MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing

Francisco Maturana*, Weiming Shen and Douglas H. Norrie**
Division of Manufacturing Engineering, The University of Calgary
2500 University Dr. NW, Calgary, AB, Canada T2N 1N4
Tel: +1 403 220 5787 Fax: +1 403 282 8406
E-mail: [maturana | wshen | norrie]@enme.ucalgary.ca

Abstract

Global competition and rapidly changing customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. Traditional centralised manufacturing systems are not able to meet such requirements. This paper proposes an agent-based approach for dynamically creating and managing agent communities in such widely distributed and ever-changing manufacturing environments. After reviewing the research literature, an adaptive multi-agent manufacturing system architecture called MetaMorph is presented and its main features are described. Such architecture facilitates multi-agent coordination by minimising communication and processing overheads. Adaptation is facilitated through organizational structural change and two learning mechanisms: learning from past experiences and learning future agent interactions by simulating future dynamic, emergent behaviours. The MetaMorph architecture also addresses other specific requirements for next generation manufacturing systems, including scalability, reliability, stability, maintainability, flexibility, real-time planning and scheduling, standardised communication, fault tolerance, and security. The proposed architecture is implemented as a multi-agent virtual manufacturing system, in simulation form, which incorporates heterogeneous manufacturing agents within different agent-based shop floors or factories. The experimental results have shown the potential of the agent-based approach for advanced manufacturing systems.

1 Introduction

Global competition and rapidly changing customer requirements are forcing major changes in the production styles and configuration of manufacturing organizations. Increasingly, traditional centralised and sequential manufacturing planning, scheduling, and control mechanisms are being found to be insufficiently flexible to respond to changing production styles and highly dynamic variations in product requirements. With the traditional centralised approaches to manufacturing planning, scheduling and control, the entire factory is generally controlled by central software, which limits the expandability and reconfiguration capabilities of the manufacturing systems. Using hierarchical organization forces the grouping of manufacturing resources into permanent, tightly coupled subgroups, where information is processed sequentially by a centralised software supervisor. This may result in much of the system being shut down by a single point of failure, as well as plan fragility and increased response overheads.

Agent technology derived from Distributed Artificial Intelligence provides a natural way to overcome such problems, and to design and implement distributed intelligent manufacturing environments. This paper introduces an adaptive agent-based mediator-centric architecture for intelligent manufacturing systems. The rest of this paper is organised as follows: Section 2 reviews some related projects; Section 3 depicts an adaptive agent-based manufacturing system architecture called MetaMorph; Section 4 discusses the mediators and coordination in MetaMorph; Section 5 describes the learning mechanisms developed for MetaMorph; Section 6 presents a prototype implementation; Section 7 gives some conclusions and perspectives.

2 Research Literature

Recently, agent technology has been considered as an important approach for developing industrial distributed systems (Jennings *et al* 1995). A number of researchers have attempted to apply agent technology to manufacturing enterprise integration, supply chain management, manufacturing scheduling and control, material handling, and holonic manufacturing systems.

Pan and Tenenbaum (1991) proposed a software Intelligent Agent framework for integrating people and

* Currently at: Rockwell Automation, Cleveland, OH, USA. E-mail: Fpmaturana@ra.rockwell.com
** Corresponding author

computer systems in large, geographically dispersed manufacturing enterprises. Roboam and Fox (1992) proposed an Enterprise Management Network to support the integration of activities of the manufacturing enterprise throughout the production life cycle with six levels: Network Layer, Data Layer, Information Layer, Organization Layer, Coordination Layer and Market Layer. Barbuceanu and Fox (1997) further proposed to organise the supply chain as a network of cooperating agents, each performing one or more supply chain functions, and each coordinating their actions with other agents. MADEFAST (Cutkosky et al 1996) was a DARPA DSO-sponsored project to demonstrate technologies developed under the DARPA MADE (Manufacturing Automation and Design Engineering) program. It was an ambitious experiment in collaborative engineering over the Internet.

In AARIA (Parunak *et al* 1997), the manufacturing capabilities (e.g. people, machines, and parts) are encapsulated as autonomous agents. Each agent seamlessly inter-operates 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. 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. Fischer (1994) proposed a hierarchical planning structure consisting of six layers: the layer of the production planning and control system, the layer of the shop floor control system, the task coordination layer, the task planning layer, the task execution layer and the machine control layer.

