



NRC Publications Archive Archives des publications du CNRC

A hybrid agent-oriented infrastructure for modeling manufacturing enterprises

Shen, W.; Norrie, D. H.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Proceedings of KAW'98: Eleventh Workshop on Knowledge Acquisition, Modeling and Management, 5, pp. 1-19, 1998-04-18

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=61fdbdb6-ac91-4fe7-b26d-e09b751e3614>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=61fdbdb6-ac91-4fe7-b26d-e09b751e3614>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



A Hybrid Agent-Oriented Infrastructure for Modeling Manufacturing Enterprises

Weiming Shen and Douglas H. Norrie
Division of Manufacturing Engineering, The University of Calgary
2500 University Dr. NW, Calgary, AB, Canada T2N 1N4
E-mail: [wshen](mailto:wshen@enme.ucalgary.ca) | [norrie](mailto:norrie@enme.ucalgary.ca)@enme.ucalgary.ca

URL: <http://img.enme.ucalgary.ca/>

Abstract

Manufacturing enterprises are now moving towards open architectures for integrating their activities with those of their suppliers and customers within wide supply chain networks. Traditional knowledge engineering approaches with large scale or very large scale knowledge bases are not suited for such widely distributed systems. Agent-based technology provides a natural way for resolving this problem. This paper presents a hybrid agent-oriented infrastructure for modeling manufacturing enterprises so as to integrate design, planning, scheduling, simulation, execution, material supply, and marketing services into a distributed intelligent open environment. In this paper, we discuss the requirements for next generation of manufacturing enterprises, and describe the main features of the proposed general infrastructure and the functions of its components. A machine-centered dynamic scheduling and rescheduling mechanism is then detailed and a prototype implementation is presented.

[1 Introduction](#)

[2 Requirements for Next Generation of Manufacturing Enterprises](#)

[3 Agent-Oriented Approach for Modeling Manufacturing Enterprises](#)

[3.1 Knowledge Sharing and Reuse](#)

[3.2 Agent-Oriented Approach for Modeling Manufacturing Enterprises](#)

[4 MetaMorph Project](#)

- [4.1 MetaMorph I](#)
- [4.2 MetaMorph II](#)
- [4.3 Characteristics of the MetaMorph II Architecture](#)
- [4.4 Knowledge Sharing and Reuse in MetaMorph II](#)
- [4.5 A MetaMorph II Scenario for a Furniture Fabrication Factory](#)
- [5 Dynamic Scheduling and Rescheduling in MetaMorph II](#)
 - [5.1 Contract Net Protocol](#)
 - [5.2 Asynchronous Communication](#)
 - [5.3 Bidding and Mediation Processes](#)
 - [5.4 Priority and Concurrent \(Parallel\) Scheduling](#)
 - [5.5 Dynamic Rescheduling](#)
 - [5.6 Clustering and Cloning](#)
- [6 Prototype Implementation](#)
- [7 Conclusion and Future Work](#)
- [References](#)

1 Introduction

Manufacturing enterprises are now moving towards open architectures for integrating their activities with those of their suppliers and customers within wide supply chain networks. To compete effectively in today's markets, manufacturers must be able to interact with customers, suppliers, and services rapidly and inexpensively. Traditional knowledge engineering approaches with large scale or very large scale knowledge bases are inappropriate because of the highly distributed nature of the systems. Agent-based technology derived from Distributed Artificial Intelligence provides a natural way for resolving this problem.

At The University of Calgary, we are now working on the MetaMorph II project whose enhanced capabilities will embody lessons learned from our previous research work. This paper describes our ongoing MetaMorph II project and presents its prototype implementation. The rest of this paper is organized as follows: Section 2 discusses the requirements for next generation of manufacturing enterprises; Section 3 introduces agent-based technology for

modeling manufacturing enterprises; Section 4 presents our current working project MetaMorph II; Section 5 describes the dynamic scheduling and rescheduling mechanisms developed for MetaMorph II; Section 6 presents the prototype implementation; Section 7 gives concluding remarks and perspectives.

2 Requirements for Next Generation of Manufacturing Enterprises

Manufacturing strategy has shifted rapidly over the past ten years to support global competitiveness, new product innovation and introduction, and rapid market responsiveness. The next generation of manufacturing systems will be time oriented versus cost or even quality based. Such manufacturing systems should meet following fundamental requirements:

Enterprise Integration

In order to support its global competitiveness and rapid market responsiveness, an individual manufacturing enterprise has to be integrated with its related management systems (e.g., purchasing, orders, design, production, planning, control, transport, resources, personnel, materials, quality, etc.) which are, in general, heterogeneous software and hardware environments. Such integration may be realized via tactical planning systems that rely heavily on distributed knowledge-based systems to link demand management directly to resource and capacity planning.

Cooperation

Manufacturing enterprises have to fully cooperate with their suppliers and customers for material supply, parts fabrication, final product commercialization, and so on. Such cooperation should be in an efficient and quick-response manner.

In a cooperative system, dynamic chains of events are embedded in concurrent information processes. Requirements imposed by customer orders, managerial decisions, and design stages are integrated with the production planning and resource allocation tasks in a complex framework that incorporates high-level decisions into the planning activities. This is essentially a cooperative, concurrent information-processing environment. Cooperation is an imperative requirement for any complete functional model for advanced manufacturing systems.

Integration of humans with software and hardware

People and computers need to be integrated to work collectively at various stages of the product development, with access to required knowledge and information. Heterogeneous sources of information must be integrated to support these needs and enhance the decision capabilities of the system. Bi-directional communication environments are required to allow effective, quick communication between human and computers to facilitate their cooperation.

Agility

Economic globalization and expanding market expectations are rapidly transforming the environment for manufacturing. Considerable attention must be given to reducing product cycles to be able to respond more quickly to customer desires. In this new scale of economic transformation, corporations are progressively reorienting their strategies to expand their share of the market and to integrate "Agile" manufacturing into their production facilities.

Agile manufacturing is the ability to adapt in a manufacturing environment of continuous and unanticipated change and thus is a key component in manufacturing strategies for global competition. To achieve agility, manufacturing facilities must be able to establish convenient associations with heterogeneous partners. Ideally, partners are contracted with "on the fly" only for the time required to complete specific tasks. This type of interaction can also be used to plan long-term strategies. Agility will bring greater flexibility to the manufacturing organization without incurring large or diverted industrial investments

Scalability

Scalability is an important property for advanced manufacturing systems. Scalability means that more resources can be incorporated into the organization as required. This property should be available at any working node in the system and at any level within the nodes. Expansion of resources should be possible without disrupting organizational links previously established.

To identify and incorporate new components into the system, organizational knowledge registries are required. When new physical components arrive in the system, representative entities are created to act as counterparts to the

components throughout their life cycles. The ability to add new components incrementally allows the system to respond flexibly to a wide variety of requests. For example, the system might dynamically add increased intelligence and manufacturing capacity to supply a rapidly expanding market or reduce capacity to adjust downwards during low demand periods. When physical components are removed from the system for maintenance or other reasons, the listing of these components is removed from the system registry. Robust registration mechanisms are needed to provide ongoing integration of new components or the removal of existing ones

Dynamic reconfiguration

Both human beings and artificial entities in manufacturing systems need to be more alert to environmental changes. Every stage in manufacturing planning is affected by dynamic variations coming from either internal or external sources. In conventional manufacturing systems, the input from customers triggers a sequence of events, starting with planning operations. At this level, requests are processed according to preestablished stages (which include specification of product design, material management, manufacturing capacity planning and availability, and preparation of production costs). The planning process also triggers requirements for subcontracting external services.

Conventionally, the planning process progressively advances through a series of sequential evaluations that correspond to system conditions from an earlier time. Therefore, any subsequent variations in the state of the environment can make these plans invalid. This type of sequential system thus becomes expensive, since there is a tendency to reuse computing resources for redundant and repetitive evaluations.

