17 Enterprise Management Network Architecture

A Tool for Manufacturing Enterprise Integration

Michel Roboam
Aerospatiale Company

Mark S. Fox
University of Toronto

Abstract

Achieving manufacturing efficiency requires that the many groups that comprise a manufacturing enterprise, such as design, planning, production, distribution, field service, accounting, sales and marketing, cooperate in order to achieve their common goal. In this chapter we introduce the concept of Enterprise Management Network as the element to facilitate the integration of distributed heterogeneous functions of a manufacturing enterprise. The integration is supported by having the network, first, play a more active role in the accessing and communication of information, and second by providing the appropriate protocols for the distribution, coordination, and negotiation of tasks and outcomes. The Enterprise Management Network is divided into six layers: Network Layer, Data Layer, Information Layer, Organization Layer, Coordination Layer, and Market Layer. Each of these Layers provides part of the elements, functions and protocols to allow the integration of a manufacturing enterprise.

17.1 Introduction

This chapter presents the architecture, the elements and the organization of an Enterprise Management Network (EMN) to support the integration of the Manufacturing Enterprise. The performance of manufacturing can be enhanced by greater integration of activities throughout the production life cycle. Integration must not only address the issues of shared information and communication, but how to coordinate decisions and activities throughout the firm.

In the history of Manufacturing systems, four steps have been identified [13]:

1. *Manufacture then to sell*: At this step the priority was given to resources. Enterprises manufactured products ready to be delivered to the market. The enterprise's goals were to maximize resource utilization using the concept of Economic Order Quantity. The result was product inventories that differed from market demands. The problem was exacerbated with the diversification of finished products; as the number of products increased so did inventories.

Reprinted from Artificial Intelligence Applications in Manufacturing
AAAI Press/The MIT Press, 1992

2. *Manufacture what will be sold*: To reduce inventories, companies attempted to predict market demand using forecasting techniques. Production plans were then based on these predictions. Material Requirement Planning I embodies this approach and operates well when the market and the products are stable. With increased product customization, MRP techniques have been found to be less useful.
3. *Integrated manufacturing*: The manufacturing enterprise must be viewed more broadly. We must include in this activity all stages from raw material acquisition to distribution and servicing of the products. Computer systems support the management of such a structure. They are based on MRP I but they add other functions such as master scheduling, load planning, stock management, dispatching, etc. They constitute the MRP II systems.
4. *Manufacture what is sold*: In order to manufacture what is sold, the enterprise has to be much more responsive to the dynamics of the market place. The CIE (Computer Integrated Enterprise) concept increases the responsiveness of enterprises in four areas: delay, cost, quality and quantity. Optimizing manufacturing efficiency requires that the many groups that comprise a manufacturing enterprise, such as design, planning, production, distribution, field service, accounting, sales and marketing, cooperate to achieve their common goal. Cooperation can take many forms:
   - Communication of information relevant to one or more groups' tasks. For example, sales informing marketing of customer requirements, or production informing the controller of production performances.
   - Feedback on the performance of a group's task. For example, field service informing design and manufacturing of the operating performance of a new product.
   - Monitoring and controlling activities. For example, controlling the execution of operations on the factory floor.
   - Negotiation of change. For example, manufacturing may want to negotiate changes in product tolerances with design engineering in order to reduce production costs.
   - Assignment of new tasks. For example, a new product manager signing up production facilities to produce a new product.