Under the international Intelligent Manufacturing Systems (IMS) Research Program, there have been several studies where intelligent agents play an important role. For example, the intelligent agents in holonic manufacturing systems (HMS) are to operate autonomously and cooperatively in an open, distributed, and intelligent manufacturing system (Christensen 1994, Van Leeuwen and Norrie 1997). In this type of system, agents are used to model holons which are software and hardware entities. A discussion on agent technology for holonic manufacturing systems can be found in (Bussmann 1998). This type of architecture allows integration of appropriate elements of hierarchical and heterarchical systems into an intelligent and open structure. A new type of control architecture, suited for such distributed intelligent manufacturing systems, using an agent-based intelligent controller, has been described in (Brennan *et al* 1997).

3 An Adaptive Multi-Agent Architecture for Intelligent Manufacturing

An adaptive agent-based architecture called MetaMorph (Maturana 1997) is proposed to address system adaptation and extended-enterprise issues at four fundamental levels: virtual enterprise, distributed intelligent systems, concurrent engineering, and agent architectures. The *virtual enterprise* deals with the strategic partnership issues associated with the unification of heterogeneous manufacturing subsystems into a large, dynamic, virtual coalition of cooperative subsystems. *Distributed intelligent systems* are then required to enable increased autonomy of each member of the manufacturing enterprise network. Each manufacturing partner (or manufacturing subsystem) will pursue individual goals while satisfying both local and external constraints. Applying *concurrent engineering* to product/process design and production management becomes fundamental for managing manufacturing information and reducing time to market. It is not possible to meet extended enterprise requirements under the present configuration of manufacturing systems. Therefore, to cope with increasing complexity and organizational issues, *intelligent agent architectures* are envisaged to build and operate extended manufacturing enterprises.

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.

3.1 Mediator-Centric Federation Organization

MetaMorph uses an agent-based mediator-centric federation architecture (Wiederhold 1992, Gaines *et al* 1995). In this particular type of federation organization, intelligent agents can link with mediator agents (also called mediators) to find other agents in the environment. Additionally, mediators assume the role of system coordinators by promoting cooperation among intelligent agents and learning from the agents' behaviour. Mediators provide system associations without interfering with low-level decisions unless critical situations occur. Mediators are able to expand their coordination capabilities to include mediation behaviours, which may be focused upon high-level policies to break decision deadlocks. Mediation actions are performance-directed behaviours.

Mediators can use brokering and recruiting communication mechanisms (Decker 1995) to find related agents for establishing collaborative subsystems (also called coordination clusters or virtual clusters). 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 then can step out of the scene to let the agents proceed with the communication themselves. Both mechanisms have been used in MetaMorph. To efficiently use these mechanisms, mediators 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.

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

3.2 Agent Classification

Within the spectrum of agent architectures, there are two basic families of agents: deliberative (intentional or cognitive) and reactive. A deliberative agent is one that contains an explicitly represented, symbolic model of the world, and in which decisions are made via logical reasoning, based on pattern matching and symbolic manipulation. A deliberative agent is able to reason about its environment and beliefs, to create plans of actions, and to execute those plans. A reactive agent is one that does not include any kind of central symbolic world model, and does not use complex symbolic reasoning (Wooldridge and Jennings 1995). A reactive agent reacts to changes in its environment or to messages from other agents. A hybrid agent is both deliberative and reactive.

There are two main types of agents in MetaMorph: resource agents and mediator agents (also called mediators). Resource agents are used to represent manufacturing devices and operations, while mediator agents are used to coordinate the interactions among agents (resource agents and also mediator agents). In the MetaMorph architecture, hybrid agent models are used to build both resource and mediator agents. These models can be classified as soft-hybrid agent models because none of the reactive-agent levels are strongly implemented in the agent structures.

Different levels of intelligence and behaviour are associated with the two different types of agents. Resource agents are autonomous and cooperative. Mediator agents are autonomous, cooperative, and learning. Figure 1 shows the basic characteristics of resource and mediator agents.

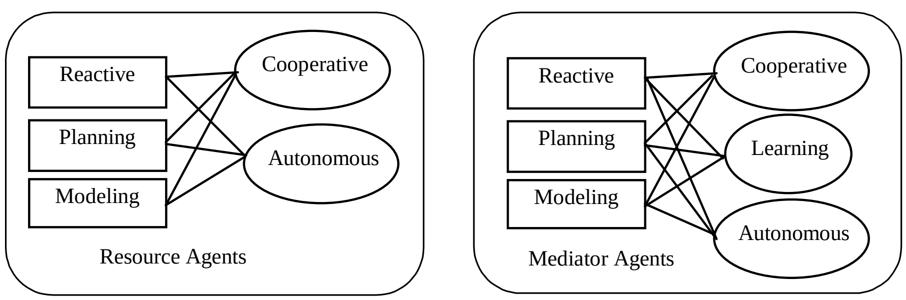

Figure 1. Basic Characteristics of Resource and Mediator Agents

3.3 Agent Coalition (Clustering)

In order to work cooperatively, agents may form coalitions (clusters) that bond dissimilar agents into harmonious decision groups. Multistage negotiation and coordination protocols that can efficiently maintain the stability of these coalitions are required. Each agent has its individual representation of the external world, goals, and constraints, so diverse heterogeneous beliefs interact within a coalition through distributed cooperation models.