Eventually, there is a transition from the planning process to a second major area of manufacturing control activity in which manufacturing plans and tasks are allocated execution times. More variations are introduced at this stage, which affect the stability of the system and its ability to execute plans according to schedule. This complex stage restricts the ability of the system to reconfigure to cope with dynamic and unforeseen changes.

Expandability is possible in such conventional manufacturing systems, but it requires major reconfiguration of the system. Tightly coupled interconnections among the system's existing components make adjusting each processing module to new component availability very tedious. Similarly, the removal of components implies a considerable readjustment of the system.

Knowledge capitalization and distribution

The efficient capture and distribution of knowledge pertaining to each aspect of the organization - finance, marketing, design, and manufacturing - coupled with its effective use, will result in startling advances in market research, product and process development, production planning and scheduling, and ultimately customer responsiveness.

The more obvious problems in information processing observed in conventional manufacturing systems are highly centralized information management and production control, which corresponds to the need to maintain an overall system view in order to minimize costs (and so to win over more of the market). Centralized databases are commonly used to accumulate system information for establishing production plans and forecasting future requirements. Powerful centralized computers process large amounts of data to create production plans and schedules. Transactions among various resources are also forced to pass through a centralized control unit. All activities in conventional manufacturing systems are limited by the accuracy and stability of the centralized processing components, yielding a fragile infrastructure.

Although centralized and sequential information-processing systems have in the past minimized hardware and software costs, their central structure is not suited to the inherent distributed nature of concurrent information flow in agile manufacturing.

Distribution of production knowledge will enhance system modularity and facilitate both integration and reconfiguration. The increased modularity reduces the complexity of organizing knowledge by maintaining knowledge locally. Information held locally can be processed concurrently, thus avoiding the limitations of sequential information processing

Concurrent Engineering

Ensuring the manufacturability of the product constitutes the first step in implementing concurrent engineering. Geometric and functional specifications, availability of raw materials, and the capability and availability of shop-floor resources each has a major influence on manufacturability. A design may be manufacturable under one combination of product requirements and shop-floor resources, but not under another. The selection and availability of stock material

from which the part will be manufactured influences the number of intermediate steps required, and hence the production cost. The capability and availability of shop-floor resources impact the process plan to be used, and again the production cost. Thus, all of these aspects need to be considered simultaneously for effective concurrent engineering.

3 Agent-Oriented Approach for Modeling Manufacturing Enterprises

3.1 Knowledge Sharing and Reuse

From the perspective of knowledge intensive engineering, we can view all relevant aspects of an organization domain in terms of 'knowledge'. This applies to the structure and nature of the organization itself, the data used within different components of the organization and the flow of this data through the organization along with the value added to it during the execution of the organizational tasks. This knowledge exists in the form of data (factual assertions about the organization and its tasks) which may reside in existing information systems (such as databases), and in the form of specific 'business rules' applied to this data in order to carry out some function within the organization, either relating directly or indirectly to the tasks at hand. The focus of knowledge systems techniques is the explicit representation of this organizational knowledge in a form that is optimal for effective reasoning about the tasks in the organization, as well as for the representation of this information to those assigned the execution of line tasks ([Davis and Oliff 1988](#)).

Previously, large scale or very large scale knowledge bases have been often advocated for engineering applications including design, manufacturing, operations, and maintenance, because these activities require an extremely huge amounts of and various kinds of knowledge ([Forbus 1988](#)). But, according to Tomiyama et al ([1995](#)), not only the quantity of knowledge but also the quality of knowledge in terms of sharability and reusability of knowledge is crucial. Knowledge intensive engineering aims at both the amount and flexibility of knowledge. A single knowledge base can make inferences in a particular circumstance but it may hard-fail. Therefore just having a large scale or very large scale knowledge base alone is not enough for modern manufacturing whose robustness and reliability are actually important.

Knowledge in modern manufacturing must be well organized and should be able to be flexibly applied to different kinds of applications. Figure 1 compares three possible different types of knowledge sharing architectures ([Tomiyama et al 1995](#)). Figure 1(a) depicts a situation with independent knowledge bases. In this case, the 'strength' of knowledge

is just a sum of each of independent knowledge bases. Integrated knowledge bases can be represented as in Figure 1(b). Here, the knowledge bases can be applied to various situations and the 'strength' of knowledge is near maximum. However, this requires having a platform with a uniform language. The Cyc project ([Lenat and Guha 1989](#)) is an example of this approach. In Figure 1(c), independent knowledge bases can communicate and form an interoperable situation, although the 'strength' of knowledge might be weaker than that in Figure 1(b). The entire knowledge base is a federation or a set of loosely coupled intelligent agents. This approach has recently been used by projects like SHADE ([McGuire et al 1993](#)), PACT ([Cutkosky et al 1993](#)), DESIRE ([Brazier and Treur 1996](#)), and DIDE ([Shen and Barth 1997](#)).



Figure 1. Knowledge sharing architectures ([Tomiya et al 1995](#))

The second architecture (Figure 1(b)) may suffer from the lack of uniform knowledge representation. Unless carefully designed, the platform language cannot cover everything. The third architecture (Figure 1(c)) may overcome this problem, if the framework is abstract enough to incorporate the different types of ontology each agent uses. However, it does not completely avoid the problem, because communication among agents requires at least understanding what other agents are talking about.

The same communication requirements justify the representation model used in SHADE ([McGuire et al 1993](#)). The model, called KIF (Knowledge Interchange Format) ([Genesereth & Fikes 1992](#)), is a machine-readable version of first order predicate calculus, with extensions to enhance expressiveness. KIF specifications define syntax and semantics; ontology defines the problem-specific vocabulary. Agents exchange sentences in KIF using the shared vocabulary.

To support the sharing and reuse of formally represented knowledge among AI systems, it is useful to define the common vocabulary in which shared knowledge is represented ([Patil et al 1992](#)). A specification of a representational vocabulary for a shared domain of discourse definitions of classes, relations, functions, and other objects is called an *ontology* ([Gruber 1993](#)). The need for a shared ontology is a direct result of the multidisciplinary nature of engineering. There are many different views of a design (function, performance, manufacturing), each with a different language. However, the various perspectives typically overlap, necessitating the sharing of information if design is to proceed concurrently and cooperatively. For information to be shared, there must be a commonly understood vocabulary. A detailed discussion on ontology can be found in ([Gruber 1993](#)). An application of ontology in enterprise modeling was proposed by Fox et al ([1996](#)).

In design applications, it is necessary to represent knowledge at several levels: domain knowledge associated with the particular vocabulary used in the design domain, but also general engineering knowledge (e.g., associated with the use of engineering units). The particular representation of the corresponding ontology requires special models, mechanisms, languages, and tools.

In distributed systems, it must be possible to exchange knowledge among agents, even if agents work in different specialties. The required formalisms for exchanging knowledge have been studied in various projects, and several proposals exist, such as the knowledge representation languages like KIF ([Genesereth & Fikes 1992](#)), in connection with ontologies. The exchange of knowledge in most existing systems, however, is carried out among human designers, using electronic mail.

3.2 Agent-Oriented Approach for Modeling Manufacturing Enterprises

The requirements described in Section 2 necessitate decentralized manufacturing facilities whose design, implementation, reconfiguration, and manufacturability allows the integration of production stages in a dynamic, collaborative network. Such facilities can be realized through agent-oriented approaches ([Wooldridge and Jennings 1995](#)) using knowledge sharing technology ([Patil et al 1992](#)). The following paragraphs briefly review several interesting projects in this domain.