Achieving the responsiveness of the fourth stage requires an information and control infrastructure beyond what is currently available today. We view the Enterprise Management Network as an attempt to remedy this situation. We view the EMN as the "nervous system" of the enterprise, enabling the functions described above. It is more than a network protocol (e.g., MAP) in that it operates and participates at the application level. The following describes the capabilities to be provided by the EMN:

- *Information routing*: Given a representation for information to be placed on the network and a representation of the goals and information needs of groups on the network, the information routing capability is able to provide the following:
  - Static routing: Transferring information to groups where the sender and the receivers are pre-defined.
  - Dynamic routing: Transferring information to groups which appear to be interested in the information. This is accomplished by matching a group's goals and information needs to the information packet.
  - Retrospective routing: Reviewing old information packets to see if they match new goals and information requirements specified by a group.
- *Closed loop system*: Often, the communication of information results in some activity, which the initiator of the communication may be interested in. The EMN will support the providing of feedback in two modes:
  - Pre-defined feedback: Operationalizes pre-defined information flows between groups in the organization. For example, production providing feedback to sales on the receipt of orders.
  - Novel feedback: Providing feedback for new and novel messages.
- *Command and control*: Given a model of the firm which includes personnel, departments, resources, goals, constraints, authority and responsibility relations, the EMN will support these lines of authority and responsibility in the assignment, execution and monitoring of goals and activities. In particular, it will manage the distribution of information and the performance of tasks.
- *Dynamic task distribution*: Supporting the creation of new organizational groups and decomposition, assignment and integration of new goals and tasks, contracting and negotiation are examples of techniques to be supported.

The design of the Enterprise Management Network is divided into six levels. The levels have been chosen based on prior experience. We believe the completion of the first version of the EMN will lead to reassignment of functions among levels and the possible creation of new levels.

1. The *Network Layer* provides for the definition of the network structure. At this level, nodes (EMN node) are named and declared to be part of the network. Message sending (or message passing) between nodes is supported along with synchronization primitives (such as "blocking").
2. The *Data Layer* provides for inter-node queries using a language patterned after SQL [5, 6].
3. The *Information Layer* provides "invisible" access to information spread throughout the EMN. The goal is to make information located anywhere in the network locally accessible without having the programs executed locally know where in the network the information is located nor explicitly request its retrieval. Distribution is based on data classes and content [17, 1, 25].
4. The *Organization Layer* provides the primitives and elements (such as goal, role, responsibility and authority) for distributed problem solving [22, 15, 16, 20, 11]. It allows automatic communication of information based upon the roles of a node in the organization.
5. The *Coordination Layer* provides protocols for coordinating the activities of EMN nodes through negotiation and cooperation mechanisms.
6. The *Market Layer* provides protocols for coordinating organizations in a market environment. It supports the distribution of tasks, the negotiation of change, and the strategies to deal with the environment.

In the rest of this chapter, the information routing capabilities of the EMN are described, starting with a description of the architecture of a network node (EMN node), and followed by a description of the six layers of the EMN architecture (The specification and implementation of this architecture and of the communication system are described in [30, 31, 32].

17.2 Enterprise Management Network Node

The Enterprise Management Network links together two or more application nodes (EMN node) by providing the architectures and mechanisms to support decision making at all levels of the organization. For example, the CORTES system [12] is composed of an uncertainty analyser, a planner, a scheduler, a factory model and dispatchers responsible for several machines Figure 17.1. Each is called an EMN node.

Each EMN node consists of the following subsystems(Currently implemented in CommonLISP):

Each is called an EMN node.