In MetaMorph, the core negotiation mechanism is based on task decomposition and dynamically-formed agent groups (clusters). High-level tasks are initially decomposed by mediators acting at the corresponding information level. Each subtask is subsequently distributed to determine the best solution plan. Mediators learn dynamically from the agent interactions and identify coalitions that can be used to establish distributed searches for the resolution of tasks.

Agent coalition is incorporated in the main problem-solving mechanism in MetaMorph. Agents are dynamically contracted to participate in a problem-solving group (cluster). In case of the situations where the agents in the problem-solving group (cluster) are only able to partially complete the task's requests, the agents will seek outside their cluster and establish conversation links with the agents in other clusters. This process is repeated, with sub-clusters being formed and then sub-sub-clusters etc as needed within a dynamically interlinked 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 described, provides scalability and aggregation properties to the system.

3.4 Agent Cloning

In MetaMorph, resources agents are cloned as needed for concurrent information processing. These clone agents are included in virtual coordination clusters where agents negotiate with each other to find the best solution for a production task. The clustering and cloning mechanisms with mediator coordination will further be discussed in the following Section.

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 (in CNC 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.

4 Mediators and Coordination in MetaMorph

Because of the mediator-centric organization, the mediator design is one of key issues in the MetaMorph project. Mediators are intended to encapsulate various manufacturing behaviours 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, has been proposed. Such generic model can enable different types of mediators to be created to cope with various activities in the factory. These meta-level activities are high-level abstractions of behaviours 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 meta-level activities are logically grouped within specific domains as part of a generic methodology for building mediators. Depending on the specific software design, a domain group may be implemented as a Class. These activity groups in the domains are, in essence, functional or activity objects that can be distributed across the factory system and linked through an intelligent network. The degree of distribution of the sub-activities (subgroups) in a domain depends on the complexity of the manufacturing behaviour to be modelled and coordinated. The domain groups are designed to each include generic specific expertise to coordinate the domain's activity.

4.1 Generic Model for Mediators

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, as shown in figure 2. Each mediator includes some or all of these activities to a varying extent.

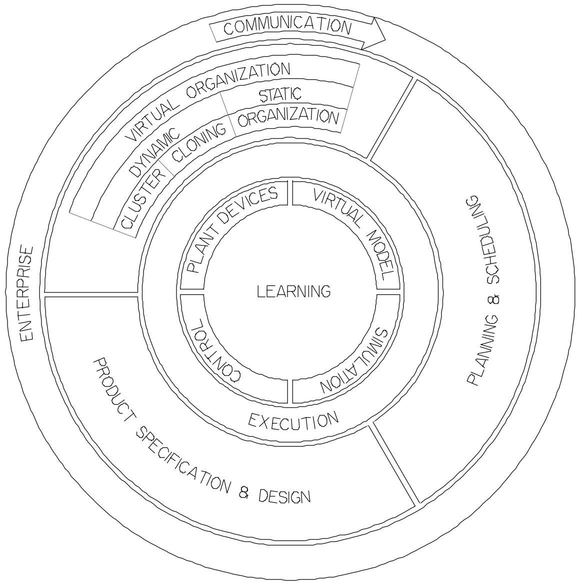

Figure 2. Generic model for mediators

Prototyping with this generic model and methodology facilitates the creation of diverse types of mediators. For example, a mediator may be specialised 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. The activity domains in figure 2 are further described as follows:

- The enterprise domain has globalised knowledge of the system and represents the facility's goals through a series of objectives. Enterprise knowledge enables environment recognition and maintenance of organizational associations.
- The product specification and design domain includes encoding data for the manufacturing task to enable mediators to recognise the tasks to be coordinated.
- The virtual organization domain is similar to the enterprise domain, but its scope is detailed knowledge of resource behaviour at the shop-floor level. This domain dynamically establishes and recognises dynamic relationships between dissimilar resources and agents.
- The planning and scheduling domain plays an important role in integrating technological constraints with time-dependent constraints into a concurrent information-processing model (Balasubramanian *et al* 1996).
- The execution domain facilitates transactions among physical devices. During the execution of tasks, it coordinates various transactions between manufacturing devices and between the devices and other domains to complete the information requirements.
- The communication domain provides a common communication language based on the KQML protocol (Finin *et al* 1993) used to wrap the message content.
- The learning domain incorporates the resource capacity planning activity, which involves repetitive reasoning and message exchange and which can be learned and automated.

