Proceedings of the IFIP International Enterprise Modeling Conference, IEMC'99, June 1999, Verdal-Norway.

MULTI-AGENT BASED MODELING FOR INTER-ENTERPRISE INTEGRATION

Z. Deng, R. Davidrajuh, B. Bang, and A. Laksâ
Narvik Institute of Technology • P.O. Box 385 • 8505 Narvik • Norway
Tel: (+47) 769 66000 • Fax: (+47) 769 66810 • E-mail: {zd, rd, bb, ala}@hin.no

ABSTRACT

This paper firstly introduces the concepts of product-alliance-centered inter-integration and virtual manufacturing system. Secondly, a multi-agent based modeling architecture and technology for the modeling and analysis of the formation and operation of virtual manufacturing system is discussed. Then, a developing testing prototype in accordance with the proposed architecture is explained.

1. INTRODUCTION

Computerized enterprise integration (CEI) concept raised in late 1960s. Conventionally, CEI is mostly concerned with *intra*-enterprise integration, i.e. integration of computer-automated islands such as computer-aided design (CAD), computer-aided process planning (CAPP), production planning and control (PP&C), computer-aided quality control (CAQ), and computer-aided manufacturing (CAM) within a given enterprise.

Today, manufacturing enterprises are faced with challenges globally. Increasingly, these challenges cannot be effectively met by isolated effort within a single enterprise. Therefore, CEI is nowadays concerning with not only what happens within a given enterprise (*intra*-enterprise integration), but also what happens among a group of enterprises (*inter*-enterprise integration).

To aid the plan and implementation of new inter-enterprise integration, or to aid the analysis and improvement of the existing inter-enterprise integration operation, naturally we need powerful modeling methodology to realize such purposes.

Inter-enterprise integration can be categorized into *product-alliance-centered* integration (*midcro* integration) and finance-alliance and risk reduction centered integration (macro integration) as shown in Figure 1 (Deng, 1998). This paper focuses on modeling methodology for *product-alliance-centered* inter-enterprise integration (*midcro* integration).

Product-alliance-centered inter-enterprise integration involves in a host enterprise integrated with enterprises in the *supply-chain* and *distribution-chain* for their common products' development, manufacturing, and distribution. The host enterprise is the organizer of the alliance.

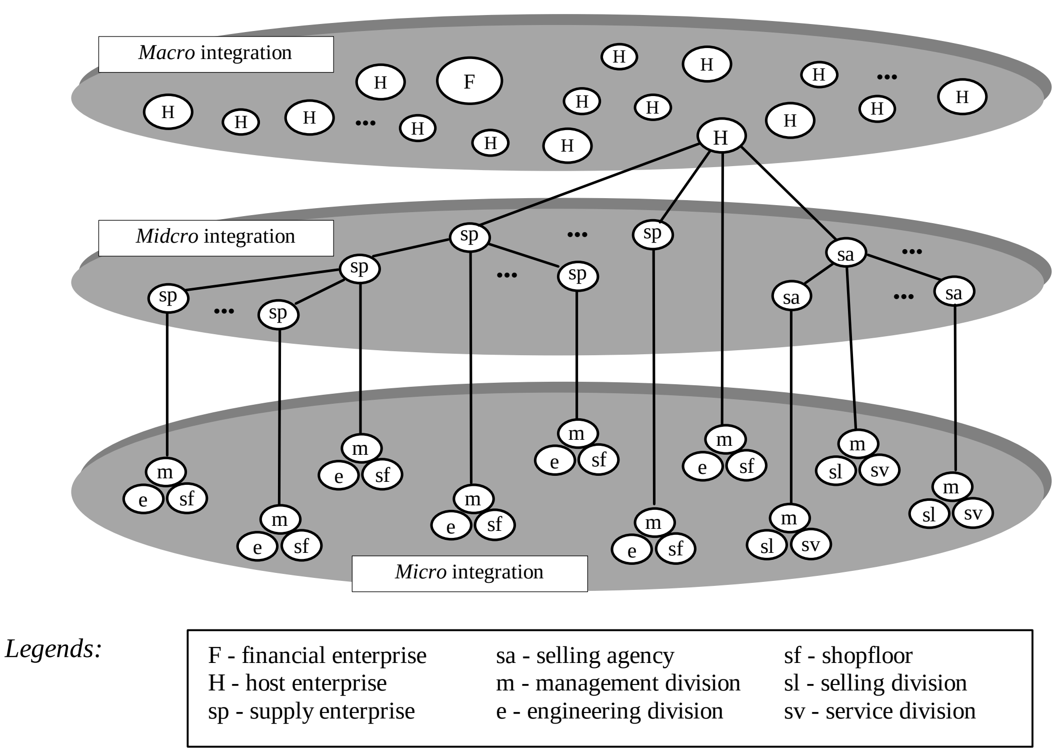

Figure 1: A three level model on enterprise integration

However, when market requirements have changed, new products should be turned out to market. In turn, new combination of enterprises (new supply chain and distribution chain) should be re-organized (see Figure 2). This dynamic organization of manufacturing can be called virtual manufacturing. Then, the dynamically organized manufacturing system is called *virtual manufacturing system* (VMS).