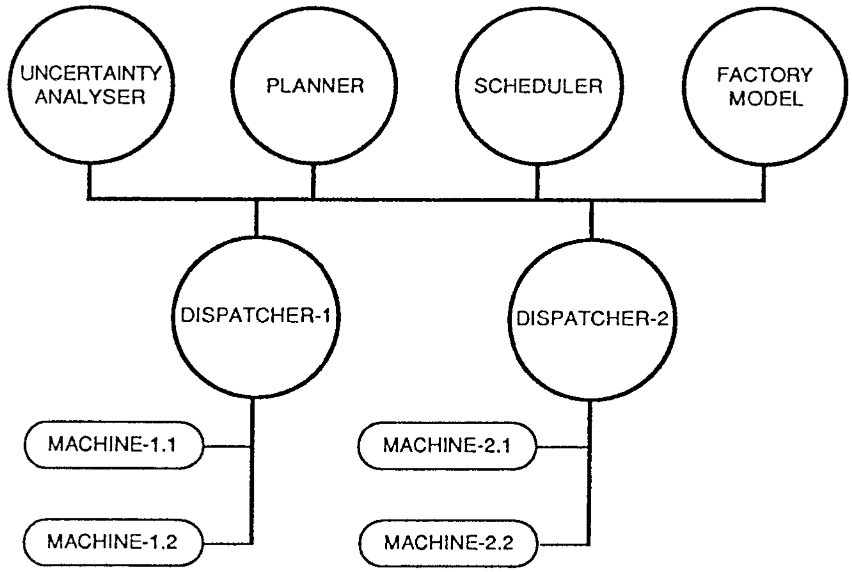

Figure 17.1
Example of a Decentralized System.

- Problem Solver,
- Knowledge Base,
- Knowledge Base Manager, and
- Communication Manager.

The *problem solving subsystem* represents all the rules and functions which allow the EMN node to solve problems related to its domain. The local execution cycle is triggered either by internal transactions generated during local problem-solving, or by external events forwarded to the EMN node by the Communication Manager.

Each EMN node contains a locally maintained *knowledge base* to support its problem solving. It is composed of entities (or objects) which may be either physical objects (products, resources, operations, etc), or conceptual objects (customer orders, process plans, communication paths, temporal relations, etc). The knowledge base is expressed as schemata (also known as frames) in the Knowledge Craft/(R) representation language CRL [19].

The *Knowledge Base Manager* manages information exchanges between the problem solving subsystem and the knowledge base, maintains the consistency of the local knowledge base, and responds to request made by other EMN nodes. In the Enterprise Man-

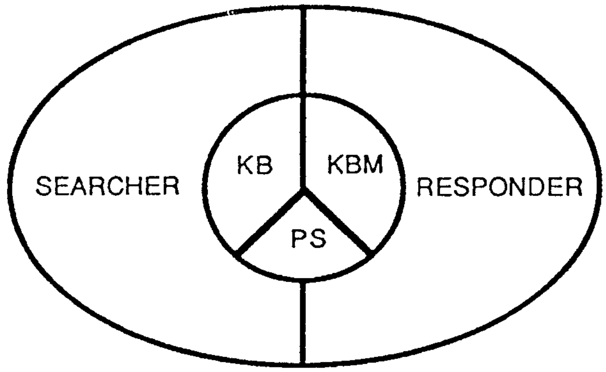

Figure 17.2
The Elements of an EMN Node.

agement Network, knowledge and data may be distributed throughout the network. It is the philosophy of the system that knowledge does not have to be available locally in order for it to be used by the EMN node. Nor does knowledge have to be globally consisten, only locally, within an EMN node. Therefore, knowledge, in the form of schemata, fall into one of two classes: that *owned* by the knowledge source and must be stored locally, and knowledge *used* by the knowledge source, in which the original is stored at another EMN node and a copy is stored locally.

A problem that arises in supporting the exchanges between the problem solving subsystem and the knowledge base, is the unavailability of schemata locally. The problem solver often refers to knowledge that cannot be found locally, but may be found in another EMN node's knowledge base. At the time of reference, the problem solver may or may not know where in the Enterprise Management Network the knowledge resides. It is the responsibility of the Knowledge Base Manager to "hunt down" the missing knowledge and to respond to like requests from other EMN nodes. To accomplish this, the EMN node has as part of it a *Communication Manager*. It manages both the search for information in the EMN and responds to requests from other EMN nodes. To perform these activities, the Communication Manager has two modules:

- The *searcher* corresponds via message sending with other EMN nodes. The searcher peforms two tasks: searching for knowledge not available locally, and the updating of knowledge changed and owned by the EMN node. The policy for updating is defined in Section 5.
- The *responder* answers messages originating from other EMN nodes' searchers, and updates the local knowledge according to updating messages.