Manufacturing requests associated with each domain are established under both static and dynamic conditions. The static conditions relate to the design of the products (geometrical profiles). The dynamic conditions depend upon times, system loads, system metrics, costs, customer desires, etc. A more detailed description of the generic model for mediator design can be found in (Maturana 1997).

4.2 Different Types of Mediators in MetaMorph

There are two different coordination tasks in MetaMorph, which requires two different levels of mediators. The first specialises in coordination of virtual groups (inter-coordination) and can be called a high-level mediator. The second is specialised for coordination among intelligent agents within a virtual group (intra-coordination) and can be called a low-level mediator. However, these mediators are not restricted to these operations alone.

Every mediator encapsulates functionality to allow local coordination and interaction with other dissimilar mediators. Mediators responsible for mapping system entities are classified as static mediators. These are characterised as high-level mediators directly connected to physical objects. In addition, there are dynamic mediators for coordinating dynamic interaction among agents. Dynamic mediators are dynamically created through agent-replication action.

Each manufacturing enterprise needs at least one high-level mediator (called enterprise mediator) to act as the system's integrator. This enterprise mediator is able to recognise all sub-level mediators, platforms, and resources in the enterprise. The enterprise mediator supplies a global view of a system during integration of plans.

4.3 Coordination in MetaMorph

Coordination initially involves two main phases: subtasking and creation of virtual coordination clusters. These activities are supported by the static mediators, the Data-Agent Managers (DAM), and the Active Mediators (AM), each of which coordinates its specific level in the overall coordination task.

At the resource community level, shop floors are provided with high-level static mediators, which constitute the enterprise model of the system. Static mediators use their classification mechanisms to learn from the system's activity and to update the organizational knowledge of the system. A high-level task initially passes through the static mediator for recognition and decomposition. According to the shop-floor capability, the subtasks are each assigned to separate coordination clusters. Each coordination clusters will also incorporate clone agents obtained from active manufacturing agents. These entity interrelationships form a concurrent coordination framework.

4.4 Active Mediators

Because coordination action is a very complex branching task, the instantiation of decentralised mediators has been chosen for as a basic mechanism for distributed coordination. The Active Mediators (AMs) carry out control and mediation actions upon clone agents. In the first stage of coordination, the AM broadcasts the unsolved task to the clone agents under its supervision. Each clone agent analyses the task and prepares bids for assisting the planning activity or rejecting the tasks. The clone agents, one by one, reply with their bids back to the AM. At this stage, coordination of the messages is essential to maintain the stability of the decision group and complete the planning process. The AM captures the messages and analyses them to either incorporate their respective agents within the decision group or to dismiss incompatible clone agents, as shown in figure 3. The final decision-making group is rearranged and committed bids are exchanged among the remaining clone agents within the coordination cluster. The communication in this coordination task is primarily selective broadcasting. The AM needs to recognise the senders of the bids and the receptor of the bids.

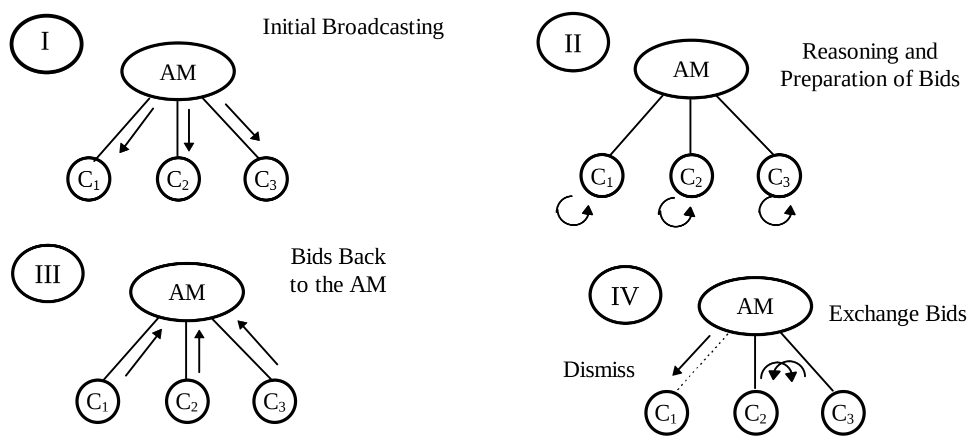

Figure 3. Active Mediator Coordination

If a coordination cluster has many clone agents, the following guidelines enhance intra-agent coordination efficiency: (1) The AM regulates the number of messages to be exchanged according to variable priorities. When there are many agents in a coordination cluster, the AM is very strict in the selection of messages. In cases of high message traffic, exchanges are limited to two messages at a time. (2) In regulating message exchange, cost tables are implemented to prioritise agent communication. (3) When the number of agents is decreased (the clone agents progressively dismiss themselves from the negotiation group), the AM relaxes the selection of messages.