The SHADE project ([McGuire et al 1993](#)) was primarily concerned with the information-sharing aspect of concurrent engineering. Rather than attempting to model the design process, it provides a flexible infrastructure for anticipated knowledge-based, machine-mediated collaboration among disparate engineering tools. SHADE is distinct from other approaches in its emphasis on a distributed approach to engineering knowledge rather than a centralized model or knowledge base. That is, not only does SHADE avoid the requirement of physically centralized knowledge, but the modeling vocabulary is distributed as well, focusing knowledge representation on specific knowledge-sharing needs.

PACT ([Cutkosky et al 1993](#)) was a landmark demonstration of both collaborative research efforts and agent-based technology. The agent interaction relies on shared concepts and terminology for communicating knowledge across disciplines, an *interlingua* for transferring knowledge among agents, and a communication and control language that enables agents to request information and services. This technology allows agents working on different aspects of a design to interact at the knowledge level, sharing and exchanging information about the design independent of the format in which the information is encoded internally.

SHARE ([Toye et al 1993](#)) was concerned with developing open, heterogeneous, network-oriented environments for concurrent engineering. It used a wide range of information-exchange technologies to help engineers and designers collaborate in mechanical domains.

Recently, PACT has been replaced by PACE (Palo Alto Collaborative Environment) [<http://cdr.stanford.edu/PACE/>] and SHARE by DSC (Design Space Colonization) [<http://cdr.stanford.edu/DSC/>].

FIRST-LINK ([Park et al 1994](#)) was a system of semi-autonomous agents helping specialists to work on one aspect of the design problem. NEXT-LINK ([Petrie et al 1994](#)) was a continuation of the FIRST-LINK project for testing agent coordination. Process-Link ([Goldmann 1996](#)) followed on from Next-Link and provides for the integration, coordination, and project management of distributed interacting CAD tools and services in a large project.

MADEFast ([Cutkosky et al 1996](#)) is a DARPA DSO-sponsored project to demonstrate technologies developed under the ARPA MADE (Manufacturing Automation and Design Engineering) program. MADE is a DARPA DSO long term program for developing tools and technologies to provide cognitive support to the designer and allow an order of magnitude increase in the explored alternatives in half the time it takes to explore a single alternative today.

SiFA ([Brown et al 1995](#)), developed at Worcester Polytechnic, is intended to address the issues of patterns of interaction, communication, and conflict resolution. DIDE proposed to use autonomous cognitive agents for developing distributed intelligent design environments ([Shen and Barth's 1997](#)). In AARIA ([Parunak et al 1997a](#)), the manufacturing capabilities (e.g. people, machines, and parts) are encapsulated as autonomous agents. Each agent seamlessly interoperates with other agents in and outside its own factory. AARIA uses a mixture of heuristic scheduling techniques: forward/backward scheduling, simulation scheduling, and intelligent scheduling. Scheduling is performed by job, by resource, and by operation. Scheduling decisions are made to minimize costs over time and production quantities.

Saad et al ([1995](#)) proposed a Production Reservation approach by using a bidding mechanism based on the Contract Net protocol to generate the production plan and schedule. Maturana et al ([1996](#)) described an integrated planning-and-scheduling approach combining subtasking and virtual clustering of agents with a modified Contract Net protocol. RAPPID (Responsible Agents for Product-Process Integrated Design) ([Parunak et al 1997b](#)) at the Industrial Technology Institute was intended to develop agent-based software tools and methods for using market place dynamics among members of a distributed design team to coordinate set-based design of a discrete manufactured product. AIMS ([Park et al 1993](#)) was envisioned to provide to the US an integrated industrial base able to rapidly respond, with highly customized solutions, to customer requirements of any magnitude, thus reinstating the US as the world leader in manufacturing. ADDYMS (Architecture for Distributed Dynamic Manufacturing Scheduling) proposed by Butler and Ohtsubo ([1992](#)) was a distributed architecture for dynamic scheduling in a manufacturing environment. Pan and Tenenbaum ([1991](#)) proposed a software Intelligent Agent (IA) framework for integrating people and computer systems in large, geographically dispersed manufacturing enterprises. This framework is based on the vision of a very large number (e.g. 10 000) computerized assistants, known as Intelligent Agents (IAs). Human participants are encapsulated as Personal Assistants (PAs), a special type of IA.

Roboam and Fox ([1992](#)) proposed an Enterprise Management Network (EMN) to support the integration of activities of the manufacturing enterprise throughout the production life cycle with six levels: (1) Network Layer provides for the definition of the network structure; (2) Data Layer provides for inter-node queries; (3) Information Layer provides for invisible access to information spread throughout the EMN; (4) Organization Layer provides the primitives and elements for distributed problem solving; (5) Coordination Layer provides protocols for coordinating the activities of EMN nodes; and (6) Market Layer provides protocols for coordinating organizations in a market environment.

4 MetaMorph Project

4.1 MetaMorph I

MetaMorph (now referred to MetaMorph I) ([Maturana and Norrie 1996](#)) is a multi-agent architecture for intelligent manufacturing developed at The University of Calgary. The architecture has been named MetaMorphic, since a primary characteristic is its changing form, structure, and activity as it dynamically adapts to emerging tasks and changing environment.

In this particular type of federation organization, intelligent agents can link with mediator agents to find other agents in the environment. Additionally, mediator agents assume the role of system coordinators by promoting cooperation among intelligent agents and learning from the agents' behavior. Mediator agents provide system associations without interfering with lower-level decisions unless critical situations occur. Mediator agents are able to expand their coordination capabilities to include mediation behaviors, which may be focused upon high-level policies to break the decision deadlocks. Mediation actions are performance-directed behaviors.

Mediator agents can use brokering and recruiting communication mechanisms ([Decker 1995](#)) to find related agents for establishing collaborative subsystems (also called coordination clusters or virtual clusters) (see Figure 2). The brokering mechanism consists of receiving a request message from an intelligent agent, understanding the request, finding suitable receptors for the message, and broadcasting the message to the selected group of agents. The recruiting mechanism is a superset of the brokering mechanism, since it uses the brokering mechanism to match agents. However, once appropriate agents have been found, these agents can be directly linked. The mediator agent then can step out of the scene to let the agents proceed with the communication themselves. Both mechanisms have been used in MetaMorph I. To efficiently use these mechanisms, mediator agents need to have sufficient organizational knowledge to match agent requests with needed resources. Organizational knowledge at the mediator level is basically a list of agent-to-agent relationships that is dynamically enlarged.

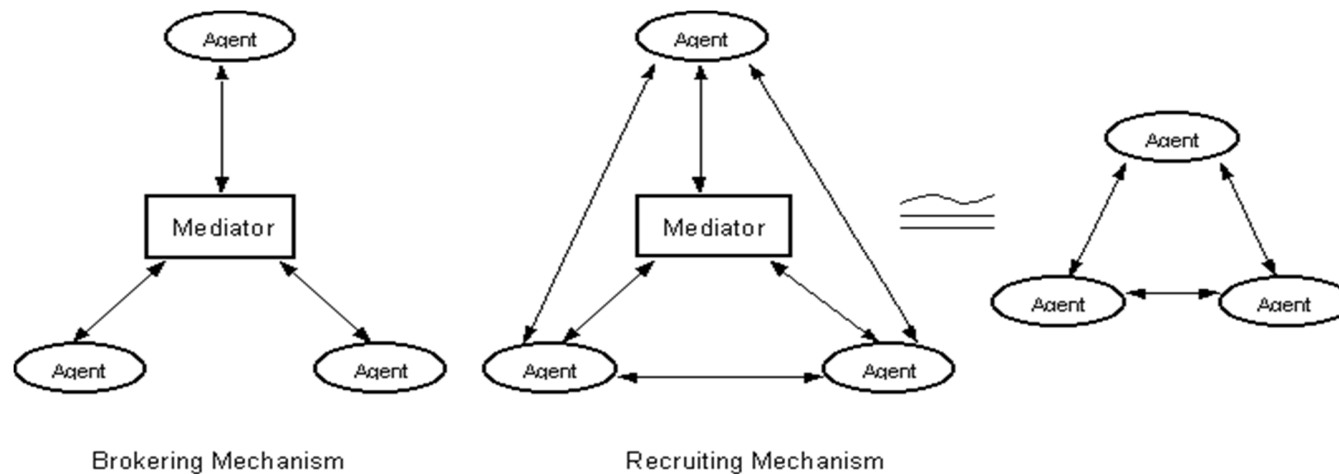


Figure 2. Brokering and Recruiting Mechanisms

The brokering and recruiting mechanisms generate two relevant types of collaboration subsystems. The first corresponds to an indirect collaboration subgroup, since the requester agent does not need to know about the existence of other agents that temporarily match the queries. The second type is a direct collaboration subgroup, since the requester agent is informed about the presence and physical location of matching agents to continue with direct communication.