The life cycle of VMS includes phases such as *business opportunity identification*, *partner selection*, *VMS formation*, *VMS operation*, and *VMS reconfiguration* (Enator, 1998).

No matter which enterprise in the “Host / Supply-chain / Distribution-chain” combination, each of them is of autonomous feature and they are geographically distributed. To model such system, multi-agent-modeling concept is a natural option, due to an agent in multi-agent environment characterizes properties of autonomy and distribution (Nwana, 1998 and Laufmann, 1998).

In following sections, firstly we shall describe the architecture of multi-agent based modeling methodology for inter-enterprise integration conceived by the authors, then a testing prototype, which is developed by the authors will be illustrated.

2. ARCHITECTURE OF MULTI-AGENT BASED MODELING METHODOLOGY FOR INTER-ENTERPRISE INTEGRATION

During the operation phase of VMS, the host enterprise coordinates the supply chain and distribution chain as shown in Figures 1 and 2 to obtain very good operational properties for the whole VMS. Herein, the very good properties mean that the VMS can supply customers with enough quantity of products with customer-satisfied quality, price, and response time, and at the same time it can get higher profit. Therefore, as shown in Figure 3, a *VMS coordinator agent* (VCA) is needed for coordinating tasks among different enterprise coordinator agents, which are *host enterprise coordinator agent* (HCA), *supply enterprise coordinator agents* (SCA), and *distribution enterprise coordinator agents* (DCA).