4.5 Data Agent Managers

To find additional resources in response to a request, in a way that does not overload the AM, inter-cluster and other 'external' coordination activities are provided by Data Agent Managers (DAMs). The DAM receives the request for additional resources and searches for appropriate agents to fulfil it. The DAM communicates with the static mediator for this resource community, and the overall system is searched for additional resources. The static mediator locates the resources and notifies the DAM with the respective address (of an individual agent or coordination cluster). Subsequently, the clone agent is informed about the new resources and direct communication between it and the new resource may be established if additional transactions are needed. AM and DAM each have separate computing processes with individual threads of execution.

4.6 Clone Agents

The resource agent connected to a manufacturing resource maintains real-time information about the resource's physical activity. This information includes the operational status of the resource, committed plans, schedules, and temporal interactions with other resources. This resource agent also uses the cloning mechanism meta-level activities to replicate itself in different coordination clusters as needed, as shown in figure 4. Through this mechanism, clone agents can be involved in distributed and concurrent planning.

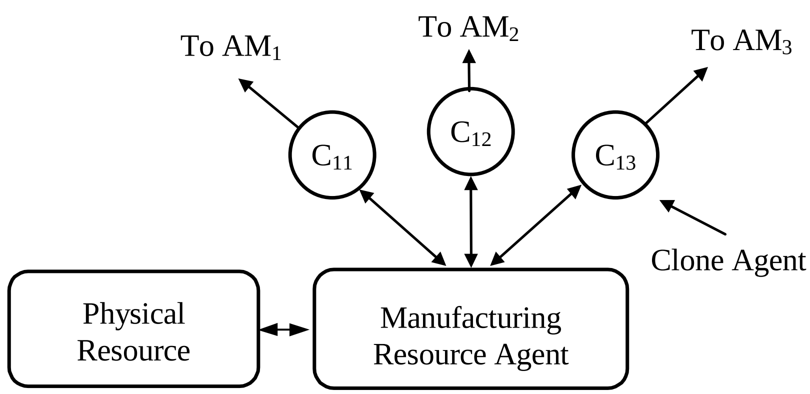

Figure 4. Clone Agent Coordination

Each clone agent initially maintains the current state of its resource when it is created. In this manner, variations in the state of the resource are dynamically introduced into the planning activity. Clone agents are subsequently affected by dynamic changes in the task priorities.

The clone agents can develop only promissory plans for tasks but cannot *commit* the resource to these (only the manufacturing resource agent can commit to a task since it manages and executes the resource's schedule). The complexity of the task affects the solution overheads and the number of clone agents that arrive at a promissory plan stage at any specific time. The manufacturing resource agent assesses these promissory plans at discrete periods.

5 Learning in MetaMorph

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 behaviours. Second, a mechanism for propagating the system's behaviour into the future is implemented to help mediators 'to learn from the future'.

The context of manufacturing requests is established under static and dynamic variations. Static variations relate to the physical configuration of products. Dynamic variations depend on time, system loads, system metrics, costs, customer desires, etc. These two main sources of information relate to a wide spectrum of emergent behaviours, which can be separated into specific behavioural patterns. A 'learning from history' mechanism based on distributed case-based learning approach was developed during MetaMorph for capturing such behavioural 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. A manufacturability or manufacturing request sent to a resource community is first filtered by the respective resource mediator to decide whether the request can be recognised as associated with a previously considered product or is unknown. If it is recognised, the resource mediator retrieves the learned patterns to send to the select group of agents identified in the patterns. For an unknown request, the resource community's mediator uses its standard matchmaking actions to specify the primary set of resource agents to be contacted regarding their capability to satisfy this request. The solution to this unknown request then proceeds through the propagation of coordination clusters and decision strategies previously described. During this process, the resource mediator involved learns from partial emergent interactions at the coordination cluster level. This learning is distributed among several coordination clusters. The plan aggregation process then enables the classification of various feature-machine-tool patterns, which are encoded and provided to the community's mediator for storage and future reuse.

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 behaviour of the virtual model which emulates the shop-floor activities. By partially projecting 'unpredictable behaviours' and agent interactions, the agent-based manufacturing system is able to correct its real-world model and provide more accurate plans. Experiments during the MetaMorph project have shown that this forecasting simulation is quite efficient for adjusting and enhancing the system's performance.