One common activity for mediator agents involved in either type of collaboration is interpreting messages, decomposing tasks, and providing processing times for every new subtask. These capabilities make mediator agents very important elements in achieving the integration of dissimilar intelligent agents. Federation multi-agent architectures require a substantial commitment to supporting intelligent agent interoperability through mediator agents.

In MetaMorph I ([Maturana and Norrie 1996](#)), mediators were used in a distributed decision-making support system for coordinating the activities of a multi-agent system. This coordination involves three main phases: (1) subtasking; (2) creation of virtual communities of agents (coordination clusters); and (3) execution of the processes imposed by the tasks. These phases are developed within the coordination clusters by distributed mediator agents together with other

agents representing the physical devices. The coordination clusters are initialized through mediator agents, which can dynamically find and incorporate those other agents that can contribute to the task.

Another related project named ABCDE (Agent-Based Concurrent Design Environment) was developed in the same research group ([Balasubramanian et al 1996](#)). It was integrated with the MetaMorph I system through a Design Mediator. The ABCDE architecture includes an environment manager, feature agents, part agents, and CAD physical layers. ABCDE agents interact with the shop-floor mediator to obtain manufacturability assessments during the product design process. Human production managers may request manufacturability evaluations using either the CAD system or the design system interface.

4.2 MetaMorph II

Based on MetaMorph I, the MetaMorph II project started at the beginning of 1997. Its objective is to integrate design, planning, scheduling, simulation, execution, material supply, and marketing services into a distributed intelligent open environment. For this purpose, we propose a hybrid architecture combining and extending the architectures used in our previous projects MetaMorph I ([Maturana and Norrie 1996](#)), ABCDE ([Balasubramanian et al 1996](#)) and DIDE ([Shen and Barth♦s 1995](#)). In this hybrid architecture, the system is primarily organized at the highest level through 'subsystem' mediators (see Figure 3). Each subsystem is connected (integrated) to the system through a special mediator. Each subsystem itself can be an agent-based system (e.g., agent-based manufacturing scheduling system), or any other type of system like functional design system, knowledge-based material management system, and so on. Agents in a subsystem may also be autonomous agents at the subsystem level. Some of these agents may also be able to communicate directly with other subsystems or the agents in other subsystems.

Manufacturing resource agents are coordinated by dynamically hierarchical mediators. For example, a shop floor Resource Mediator coordinates Machine Mediators, Tool Mediators, Worker Mediators, and so on (see Figure 4). A machine agent can also communicate and negotiate directly with a Worker Mediator, worker agents, a Tool Mediator and tool agents.