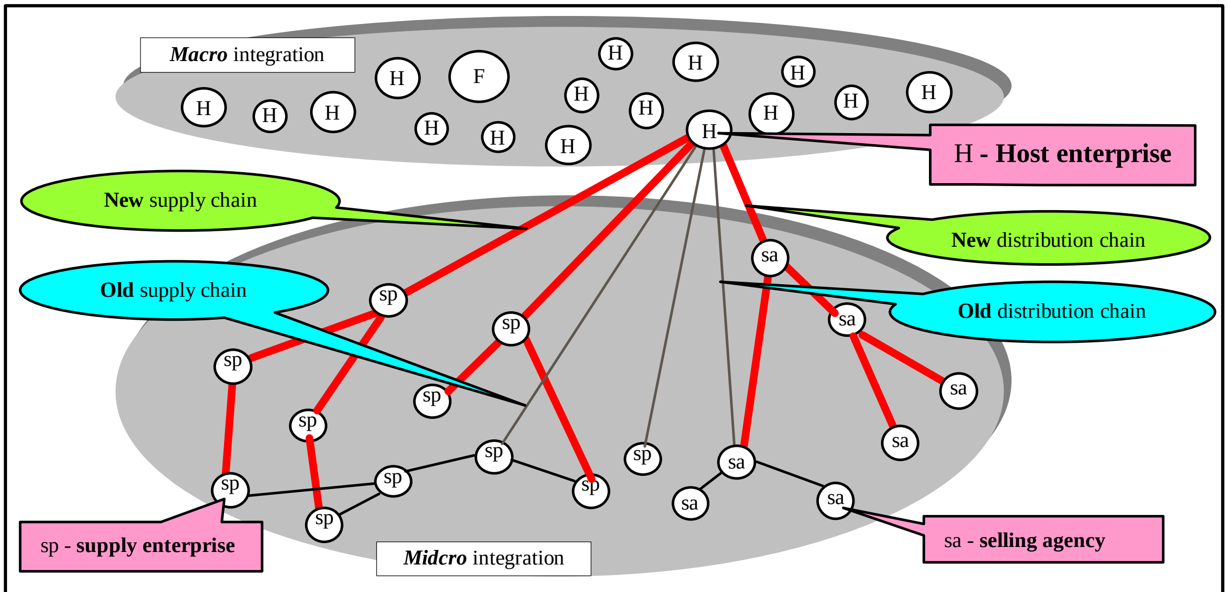

Figure 2: Organization and reorganization of inter-enterprise integration

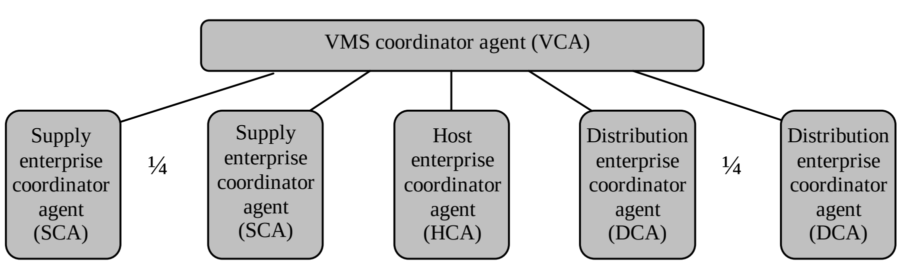

Figure 3: Multi-agent relationship of virtual manufacturing system

The tasks of VMS coordinator agent include 1) identifying right collaborative partners to form a VMS, 2) coordinating material flow through the VMS from supplying sources of raw material, to shipment of final products, 3) striving to get highest profit and to balance the profit among collaborative partners.

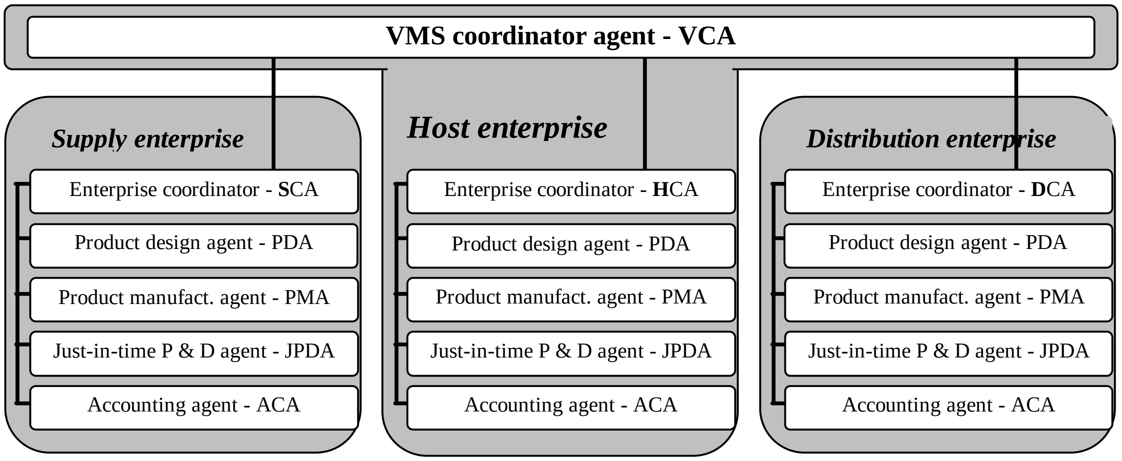

Figure 4: Relationship of sub-agents within enterprise agent

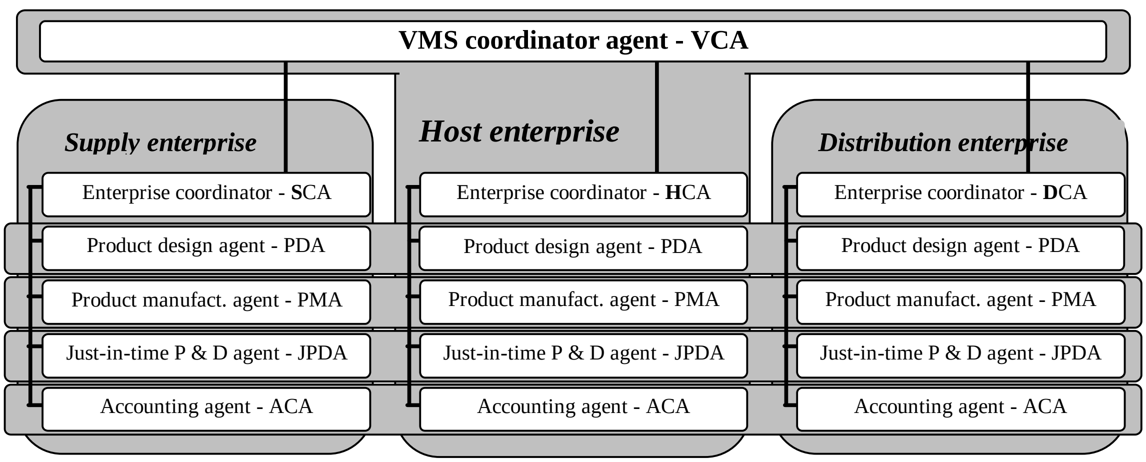

Figure 5: Vertical and horizontal relationship of agents in virtual manufacturing system