A detailed description of these learning and reasoning mechanisms in MetaMorph have been presented separately (Maturana et al 1997).

6 Prototype Implementation

The MetaMorph architecture and coordination protocols, described previously, have been used for implementing a distributed concurrent design and manufacturing system in simulated form. This virtual system incorporates heterogeneous manufacturing agents in different agent-based shop floors or factories (physically separated) that are dynamically interconnected to carry out concurrent manufacturability evaluation, production planning and scheduling. This system is composed of the following multi-agent modules: Enterprise Mediator, Design System, Shop Floors, and Execution Control & Forecasting, as shown in figure 5. Each multi-agent module uses common enterprise integration protocols to allow agent interoperability.

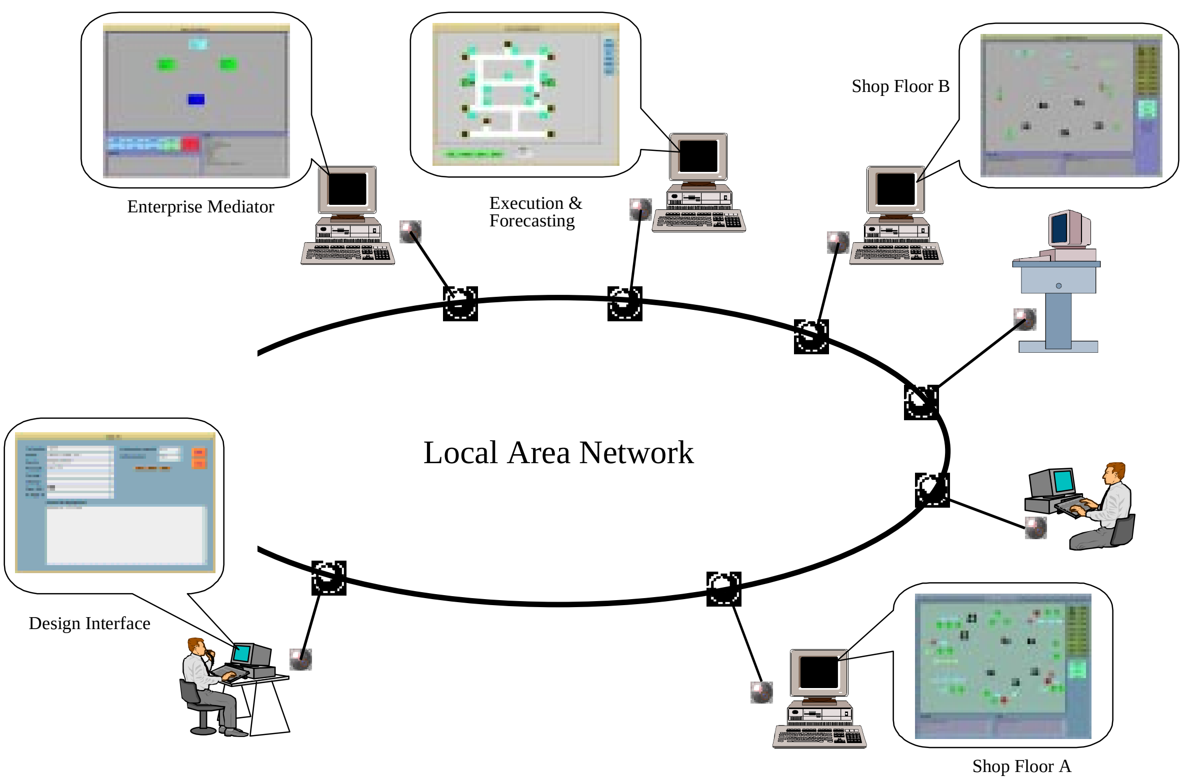

Figure 5. Prototype Implementation of MetaMorph Architecture

The multi-agent modules are implemented within a distributed computing platform consisting of four HP Apollo 715/50 work stations, each running an HP-UX 9.0 operating system. The workstations communicate with each other through a Local Area Network (LAN) and TCP/IP protocol. Graphical interfaces for each multi-agent module are created in VisualWorks 2.5 (Smalltalk) programming language, which is also used for programming the modules. The KQML protocol (Finin *et al* 1993) is used as high-level agent communication language. The whole system is coordinated by high-level mediators, which help provide the integration mechanisms for realising an extended enterprise (Maturana and Norrie 1996).