Figure 17.2 summarizes the structure of an EMN node.

The communication manager manages four types of interaction:

- *Triggering*: Information that triggers the node's processing.
- *Dynamic retrieval*: Requests for information not available in its knowledge base but necessary to perform its task. These information needs appear during the internal activity (processing) of a EMN node.
- *Updating information*: When an EMN node that owns some schemata modifies them, the searcher dispatches the modifications to other EMN nodes that have local copies. The responder may or may not update a local copy depending on the usage at the receiving EMN node. Ownership of a schema means that the EMN node is the only one allowed to modify globally the content of a schema. But each EMN node that has a local copy of a schema can modify locally the content of that schema.
- *Transaction request*: Similar to remote procedure calls.

We summarize all these exchanges between the modules of an EMN node in Figure 17.3. This figure shows the types of information sent and received by each module (M stands for Message, A stands for Answer, R stands for Request, T stands for Translator and CT stands for Correspondence Table). We will discuss in the following sections the content of these transactions.

The first three layers of the Enterprise Management Network architecture are defined in the remaining sections. Each layer provides further detail on the functionality and operation of EMN nodes. To illustrate the specific content of these layers, we will consider a decentralized system composed of 3 agents, connected by a network. Each agent has a specific Problem Solving subsystem (PS) and a specific Knowledge Base subsystem (KB) as shown in figure 17.4. We describe in this first figure a decentralized system without the Enterprise Management Network. We will use this example by adding at each level specific elements, functions and protocols.

17.3 Network Layer

17.3.1 EMN Node Identification

The architecture presented in this document covers only the upper levels of a network protocol. We assume the existence of lower levels provided by a network such as DECNET (DECNET is a registered trademark of Digital Equipment Corporation.).

The Network Layer defines the network structure:

- *EMN nodes* represent problem solving agents. They include the basic communication objects: mail box, semaphore box, queues and low level messages. To support

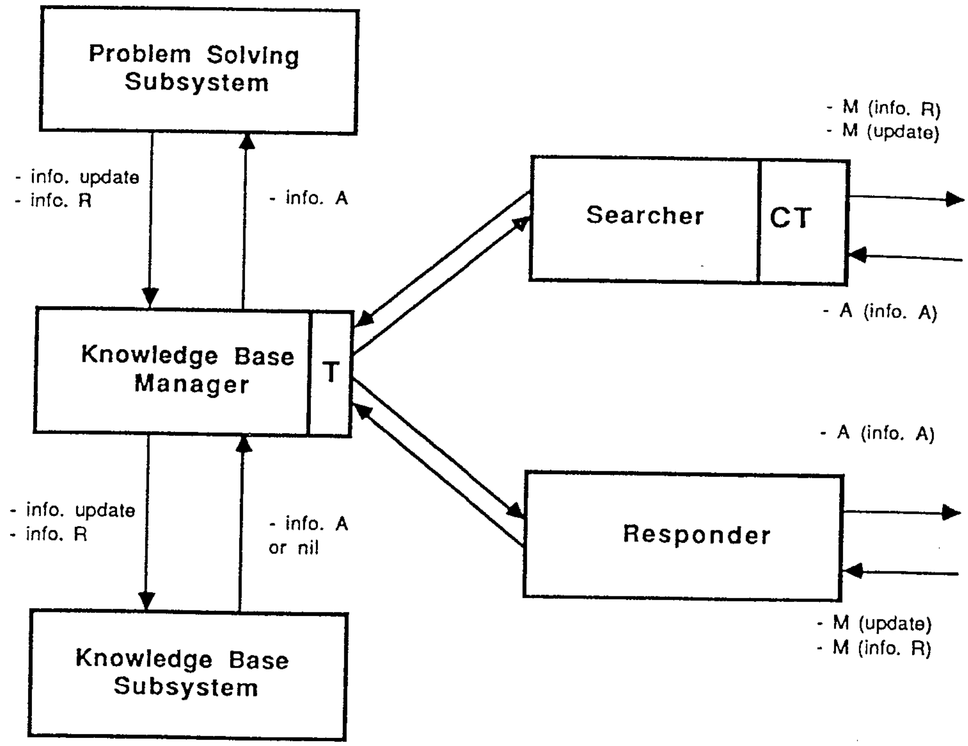

Figure 17.3
Information Exchanges Overview.