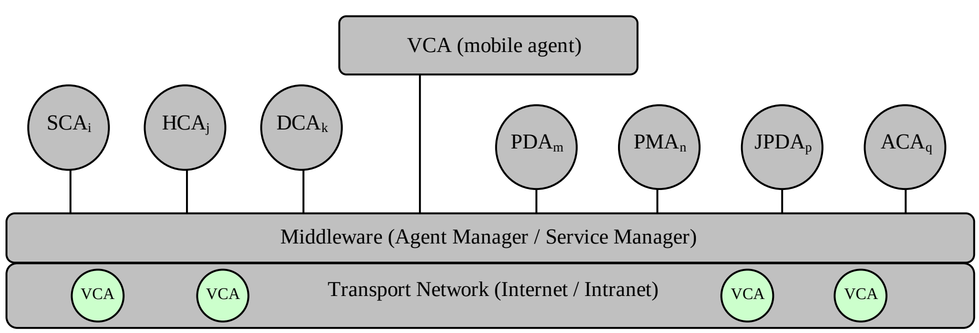

Figure 6: Physical process of agent coordination process supported by middleware and transport network

Enterprise coordinator agent (SCA, HCA, or DCA) has tasks such as:

- Coordinating with VMS coordinator agent to achieve VMS system-wide global objectives and to negotiate with the VMS coordinator agent for its own enterprise local tasks;
- Coordinating internally with its local agents *1)* to develop and design customer keen products; *2)* to manufacture quality-assured and quantity sufficient products; *3)* to procure materials and distribute products just in time; and *4)* to account for lowering cost, giving reasonable price, and getting higher profit for the enterprise.

Thus, we may define sub-agents inside an enterprise as *product design agent* (PDA), *product manufacturing agent* (PMA), *just-in-time procurement and distribution agent* (JPDA), and *accounting agent* (ACA), which are under the coordination of enterprise coordination agent as shown in Figure 4.

In reality, other than *vertically hierarchical relationship* of agents as shown in Figure 4, there exists also *horizontal relationship* among agents in different enterprises. For example, when host enterprise product design agent PDA is designing products, it may coordinate and negotiate, not only with product manufacturing agent (PMA), just-in-time procurement and distribution agent (JPDA), and accounting agent (ACA) inside the same enterprise, but also with PDAs in supply and distribution enterprises. Thus, the *horizontal coordination* among agents can be logically shown in Figure 5.

All agents mentioned above such as coordination agents (VCA, SCA, HCA, and DCA) and functional agents (PDA, PMA, JPDA, and ACA) possess features of distribution, coordination, and negotiation. They can be categorized into classes of mobile agent and static agent. In above-mentioned architecture, only VCA is the mobile agent, others are static agents. Thus, either vertical coordination or horizontal coordination is physically realized via VCA mobile activities.

Figure 6 shows the agents' coordination processes where the VCA mobile agent supported (controlled and managed) by the middleware travels through the transport network (Internet / Intranet) to fulfil coordination task (Davidrajuh, 1999). Whenever a VMS formation or a VMS operation task is launched, the VCA mobile agent travels through the transport network to negotiate and work with static agents until the fulfillment of the task. As well, in case of that a collaborative product-design task is launched, a PDA agent will first coordinate with VCA mobile agent, then via the VCA, which travels through the transport network to collaborate with other relative PDAs to fulfil the design task.

3. A TESTING PROTOTYPE

Based upon above multi-agent modeling architecture, a testing prototype was developed in our laboratory. Current version of our prototype under development is:

1) Based on Windows NT,
2) Based on network protocol TCP/IP, and Communication between agents are similar to that of Java RMI.
3) Tasks of agents described in sections 2 are partially implemented,

The agent prototype is shown in Figure 7. We choose Concordia (Concordia, 1998) as the agent development system upon which we developed agents.

As mentioned above, among the types of agents, only the VCA is mobile, all the other agents are static ones. VCA can be implemented in two different ways: independently launched as a mobile agent to serve the agents HCA, SCA and DCA; or VCA could be launched by HCA which maintains (and controls) activities of VCA. The second alternative is implemented in the prototype.