MetaMorph II Infrastructure

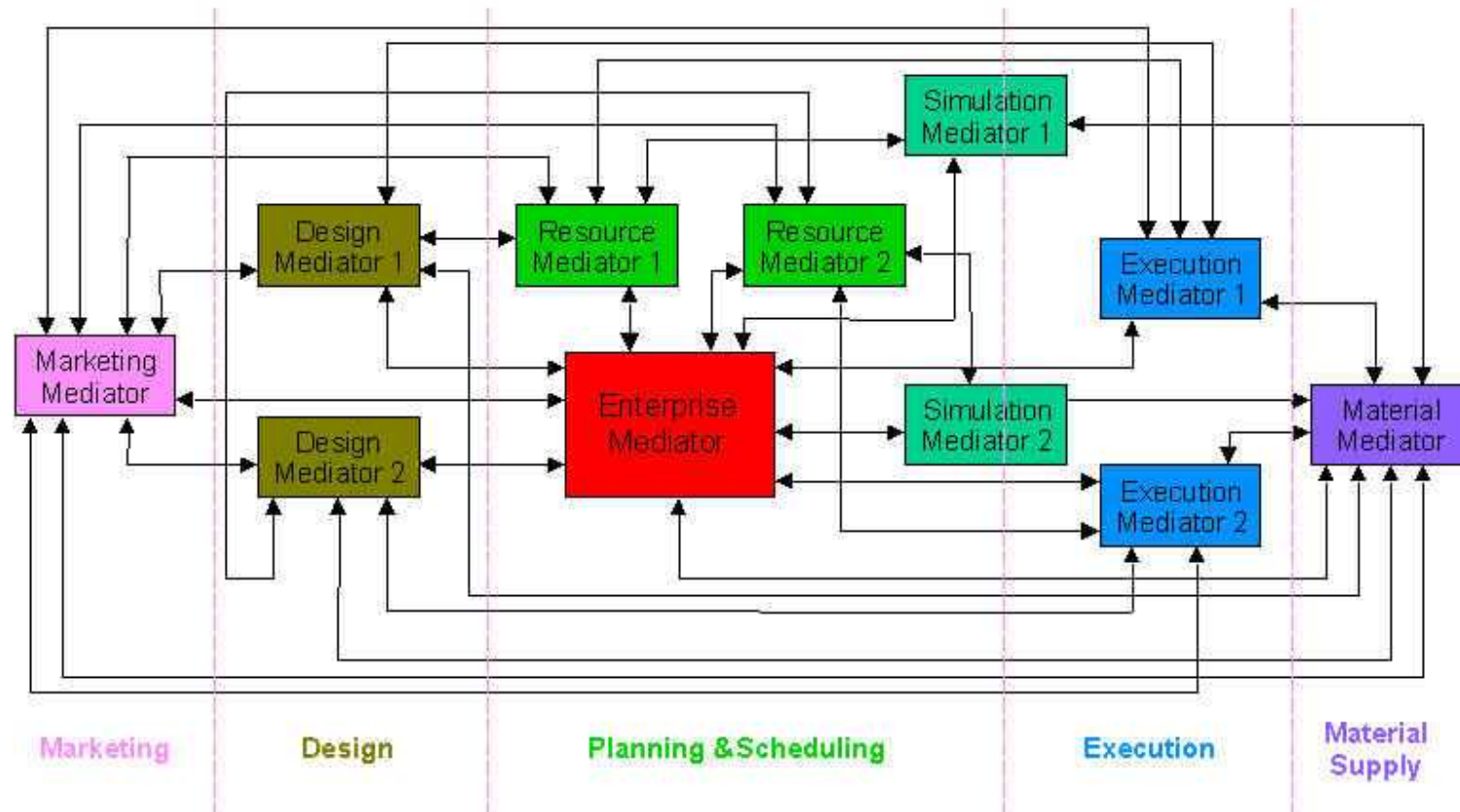


Figure 3. Functional Architecture of MetaMorph II

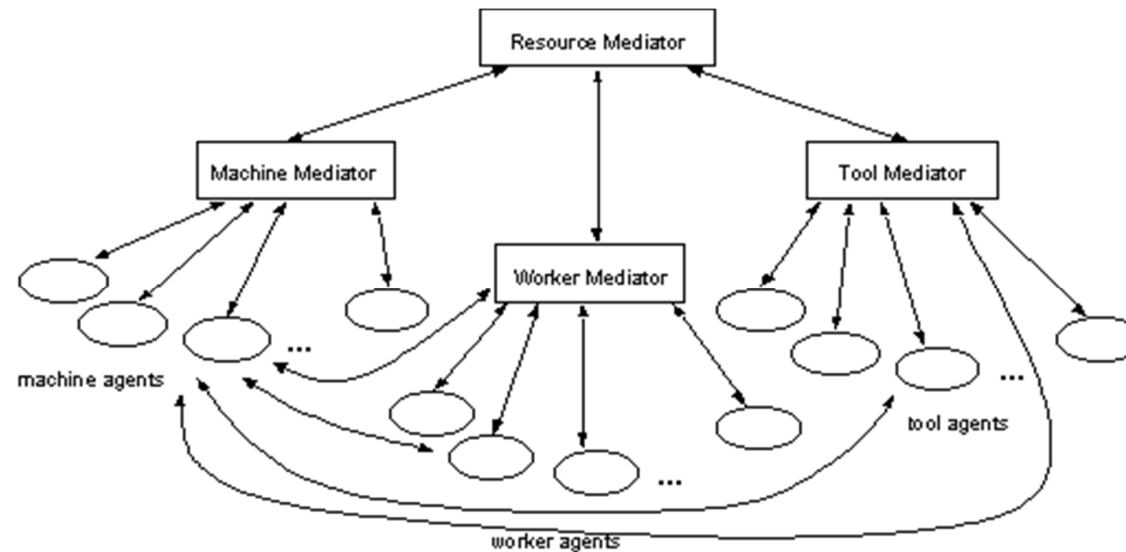


Figure 4. Organization of resource agents

4.3 Characteristics of the MetaMorph II Architecture

MetaMorph II is an extension of MetaMorph I in multiple dimensions (cf. Figure 3):

- a. *Integration of Design and Manufacturing*: Agent-based intelligent design systems will be integrated into the MetaMorph II. Some features and mechanisms used in the DIDE project and ABCDE project will be applied to develop this type of subsystem. Each such subsystem connects within MetaMorph II with a Design Mediator which serves as the coordinator of this subsystem and its only interface to the whole system. Several design systems can be connected to MetaMorph II simultaneously. Each design system may be an agent-based system or another type of design system.
- b. *Extension to marketing*: This will be realized by several easy-to-use interfaces for marketing engineers and end customers to request product information (performance, price, manufacturing period, etc), select a product, request modifications to a particular specification of a product, and send feedback to the enterprise.

- c. *Integration of Material Supply and Management System*: A Material Mediator will be developed to coordinate a special subsystem for material handling, supply, stock management etc.
- d. *Improvement of the Simulation System*: Simulation Mediators will be developed to carry out production simulation and forecasting. Each Simulation Mediator corresponds to one Resource Mediator and therefore to one shop floor.
- e. *Extension to Execution Control*: Execution Mediators will be developed to coordinate the execution of the machines, transportation AGVs, and workers as necessary. Each shop floor is in general assigned with one Execution Mediator.

The implementation environment of the MetaMorph II can be heterogeneous due to its TCP/IP communication, which has been tested under MetaMorph I on a network of HP workstations, SUN workstations and micro-computers. It should be noted that a real implementation in a particular manufacturing enterprise does not need all components of this architecture and some mechanisms may also be simplified according to the real situation of the enterprise.

4.4 Knowledge Sharing and Reuse in MetaMorph II

Our approach to knowledge sharing and reuse in manufacturing systems is to model manufacturing enterprises using agent-based technology with the mediator-centric federation architecture ([Wiederhold 1992](#); [Gaines et al 1995](#)). Two knowledge sharing mechanisms are employed in MetaMorph architecture at two levels. At the higher level (system level), the architecture as shown in Figure 1(c) is used, i.e., each mediator has its independent knowledge base which cooperate with other knowledge bases of the other mediators. The lower level (mediator level) uses the architecture as shown in Figure 1(b), i.e., agents coordinated by a mediator (e.g., resource agents coordinated by a Resource Mediator) share an integrated knowledge base.

4.5 A MetaMorph II Scenario for a Furniture Fabrication Factory

Consider a customer wanting to buy a set of furniture. He or she comes to a customer service office of a furniture fabrication factory which is equipped with the MetaMorph II system. The customer may use the special 3-D graphical interface of the Marketing Mediator of the MetaMorph II to look at items of furniture from various directions under selected color light, navigate and compare among different sets of furniture. He or she may request answers to

questions about a desired or customized set of furniture concerning performance, price, manufacturing period etc. The customer is also allowed to propose several specifications for this special set of furniture, and again ask for price and manufacturing time. The MetaMorph II system will reply to the customer's questions immediately if the information exists in its database system. If the information cannot be found, MetaMorph II will start a product planning procedure for obtaining manufacturing time and an estimated price for this special product, and then send a message to the Marketing Mediator to answer the customer's requests.

If the customer is not satisfied with the fabrication period, he or she may be allowed to request a special delivery date for the special product. MetaMorph II will receive this request and start a re-planning procedure to negotiate with other products in progress, to try to find a possible solution for product delivery within the requested period, and then send a message to the Marketing Mediator with a positive (or negative) response and an estimated price for this special order.

5 Dynamic Scheduling and Rescheduling in MetaMorph II

Cooperative negotiation among resource agents in MetaMorph II is realized by combining the mediation mechanism based on dynamically hierarchical mediators with the bidding mechanism based on the Contract Net protocol ([Smith 1980](#)) for generating and maintaining dynamically production schedules. Such cooperative negotiation may be invoked through either of two scheduling mechanisms according to the manufacturing enterprise condition: Machine-Centered Scheduling and Worker-Centered Scheduling.

With Machine-Centered Scheduling, the Resource Mediator sends a request message to Machine Mediator at first. It is the machine agents that take the responsibility to negotiate with worker agents and tools agents. With Worker-Centered Scheduling, the Resource Mediator sends a request message to the Worker Mediator first. It is the worker agents that take the responsibility to negotiate with machine agents that in turn negotiate with tool agents. The scheduling system can be shifted between these two mechanisms according to the manufacturing enterprise situation. In this paper, we discuss only the Machine-Centered Scheduling mechanism.

In MetaMorph II, parts are also modeled as agents \setminus part agents. Part agents are created and coordinated by Design Mediators. In the current implementation, part agents do not communicate directly with resource agents. All

manufacturing task requests are sent to the Resource Mediators. Therefore, we do not discuss here the communication and negotiation between part agents and resource agents.

5.1 Contract Net Protocol

The Contract net is a negotiation protocol proposed by Smith ([1980](#)) which facilitates distributing subtasks among various agents. The agent wanting to solve the problem broadcasts a call for bids, waits for a reply for some length of time, and then awards a contract to the best offer(s) according to its selection criteria. This protocol has been widely used for multi-agent negotiation. Some researchers have proposed modified versions of Contract Net protocol for special applications, such as the Extended Contract Net Protocol (ECNP) proposed by Fischer et al ([1995](#)), and the B-Contract-Net by Scalabrin ([1996](#)).

The Contract Net protocol used in our implementation has been extended to meet our special requirements. For example, in the bidding and mediation process for selecting a machine for a manufacturing task, after receiving the different propositions (bids) from different machine agents, the Machine Mediator will select a machine agent to perform the task according to its criteria (mainly manufacturing cost criteria) and award a contract to it. Other machines are not selected to perform the task, but they are taken as alternatives which may be contacted (negotiated with) in the future in the case of unforeseen situations such as machine breakdown. This will greatly reduce the rescheduling time when such unforeseen situations take place. The information about the alternative resources (machines, workers and tools) will be sent to the part agents with the information about the selected resources. The part agents save these two sorts of information in their knowledge bases. When the selected resources cannot perform the scheduled tasks due to unforeseen situations, the part agent may negotiate directly with alternative resource agents.