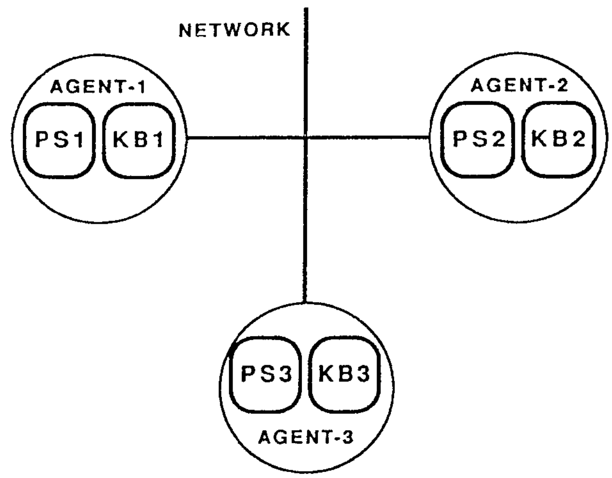

Figure 17.4
Decentralized System Example.

EMN node initialization, we use schemata such as the EMN node schema (figure 17.5).

- *Channels* define communication links between EMN nodes. Each channel is defined as an instance of the channel schema.
- *Messages* can be sent along channels between EMN nodes. During information transfer, an EMN node may be suspended (blocked) while awaiting a reply. Each message is defined as an instance of message schema. These instances are stored in queues.
- *Protection* is provided so that messages can only be processed by legal EMN nodes. We have defined some mechanisms using the channel definition that allow for the recognition of legal and illegal message destinations.
- *Primitives* are defined to support information exchange and information reception (figure 17.6). In addition, blocking and unblocking functions allowing the interruption of the internal problem solving of an agent have been defined. They can be triggered with one or both of the primitives.

The message passing primitive includes several steps (Figure 17.6). The first one identifies the message destination. Channels provide EMN nodes with knowledge of other existing EMN nodes in the network. A message can be sent from one EMN node to another only if the corresponding channel already exists. The second step is the

EMNnode		
SLOT	FACET	VALUE
Name	*Value:* *Restriction:*	type string
Network	*Value:* *Restriction:*	type network-name*
DomaEMN-model	*Value:* *Restriction:*	type string
Problem-solver	*Value:* *Restriction:*	type string
Mail-box	*Value:* *Restriction:*	type symbol
Semaphore-box	*Value:* *Restriction:*	type symbol
Channels	*Value:* *Restriction:*	type channel-name*

Figure 17.5
EMN Node Schema.

verification of the channel (and also mail-box) status. Two EMN nodes cannot at the same time write a new message in one mail box. For that reason we use a channel locking mechanism. In order for an EMN node to write a new message in a mail box, it must wait until this mail box is unlocked. The last step is a sequence of three phases:

1. Lock the channel (by creating a semaphore box).
2. Write the message into the destination mail box.
3. Unlock the channel (by deleting the semaphore box).

At the Network Layer, a second primitive is defined: message reception (figure 17.6). This function allows for reading the content of the mail box. When it is triggered, it first checks the mail box status. If it is locked, it waits until it is unlocked.

The second step of this function is performed in three phases: lock mail box, read mail box, and unlock mail box.

The last primitive provided at the Network Layer is the blocking mechanism. This function allows for the interruption of the Problem Solving process. When the problem solver requires additional information, it remains suspended until it is received.

