internet enabled virtual enterprises. This section discusses the benefits of implementing the Internet Enabled virtual enterprises with the ICAS components and mechanisms.

- *Cooperation Domain Servers (CDSs)*: In the Internet enabled supply chain or virtual enterprises, one of the most important services is the customer-supplier or business-business negotiation. CASA's *Cooperation Domain Servers* provide a natural way and media for such negotiation. It becomes more effective when it works together with the Virtual Clustering mechanism described above.
- *Yellow Page Agents (YPs)*: Because of the dynamics and complexity of the Internet enabled supply chain or virtual enterprises, *Yellow Page Agents* will play a very important role for providing look-up and search services.
- *Local Area Coordinators (LACs)*: *LACs* can significantly facilitate the organization and management of the Internet enabled supply chain or virtual enterprises. They are especially useful and important for the Internet enabled supply chain of large international manufacturing enterprises or widely geometrically distributed virtual enterprises.
- *Collaborative Interface Agents (CIAs)*: Different types of user interfaces are needed for the Internet enabled supply chain or virtual enterprises: interfaces for customers to input orders; interfaces for marketing and operation managers, production managers, enterprise general managers, etc.

Collaborative Interface Agents proposed and developed under ICAS can be employed to meet such requirements. Each type of interface may be composed of only several modules of the general CIA model.

- *High-Level Collaboration Agents (HLCAs)*: This type of *collaboration agents* can be used to provide collaboration services which cannot be covered by the basic CASA collaboration (cooperation) services, e.g., regional marketing coordinators, regional operation coordinators, local or regional production coordination, etc.
- *Knowledge Management Agents (KMAs)*: Managing a complex supply chain or a virtual enterprise needs a lot of knowledge ranging from customer requirements, marketing, product modeling, and project management etc. Several *knowledge management agents* may be developed to facilitate the management of supply chain or virtual enterprises.
- *Ontology Server(s)*: For the Internet enabled supply chain or a virtual enterprise as we described at the beginning of this paper, the ontology problem becomes more crucial because it is related to highly heterogeneous environments, e.g., multiple communication languages and multiple knowledge/data/information interchange formats (e.g., KIF, EDI, etc) may be used. Thus, one or more *ontology server(s)* should be employed to provide a standard for the supply chain or virtual enterprise on the terms used in communication languages and knowledge related these terms' definitions, attributes, relationships and constraints.
- *Virtual Clustering*: When a customer submits an order through a collaborative interface agent as described above, one regional marketing and operation center of the manufacturing enterprise needs to negotiate with different factories and partners for raw material (or parts) supply, parts fabrication, parts transportation, and product assembly etc. A number of collaborative agents including collaborative interface agents for customers and managers, automatic software agents for cost calculation, production planning, conflict detection and resolution and so on are involved in this negotiation process. The *Virtual Clustering* mechanism provides an efficient way to form a virtual collaboration group to facilitate such negotiation. At the sub-system level, this mechanism has been proved to be very useful during production planning and scheduling.
- *Agent Cloning*: Similar to the Virtual Clustering mechanism described above, agent cloning mechanism will be very useful for some collaborative agents to be involved in several collaboration groups (clusters) simultaneously, which will also reduce network communication load significantly. While at the sub-system level, it has also been proved to be very efficient during agent-based manufacturing planning and scheduling.
- *Adaptation and Learning*: A complex virtual enterprise or supply chain needs to dynamically adapt to its changing environment, including changing marketing conditions, virtual enterprise configuration, intra-/inter-enterprise interactions, financial policies etc. The implementation of learning mechanisms, e.g., at the high-level agents such as LACs, CIAs, HLCAs and KMAs, can enhance the performance of the supply chain or virtual enterprise networks.

CONCLUSIONS

Our previous research experience and the results of our ongoing research projects related to collaborative agents have shown that agent-based approaches potentially offer many advantages for the development of distributed intelligent manufacturing systems and the implementation of virtual enterprises. These advantages include open and dynamic system architecture, modularity, reconfigurability, scalability, upgradeability, and robustness (including fault recovery).

The Collaborative Agent System Architecture (CASA) and the Infrastructure for Collaborative Agent Systems (ICAS) are initially proposed as a general approach for Internet based collaborative agent systems. The theoretical analysis and the preliminary results of case studies have shown that the proposed architecture and infrastructure with domain independent components and mechanisms are quite suitable for implementing Internet enabled virtual enterprises.

REFERENCES

1. Brugali, D., Menga, G. and Galarraga, S. Inter-Company Supply Chains Integration via Mobile Agents. In Jacucci, G. (ed.), *Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility and the Virtual Enterprise*, Kluwer Academic Publisher, pp.43-54, 1998.
2. Camarinha-Matos, L.M., Afsarmanesh, H., Garita, C. and Lima, C. Towards An Architecture for Virtual Enterprises. In *Proceedings of the 2nd World Congress on Intelligent Manufacturing Processes & Systems*, Budapest, 1997.
3. Finin, T. and Labrou, Y. KQML as an agent communication language. In J.M. Bradshaw, (ed.), *Software Agents*, MIT Press, Cambridge, pp. 291-316, 1997.
4. Fox, M.S., Chionglo, J.F. and Barbuceanu, M. The Integrated Supply Chain Management System. Internal Report, Dept. of Industrial Engineering, Univ. of Toronto, 1993.
5. Gruber, T. A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 5(2), 199-220, 1993.
6. Maturana, F. and Norrie, D.H. Multi-Agent Mediator Architecture for Distributed manufacturing. *Journal of Intelligent Manufacturing*, 7, 257-270, 1996.
7. NIIIP. Vision of The National Industrial Information Infrastructure Protocols (NIIIP), 1999, <http://www.niiip.org/vision.html>
8. Norrie, D.H. and Shen, W. Applications of Agent technology for Agent-Based Intelligent Manufacturing Systems. *IMS'99*, Leuven, Belgium, 1999.
9. Parunak, V. and VanderBok, R. Modeling the Extended Supply Network. Presented at ISA-Tech'98, Houston, TX, 1998, available through the Internet at: <http://www.erim.org/~van/isa98.pdf>
10. Shen, W. and Norrie, D.H. An Agent-Based Approach for Dynamic Manufacturing Scheduling. In *Working Notes of the Agent-Based Manufacturing Workshop*, Minneapolis, MN, pp. 117-128, 1998.
11. Shen, W., Maturana, F. and Norrie, D.H. Learning in Agent-Based Manufacturing Systems. In *Proceedings of AI & Manufacturing Research Planning Workshop*, Albuquerque, NM, The AAAI Press, pp. 177-183, 1998.
12. Shen, W. and Norrie, D.H. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems, an International Journal*, 1(2), 129-156, 1999. (Its extended version in HTML format is available at: <http://imsg.enme.ucalgary.ca/publication/abm.htm>)
13. Shen, W., Norrie, D.H. and Kremer, R. Developing Intelligent Manufacturing Systems Using Collaborative Agents. *IMS'99*, Leuven, Belgium, 1999.
14. Shen, W., Ulieru, M., Norrie, D.H. and Kremer, R. Managing Internet Enabled Supply Chain with CASA. In *Proceedings of Agents'99 Workshop on Agent Based Decision-Support for Managing the Internet-Enabled Supply-Chain*, Seattle, WA, 1999.
15. Swaminathan, J.M., Smith, S.F., and Sadeh, N.M. A Multi-Agent Framework for Supply Chain Dynamics. In *Proc. of NSF Research Planning Workshop on AI & Manufacturing*, Albuquerque, NM, 1996.