5.2 Asynchronous Communication

The mediator-mediator and mediator-agent communications are asynchronous, and the communication mode can be point-to-point (between two agents), broadcast (one to all agents), or multicast (to a selected group of agents). In the current implementation, we use the multicast mode which can be extended to point-to-point and broadcast. As in the DIDE project ([Shen and Barth 1995; Shen and Barth 1997](#)), five types of messages are used: *request*, *inform*,

announce, *bid* and *notice*. They can be grouped into two categories: requests and assertions (*request*, *inform*, *notice*); call for bids and offers (*announce*, *bid*). The messages are formatted in an extended KQML format.

In order to realize the asynchronous communication, each mediator and resource agent has an input message box. All incoming messages are firstly queued in this input message box. These incoming messages are treated according their priority. The message with highest priority is treated first. For the messages with same priority, the first arriving message is treated first.

5.3 Bidding and Mediation Processes

At the Machine Mediator Level

As shown in Figure 5, when the Machine Mediator receive a task *request* from the Resource Mediator, it sends out multicast *announce* messages to concerned machine agents according to its knowledge about the machine agents registered with it. After receiving an announce message, a machine agent calculates processing time and cost, verifies its reservation plan, and negotiates with worker agents through the Worker Mediator and tool agents through the Tool Mediator (described in the following paragraphs). After receiving the propositions (bids) from the Worker Mediator and the Tool Mediator, the machine agent sends a *bid* with the information about the processing time and cost, proposed start time and end time, other free time slots, and the information about the associated worker agent and tool agent to the Machine Mediator.

Waiting for a predefined period after sending out the *announce* messages to machine agents, the Machine Mediator analyses the propositions (bids) from different machine agents, and selects one machine agent and its associated worker agent and tool agent(s) to perform the task according its selection criteria based on the cost model (detailed in [Maturana et al 1996](#)). As soon as the decision is made, the Machine Mediator sends an award message (*notice* type) to the selected machine agent which then locks a time slot and reserves it for the requested production task. This time slot now becomes unavailable to other tasks. At the same time, the Machine Mediator sends an *inform* message with the information about the selected resources and the alternative resources directly to the part agent (not through the Resource Mediator). After receiving the award message, the winning machine agent should also send an award

message to the winning worker agent and the winning tool agent(s). A more complex mediation mechanism is to be developed to optimize the scheduling for a group of parts or for a given period.

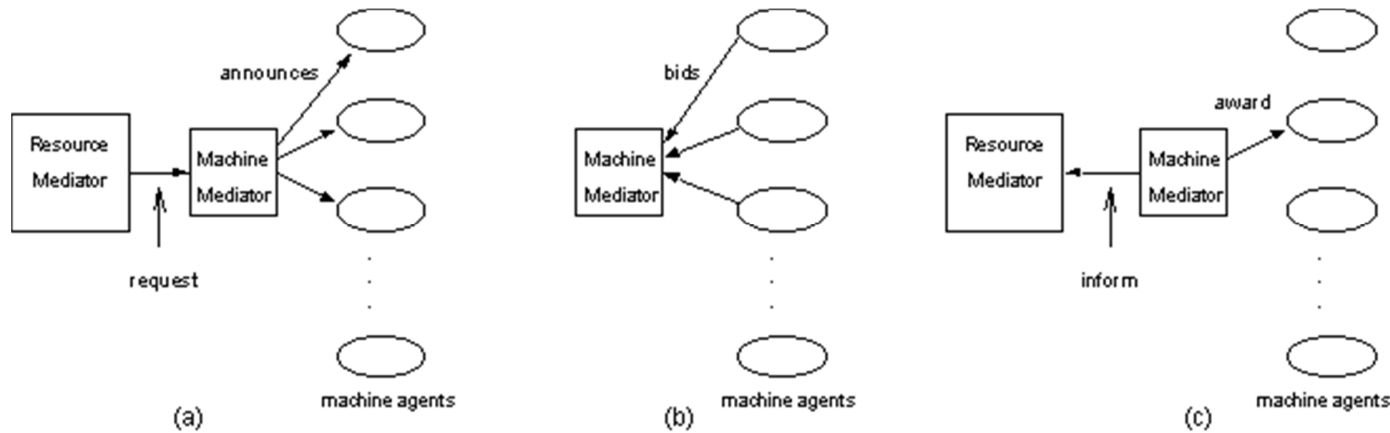


Figure 5. Mediation and bidding mechanism for Machine Mediator and machine agents

(Machine-Centered Scheduling)

At the Worker Mediator Level

The Bidding process between the Worker Mediator and the worker agents, and the Mediation process at the Worker Mediator, are basically similar to these at the Machine Mediator level. But the Worker Mediator does not send out an award message to the selected worker agent for reservation until it receives a award message from the machine agent who sent the *request* message previously and it is noticed that this machine agent has been selected as the winning agent. Otherwise, it does not make an award to any worker agent.

At the Tool Mediator Level

The Bidding process between the Tool Mediator and the tool agents, and the Mediation process at the Tool Mediator, are similar to these at the Worker Mediator level. The main difference is that at the Worker Mediator level, only one worker agent can be awarded, but at the Tool Mediator level, several tool agents may be awarded a requested task.

5.4 Priority and Concurrent (Parallel) Scheduling

In our current implementation, all scheduling requests come from a feature-based functional design system via a Design Mediator (as shown in Figure 3). Each product corresponding to a customer order has been decomposed into related manufacturing features. Each manufacturing feature can be realized by manufacturing tasks (operations). Each product is modeled as a part agent containing the information about this product including the feature data, due time, its plan which will be assigned with times and resources during the scheduling process.

The parts with earliest due time (in other words, they have highest priority) are scheduled first. All the parts with the same due time (with the same priority) will be scheduled at the same time, i.e., all the part agents with the same due time send out their first task request simultaneously. They are scheduled simultaneously. Each part agent sends out next task request as soon as its current task has been successfully scheduled.

5.5 Dynamic Rescheduling

As we intend to develop an agent-based manufacturing scheduling system for a real manufacturing enterprise, the system should respond dynamically to emergent production orders and unforeseen situations. Several mechanisms have been developed for responding to some special possible emergent situations (cases).

Case 1: Rescheduling for new emergent orders

Because the system developed is to be customer oriented as described in Section 3, rescheduling for new emergent orders is quite frequent and very important. When a new emergent order arrives after the normal scheduling is completed, the Resource Mediator will first try to schedule it normally by negotiating with resource agents. If it cannot find time slots for scheduling this emergent order, our solution is to release some scheduled orders (release the production plan of the corresponding part agents) for which the scheduled end time is well before the due time. The

scheduled part agents are released one by one until the emergent order is successfully scheduled or no possible part agents can be released. These released part agents are then rescheduled after the scheduling of the emergent order is finished.

Case 2: Rescheduling after cancellation of orders

For the orders to be cancelled, all that needs doing is only to release the corresponding part agents. In general, the other production tasks may not need to be rescheduled. Of course, it may be rescheduled for an optimization reason (see next case).

Case 3: Rescheduling for the optimization of the current schedule

If it is found that the schedule does not effectively utilize the manufacturing resources (e.g., the utilization of the machines is less than a predefined percentage) due to the changing situations such as cancellation of several orders, earliness of some production tasks, and so on, a rescheduling process may be undertaken for all production tasks which have not been started. In certain cases, it may be more economic for the factory (here the job shop) to intentionally and occasionally make several resources (machines, workers etc) non-active before rescheduling.

Case 4: Rescheduling after machine breakdown with the product destroyed

When a machine breakdowns and the product is destroyed due to machine failure, the concerned product has to be rescheduled for all its tasks. This situation can be considered as an insertion of a new emergent order and the product (order) can be rescheduled as in the case 1.

Case 5: Rescheduling after machine breakdown without destroying the product

If a machine breakdowns without destroying the product, only the uncompleted tasks of this product should be rescheduled. This can also be rescheduled as in the case 1 except that the first task to be scheduled is the current task rather than the first task of the product.