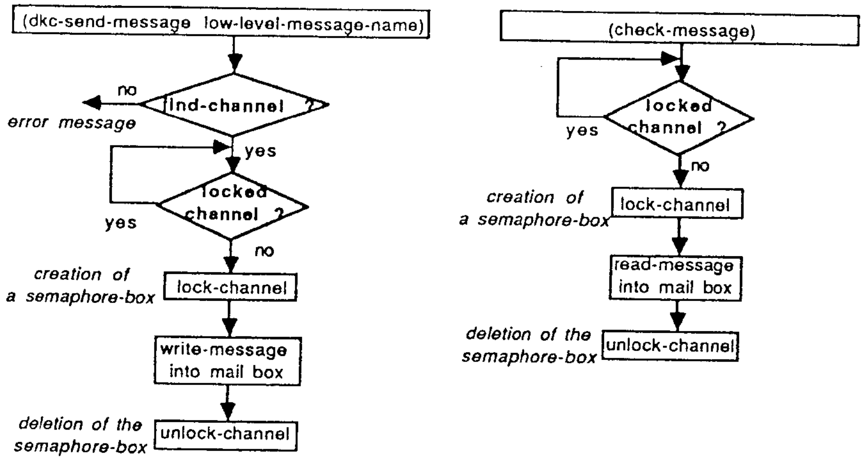

Figure 17.6
Message Passing and Message Reception Algorithms.

17.3.2 Network Layer Example

The example depicted by Figure 17.4 is extended by adding the network layer on this empty structure. The main modifications, in bold, which occur on the structure defined in Figure 17.7 are:

- The initialization of the EMN node, which includes the definition, for each agent of the decentralized system, of a name and of the mail box and semaphores.
- The creation of links between the EMN node channels.

17.4 Data Layer

17.4.1 Object and Language Concepts

The Data Layer provides EMN nodes with the capability to explicitly request and send information from/to other EMN nodes. Information is represented as schemata (aka frames) in a node's knowledge base. The protocol for requesting and asserting information between EMN nodes is based on a subset of SQL [5, 21]. In this version, schemata correspond to tables, and slots correspond to fields. Protection is provided at the schema

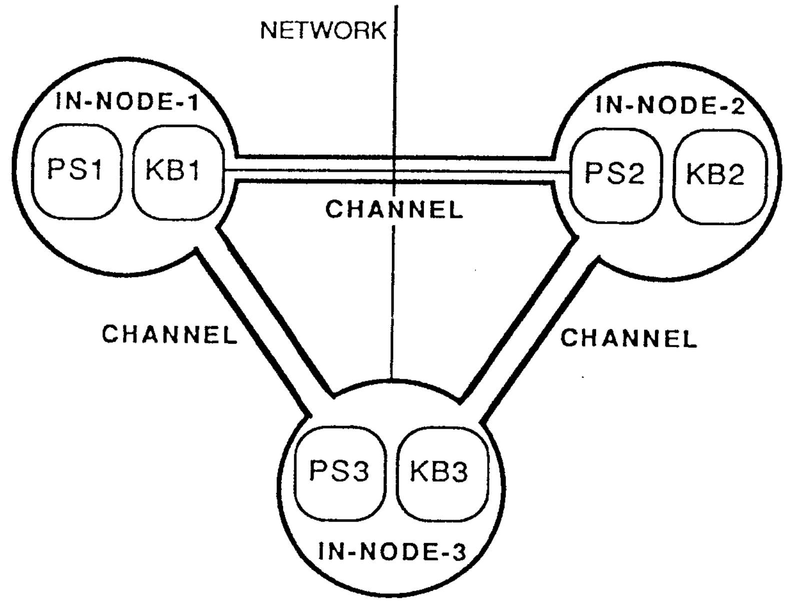

Figure 17.7
Network Layer Implementation Example.

level; access to schemata may be locked and the requesting EMN node blocked until the schema is unlocked. The following is a simple example of slot retrieval query.