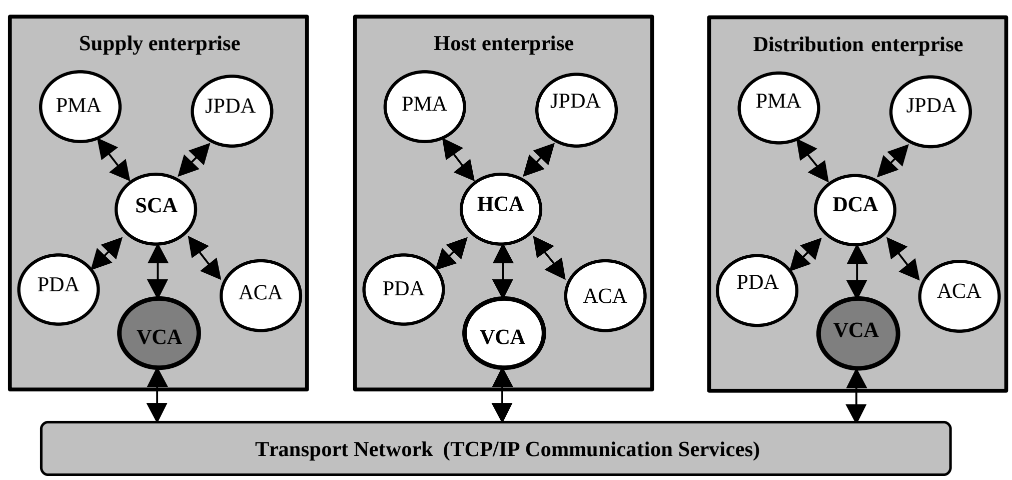

Figure 7: The agent prototype

When VCA travels, its internal state including all member variables persists between locations. Any data that is to travel with VCA is stored within its member variables (objects which implements *Serializable* interface). Therefore horizontal communication among same kind of agents (e.g. PDA-PDA) can be done by placing data in the member variables of VCA which are then taken between the collaborating agents. This technique is simple and is used in this prototype. However, for more efficient horizontal communication, Concordia provides collaboration and distributed event frameworks that are more structured mechanism for allowing agents to interact.

4. CONCLUSION

How to model and analyze inter-enterprise integration is one of the most concerns in nowadays' enterprise modeling circles. Due to the complexity and large scale of virtual manufacturing system, it is difficult to pursue an effective modeling methodology for it. This paper attempts to make use of multi-agent technology for this purpose because the agents possess characteristics of autonomy, coordination, distribution, and mobility. So we can implement independent enterprises and their independent functions as agents to be not only autonomous, but also coordinated. As well, even they are geographically widely distributed, but by means of the mobility of agent, they are communicative.

The architecture proposed in this paper is based on the integration among small and medium sized enterprises. For the large enterprise, the vertical structure of the agents needs to be extended to posses more levels of hierarchy. Because of great working load in agent programming, we have only partially implemented the coordinated agents in our testing prototype. Further works are necessary leading to develop a complete experimental paradigm for exploring the "Plug and Play" inter-enterprise integration to cope with the natural multi-vendor collaboration environment.

5. LITERATURE

Concordia: **Product Information**, 1998, http://web.vsisinc.com/concordia/product_information/what_is_it.htm

Davidrajuh, R., Deng, Z., and Wang, K.: **Realizing Integrated Intelligent Manufacturing System with Distributed Intelligent Agents**, Proceedings of the 15th International Conference on Computer-Aided Production Engineering, April 1999, Durham, UK.

Deng, Z.: **A Model of Methodology Need for AQAL Production System**, Re-engineering for Sustainable Industrial Production, 1997, Chapman & Hall, ISBN 0-412-79950-2.

Deng, Z., Bang, B., Laksa, A., and Nadarajah, S.: **A Model of enterprise integration and collaboration tools and communication infrastructure for inter-enterprise collaboration**, Globallization of Manufacturing in the Digital Communications Era of the 21st Century, 1998, Kluwer Academic Publishers, ISBN 0-412-83540-1.

Enator: **Virtual Enterprising**, 1998, http://195.100.12.162.

Laufmann, S. C.: **Agent Software for Near-Term Success in Distributed Applications**, AGENT TECHNOLOGY - Foundations, Applications, and Markets, 1998, Springer, ISBN 3-540-63591-2.

Nwana, H. S. and Ndumu, D. T.: **A Brief Introduction to Software Agent Technology**, AGENT TECHNOLOGY - Foundations, Applications, and Markets, 1998, Springer, ISBN 3-540-63591-2.