Case 6: Rescheduling after tardiness of a task

If the tardiness of a task will affect the next task of the current product, the subsequent tasks of this product have to be rescheduled. This can be done as in the case 5. An optimization rescheduling as in the case 3 may be done if it is necessary.

5.6 Clustering and Cloning

Clustering and cloning approaches for manufacturing scheduling were proposed and developed during the MetaMorph I project ([Maturana and Norrie 1996](#)). In order to reduce scheduling time through parallel computation, resources agents are cloned as needed. These clone agents are included in virtual coordination clusters where agents negotiate with each other to find the best solution for a production task. Decker et al ([1997](#)) used a similar cloning agent approach for one of an information agent's possible responses to overloaded conditions.

In the current implementation, these two concepts have been used with improved mechanisms. When the Machine Mediator receives a *request* message from the Resource Mediator which itself is requested by a part agent, it creates a clone Machine Mediator, and sends *announce* messages to a group of selected machine agents according to its knowledge. After receiving the *announce* message, a machine agent creates a clone agent and participates in the negotiation cluster. During the negotiation process, the clone machine agent needs to negotiate with tool agents and worker agents. It sends a request message to the Worker Mediator and the Tool Mediator. Similar to the Machine Mediator, the Worker Mediator and the Tool Mediator create their clone mediator agents. They send *announce* messages to worker agents and tool agents to call for bidding. The concerned worker agents and tool agents create their clones which will then participate in the negotiation cluster (see Figure 6).

According to experiments during the MetaMorph I project, clustering and cloning mechanisms can be very useful approaches for improving the manufacturing scheduling performance. In the case that the system is running in simulation mode, resource agents are active objects with goals and associated motivations. They are, in general, located in the same computer. The clone agents are, in fact, clone objects. In the case of real on-line scheduling, the cloning mechanism can be used to "clone" resource agents from remote computers (like NC machines, manufacturing cells and so on) to the local computer (where the resource mediators reside) so as to reduce communication time and consequently to reduce the scheduling/rescheduling time. This idea is related to mobile agent technology ([Rothermel and Popescu-Zeletin 1997](#)).

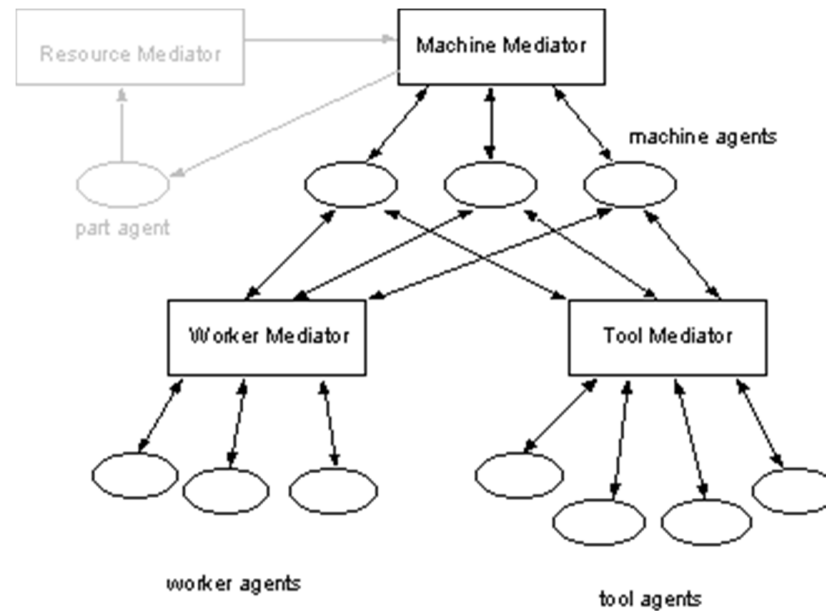


Figure 6. Virtual coordination cluster

6 Prototype Implementation

The previous version of the MetaMorph was implemented on a network of HP workstations using VisualWorks[†]. Communication among mediators was realized using TCP/IP protocol and the inter-mediator messages were formatted in KQML format ([Finin et al 1993](#)). This version has also been tested on a local network of HP workstations, SUN workstations and PCs.

To fit within the manufacturing enterprise where part of the system will prospectively be incorporated, the present implementation is being developed on a network of PCs. The present implementation of MetaMorph II consists of four mediators (Figure 7): Enterprise Mediator, Design Mediator, Resource Mediator and Marketing Mediator. Enterprise Mediator can be considered as the administration center of the manufacturing enterprise. Other mediators register with this mediator. Design Mediator is used to integrate a functional design system. Resource Mediator coordinates an

agent-based manufacturing scheduling subsystem described in Section 5.3. Marketing Mediator is used to integrate the customer services to the system. The main development language is VisualWorks[†] (Smalltalk). The graphical interface for customer services is to be developed using 3D studio[‡] and Visual C++[§]. The current implementation is being developed initially as a simulation.

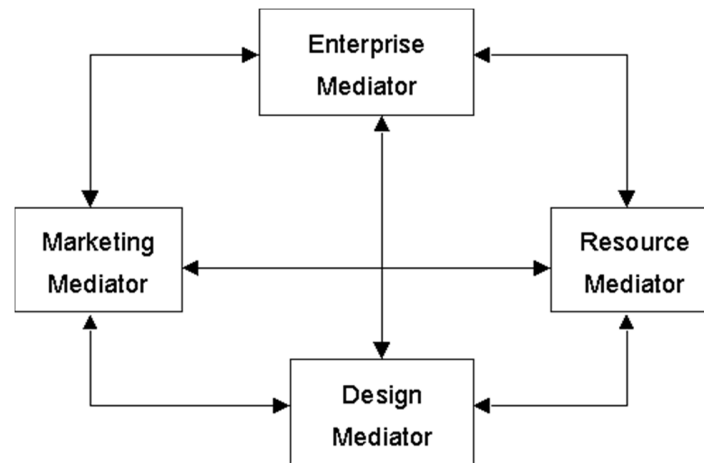


Figure 7. An implementation of MetaMorph II

7 Conclusion and Future Work

Worldwide market competition and frequently changing customer requirements are forcing manufacturing enterprises to reconsider how they are organized. One solution for manufacturing enterprise is to move towards open architectures for integrating their activities with their suppliers and customers within wide supply chain networks. Traditional knowledge engineering approaches with large scale or very large scale knowledge bases become inadequate in this highly distributed environment. Agent technology derived from Distributed Artificial Intelligence provides a promising approach for efficient enterprise integration.

Different agent-based architectures have been proposed in the research literature. According to our previous research experience, the autonomous agent architecture is well suited for developing distributed intelligent design systems when

existing engineering tools are encapsulated as agents and connected to the system for providing special services, and the system consists of a small number of agents. In the federation architecture with facilitators or mediators, a hierarchy is imposed for every specific task which provides computational simplicity and manageability. This type of architecture is quite suitable for developing distributed manufacturing systems which are complex, dynamic, and composed of a large number of resource agents. This paper proposed a hybrid architecture combining the main features of the above mentioned different architectures. Compared with other architectures, this hybrid architecture has following advantages:

- a. It is easy to integrate existing software and hardware so as to resolve the legacy problem;
- b. Knowledge capitalization is realized primarily at the mediator level. When the system configuration changes, only the mediators should be informed and updated, which obviously reduces communications among subsystems and agents;
- c. It is easy to maintain the system because of its modular architecture;
- d. It is easy to connect remote services such as marketing services, material supply services through communication systems such as the Internet;
- e. It also provides the system with flexibility and scalability.

However, the hybrid architecture and the heterogeneous environment may increase the complexity of manufacturing systems and the difficulty for system development and integration. Some drawbacks of centralized systems, such as bottleneck problem, may emerge due to the mediator-centric organization.