The Enterprise Mediator acts as the coordinator for the enterprise, and all of the manufacturing shop floors and other modules are registered with it. Registration processes are carried out through macro-level registration communications. Each multi-agent manufacturing module offers its services to the enterprise through the Enterprise Mediator.

A graphical interface has been created for the Enterprise Mediator. Both human users and agents are allowed to interact with the Enterprise Mediator and registered manufacturing modules via KQML messages. Decision rules and enterprise policies can be dynamically modified by object-call protocols through input field windows by the user. Action buttons support quick access to any of the registered manufacturing modules, shown as icon-agents, as well as to the Enterprise Mediator’s source code.

The Enterprise Mediator offers three main services: integration, communication, and mediation. Integration permits the registration and interconnection of manufacturing components, thereby creating agent-to-agent links. Communication is allowed in any direction among agents and between human users and agents. Mediation facilitates coordination of the registered mediators and shop floor resources.

The design system module is mainly a graphical interface for retrieving design information and requesting manufacturability evaluations through the Enterprise Mediator (which also operates as shop-floor manager and message router). Designs are created in a separate intelligent design system named the Agent-Based Concurrent Design Environment (ABCDE), developed in the same research group (Balasubramanian *et al* 1996).

Different shop floors can be modelled and incorporated in the system as autonomous multi-agent components each containing machine and tool agent communities. Shop-floor resources are registered in each shop floor using macro-level registration policies. Machine and tool agents are incorporated into the resource communities through micro-level registration policies. The shop-floor modules encapsulate the planning activity of the shop floor. Each shop floor interface is provided with a set of icon-agents to represent shop-floor devices. Shop-floor interfaces provide standardised communication and coordination for processing manufacturability evaluation requests. These modules communicate with the execution control and simulation module to refine promissory schedules. Coordination services are supported by the community’s mediator and DAM and AM mediators as described in Section 4.

The execution control and forecasting module is the container for execution agents and process-interlocking protocols. Shop-floor resources are introduced in the system, thereby instantiating icon-agents and specifying data files for each resource. This module includes icon-agents for its graphical interface to represent machines, warehouses, collision avoidance areas, and AGV agents. Standard operation times (i.e. loading, processing, unloading, and transportation times) are already provided but can be scaled to each resource’s desired characteristics. Each resource can be created to enforce a specific dispatching rule (i.e. weighted shortest processing time, earliest due date, shortest processing time, FIFO, LIFO, etc). Parts are modelled as part agents that are implemented as background processes.

A local execution mediator is embedded in the module to integrate and coordinate shop-floor resources. This local execution mediator communicates with the resource mediator to get promissory plans and to broadcast forecasting results.

The system can be run in different time modes: real-time and forecasting. In the real-time mode, the speed of the shop-floor simulation is proportional to the execution speed of the real-time system. In the forecasting mode, the simulation speed is 40-60 times faster than the real-time execution.

7 Conclusions and Perspectives

An adaptive agent-based architecture called MetaMorph has been proposed for intelligent manufacturing systems using the mediator-centric federation approach. Such an architecture facilitates multi-agent coordination by minimising communication and processing overheads. Adaptation is facilitated through organizational structural change (to suit both external and internal (evolving) task requirements) and through two learning mechanisms: learning from past experiences and learning future agent interactions by simulating future dynamic, emergent behaviours. The architecture is generic and can be applied to distributed organizations in other domains in addition to that of manufacturing.

The MetaMorph architecture also addresses other specific requirements for next generation manufacturing systems, including scalability, reliability, maintainability, flexibility, real-time planning and scheduling, standardised communication, fault tolerance, stability, learning, forecasting, and security. These requirements have been considered in developing the proof-of-concept simulated manufacturing system used to evaluate the MetaMorph concepts for a production environment:

- *Scalability*: Physical devices are represented through icon-agents in graphical interfaces. The icon-agents are representatives of intelligent objects that are registered in the resource community mediators. Each graphical interface corresponds to an agent-based subsystem and is registered in the static mediator domain. The system can add or remove agents and agent subsystems through message communication.
- *Reliability*: Each agent performs its activity autonomously and through cooperative interaction to accomplish plans. Coordination protocols are provided to assist during agent negotiation.
- *Maintainability*: Intelligent agents are loosely connected to the graphical interfaces. This facilitates the modification of agent capabilities and rules by simply accessing the agent objects and methods. Agents may also be modified and updated through message communication.
- *Flexibility*: Single agent capabilities are unified in collaboration groups to adapt to diverse reasoning circumstances. The coordination protocols to assist such collaboration groups are scalably implementable.
- *Real-time planning and scheduling*: Planning and scheduling of operations are carried out in real-time through interrelated concurrent processes (Maturana *et al* 1996).
- *Standardised communication*: Agents share a common ontology to express their beliefs and intentions. The KQML has been customised for the simulated manufacturing system developed.
- *Fault tolerance*: Malfunctioning of a resource agent (shop-floor agent) is kept at a local level. A resource breakdown may be simulated by introducing a breakdown period into the resource. Each job allocated within the halt-period is rescheduled to other available time slots found in the same resource (the malfunctioning resource) or in a different resource
- *Stability*: Agents are coordinated by distributed mediator agents in coordination clusters to prevent inappropriate responses.
- *Learning*: Mediator agents are capable of learning emergent relationships from the past and from the future during agent interactions. These two learning activities allow rapid responses during task planning and adjustment of plans and schedules.
- *Forecasting*: Production forecasting is incorporated in the system through agent-based simulation for both management and production planning. Forecasting models can interactively support planning and scheduling decisions.
- *Security*: Security can be controlled in the corporate environment by establishing management policies for computer and interface access; i.e. users are provided with appropriate access privileges. Manufacturing organizations must use secure intra-networks for bi-directional exchange of information. The system's integrity must be ensured, while at the same time maintaining transparent accessibility with external resources.

The experimental results from the MetaMorph simulation have shown the potential of the agent-based approach for advanced manufacturing systems. Work has now commenced on MetaMorph II, whose enhanced capabilities will embody lessons learned from our previous research work. Our short-term perspectives are to implement the MetaMorph II prototype in simulated form, by improving the original MetaMorph's coordination protocols, learning mechanisms, inter-agent and inter-mediator communication mechanisms. The long-term perspectives are to apply our experimental prototypes to develop industrial applications.

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.

BARBUCEANU, M. and FOX, M., 1997, Integrating communicative action, conversations and decision theory to coordinate agents. Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA.

BRENNAN, R.W., BALASUBRAMANIAN, S., and NORRIE, D., 1997, Dynamic control architecture for advanced manufacturing systems. Proceedings of International Conference on Intelligent Systems for Advanced Manufacturing, Pittsburgh, PA.

BUSSMANN, S., 1998, An agent-oriented architecture for holonic manufacturing control. Proceedings of First International Workshop on IMS, Lausanne, Switzerland, pp. 1-12.

CHRISTENSEN, J., 1994, Holonic manufacturing systems: initial architecture and standards directions. Proceedings of First European Conference on Holonic Manufacturing Systems, Hanover, Germany.

CUTKOSKY, M., ENGELMORE, R., FIKES, R., GENESERETH, M., GRUBER, T., MARK, W., TENENBAUM, J. and WEBER, J., 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.

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

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.

FISCHER, K., 1994, The design of an intelligent manufacturing system. Proceedings of the 2nd International Working Conference on Cooperating Knowledge-based Systems, University of Keele, UK, pp. 83-99.

GAINES, B., NORRIE, D., and LAPSLEY, A., 1995, Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise. Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, pp. 964-969.

JENNINGS, N.R., CORERA, J.M. and LARESGOITI, I., 1995, Developing industrial multi-agent systems. Proceedings of the First International Conference on Multi-Agent Systems, San Francisco, CA, The AAAI press/The MIT press, pp. 423-430.

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. Proceedings of the International Conference on Concurrent Engineering: Research and Applications, Toronto, Ontario, pp. 272-279.

MATURANA F., BALASUBRAMANIAN S. and NORRIE D.H., 1997, Learning coordination patterns from emergent behaviour in a multi-agent manufacturing system, Proceedings of Intelligent Systems and Semiotics '97: A Learning Perspective, Gaithersburg, Maryland.

MATURANA, F., 1997, MetaMorph: an adaptive multi-agent architecture for advanced manufacturing systems, Ph D thesis, The University of Calgary.

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., 1997, The AARIA agent architecture: an example of requirements-driven agent-based system design. Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA.

ROBOAM, M. and FOX, M., 1992, Enterprise management network architecture. In A. Famili, D.S., Nau, S.H. and Kim, *Artificial Intelligence Applications in Manufacturing*, (The AAAI Press), pp. 401-432.

SAAD, A., BISWAS, G., KAWAMURA, K., JOHNSON, M. E. and SALAMA, A., 1995, Evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems. Proceedings of the 1995 International Joint Conference on Artificial Intelligence, Workshop on Intelligent Manufacturing, Montreal, Canada, pp. 310-321.

VAN LEEUWEN, E.H. and NORRIE, D.H., 1997, Intelligent manufacturing: holons and holarchies. *Manufacturing Engineer*, **76**(2), 86-88.

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. *Knowledge Engineering Review*, **10**(2), 115-152.