MetaMorph II is an ongoing project. A major short-term research goal is to realize a prototype integrating customer services, functional design and manufacturing scheduling, in simulation mode, and then to implement this approach in a real manufacturing factory.

References

Balasubramanian, S., Maturana, F. and Norrie, D. (1996). Multi-agent planning and coordination for distributed concurrent engineering. *International Journal of Cooperative Information Systems*, 5(2-3): 153-179.

Brazier, F.M.T. and Treur J.J. (1996). Modelling Project Coordination in a Multi-Agent Framework. In Proceedings of WET ICE'96, Stanford, CA.

Brown, D., Dunskus, B., Grecu, D. and Berker, I. (1995). SINE: Support for Single Function Agents. In Proceedings of AIENG'95, Applications of AI in Engineering, Udine, Italy.

Butler, J. and Ohtsubo, H. (1992). ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. *Artificial Intelligence Applications in Manufacturing*, Famili, A., Nau, D.S., and Kim, S.H., (eds.), The AAAI Press, pp. 199-214.

Cutkosky, M.R., Engelmores, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C. (1993). PACT: An Experiment in Integrating Concurrent Engineering Systems. *IEEE Computer*, **26**(1): 28-37.

Cutkosky, M.R., Tenenbaum, J.M. and Glicksman J. (1996). Madefast: Collaborative Engineering over the Internet. *Communication of the ACM*, **39**(9):78-87.

Davis, J. and Oliff, M. (1988). Requirements for the Integration of Manufacturing Planning Islands Using Knowledge Based Technology. *Expert Systems and Intelligent Manufacturing*, Oliff, M. (editor), North-Holland, pp. 25-42.

Decker, K. (1995). Environment centered analysis and design of coordination mechanisms. Ph.D. Thesis, Dept. of Computer Science, University of Massachusetts, Amherst.

Decker, K., Sycara, K. and Williamson, M. (1997). Cloning for Intelligent Adaptive Information Agents. *Multi-Agent Systems: Methodologies and Applications*, Lecture Notes in Artificial Intelligence 1286, Zhang, C. and Lukose, D., (eds.), Springer, pp. 63-75.

Finin, T., Fritzon, R., McKay, D. and McEntire, R. (1993). KQML - A Language and Protocol for Knowledge and Information Exchange. Tech. Report, University of Maryland, Baltimore.

Fischer, K., Muller J.P, Pischel, M. and Schier, D. (1995). A Model for Cooperative Transportation Scheduling. In Proceedings of the First International Conference on Multi-Agent Systems, AAAI Press/MIT Press, San Francisco, California, pp. 109-116.

Forbus, K.D. (1988). Intelligent computer-aided engineering. *AI magazine*, **9**(3):23-36.

Fox, M.S., Barbuceanu, M., and Gruninger, M., (1996). An Organization Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour. *Computers in Industry*, **29**:123-134.

Gaines, B.R., Norrie, D.H., and Lapsley, A.Z. (1995). Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise. In Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, pp. 964-969.

Genesereth, M. and Fikes, R. (1992). Knowledge Interchange Format, Version 3.0 Reference Manual. Computer Science Department, Stanford University, Tech Report Logic-92-1.

Goldmann, S. (1996). Procura: A Project Management Model of Concurrent Planning and Design. In Proceedings of WET ICE'96, Stanford, CA.

Gruber, T. (1993). A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, **5**(2), pp. 199-220.

Lenat, D.B. and Guha, R.V. (1989). *Building Very Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Reading, MA, USA.

Maturana, F. and Norrie, D. (1996). Multi-Agent Mediator Architecture for Distributed manufacturing. *Journal of Intelligent Manufacturing*, **7**:257-270.

Maturana, F., Balasubramanian, S. and Norrie, D.H. (1996). A Multi-Agent Approach to Integrated Planning and Scheduling for Concurrent Engineering. In Proceedings of the International Conference on Concurrent Engineering: Research and Applications, Toronto, Ontario, pp. 272-279.

McGuire, J., Huokka, D., Weber, J., Tenenbaum, J., Gruber, T., and Olsen, G. (1993). SHADE: Technology for Knowledge-Based Collaborative Engineering. *Journal of Concurrent Engineering: Applications and Research*, **1**(3).

Pan, J.Y.C. and Tenenbaum, M.J. (1991). An intelligent agent framework for enterprise integration. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(6): 1391-1408

Parunak, H.V.D., Baker, A.D. and Clark, S.J. (1997a). The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA.

Parunak, H.V.D., Ward, A., Fleischer, M. and Sauter, J. (1997b). A Marketplace of Design Agents for Distributed Concurrent Set-Based Design. In Proceedings of the Fourth ISPE International Conference on Concurrent Engineering: Research and Applications, Troy, Michigan.

Park, H., Tenenbaum, J. and Dove, R. (1993). Agile Infrastructure for Manufacturing Systems (AIMS): A Vision for Transforming the US Manufacturing Base. Defense Manufacturing Conference.

Park, H., Cutkosky, M., Conru, A. and Lee, S.H. (1994). An Agent-Based Approach to Concurrent Cable Harness Design, *AIEDAM*, **8**(1).

Patil, R., Fikes, R., Patel-Schneider, P., Mckay, D., Finin, T., Gruber, T. and Neches, R. (1992). The DARPA Knowledge Sharing Effort: Progress Report. Rich, C., Nebel, B., and Swartout, W. (eds.), *Principles of Knowledge Representation and Reasoning*: Proceedings of the Third International Conference, Cambridge, MA, Morgan Kaufmann.

Petrie, C., Cutkosky, M., Webster, T., Conru, A. and Park, H. (1994). Next-Link: An Experiment in Coordination of Distributed Agents. Position paper for the AID-94 Workshop on Conflict Resolution, Lausanne.

Roboam, M. and Fox, M.S. (1992). Enterprise Management Network Architecture. *Artificial Intelligence Applications in Manufacturing*, Famili, A., Nau, D.S. and Kim, S.H., (eds.), The AAAI Press, pp. 401-432.

Rothermel, K. and Popescu-Zeletin, R. (eds.) (1997). *Mobile Agents*, Lecture Notes in Computer Science 1219, Springer.

Saad, A., Biswas, G., Kawamura, K., Johnson, M.E. and Salama, A. (1995). Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems. In Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI'95), Workshop on Intelligent Manufacturing, Montreal, Canada, pp. 310-321.

Scalabrin, E. (1996). *Conception et Réalisation d'environnement de développement de systèmes d'agents cognitifs*. Thèse de Doctorat, CNRS UMR Heudiasyc, Université de Technologie de Compiègne.

Shen, W. and Barthès, J.P. (1995). DIDE: A Multi-Agent Environment for Engineering Design. In Proceedings of the First International Conference on Multi-Agent Systems, San Francisco, USA, The AAAI press/The MIT press, pp. 344-351.

Shen W. and Barthès J.P. (1997). An Experimental Environment for Exchanging Engineering Design Knowledge by Cognitive Agents. *Knowledge Intensive CAD-2*, Mantyla M., Finger S. and Tomiyama, T., (eds.), Chapman and Hall, pp. 19-38.

Smith, R.G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29(12):1104-1113.

Tomiyama, T., Umeda, Y., Ishii, M., Yoshioka, M. and Kiriyama, T. (1995). Knowledge systematization for a knowledge intensive engineering framework. *Knowledge Intensive CAD-1*, Tomiyama, T., Mantyla, M. and Finger, S., (eds.), Chapman & Hall, pp. 55-80.

Toye, G., Cutkosky, M., Leifer, L., Tenenbaum, J. and Glicksman, J. (1993). SHARE: A Methodology and Environment for Collaborative Product Development. In Proceeding of 2nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Press.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer*, **25**(3):38-49.

Wooldridge, M. and Jennings, N.R. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115-152.