The information flow between EMN nodes depends on the needs of each of them. These exchanges are done to satisfy a request for information not available in the knowledge base subsystem of an EMN node. The request is done at first on a specific type of information using for example the CRL command GET-VALUE. For example, in CRL: (GET-VALUE 'Machine 'Capacity) "machine" is the schema" name and "capacity" is the slot name. If this value is available in the knowledge base subsystem, it is returned; if not, an error message is returned). These exchanges can also be performed to ensure consistency. We will see at the next layer that several kind of communication processes can be defined. At this Layer we must provide all the elements to support these communication types.

Two sets of elements must be defined:

- The Objects manipulated (schemata),
- The query language (Figure 17.8).

Objects provide for information definition and information exchange. Both must take into account the several types of communication capabilities and must be compatible

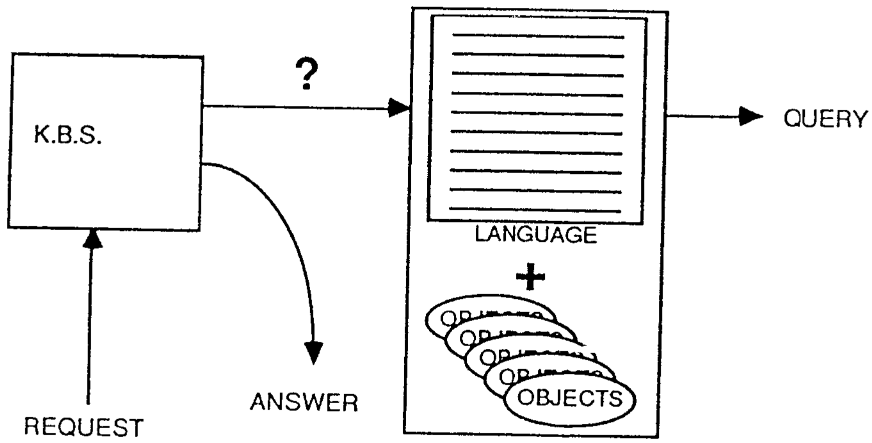

Figure 17.8
Query Elements.

(because the query language manipulates the objects and the result of a query is an object).

We have identified four types of objects:

- *Information*: A reference to the knowledge base. Each information is represented as a schema or part of a schema (slot).
- *Message*: Defined as a combination of an information need, a producer and a destination. Each message is an instance of the generic message schema.
- *Answer*: Generated to respond to a specific message schema. The answer schema includes all the slots of the message schema plus a status slot (which indicates whether the information request is provided or not) and the schema-name-answer slot (which value is 'nil or the needed information).
- *Communication Schema*: the schema (Figure 17.9) that permits the Enterprise Management Network to efficiently acquire and distribute information. By using a dictionary (and a communication language), each EMN node has the capability of a mutual understanding. The correspondence table allows for efficient information search by defining the owner of the used schemata. The User-table defines the users of the schemata owned by a specific EMN node.

To define the different query possibilities, we use a subset of an SQL type of query language (SQL is a language defined to access a relational data structure System R). The general level of the language is comparable to that of relational algebra. SQL (Structured Query Language) is more than a query language. It provides not only retrieval functions but also a full range of update operations, and also many other facilities [5,21]) which are adapted to schemata (table–schema, field–slot). The following is an example query:

```
Example:

Query: get the capacity      Result:
of the machine-1

SELECT capacity              {{MACHINE-1
FROM machine-1;              CAPACITY: 100-p/h  }}
```

Communication		
SLOT	FACET	VALUE
Correspondance-table	*Value:*	type (information EMN node-name)*
	Restriction:	
User-table	*Value:*	type (information (EMN node-name [, EMN node-name, ...]))*
	Restriction:	
Locked-schemata	*Value:*	type information*
	Restriction:	
Shared-schemata	*Value:*	type information*
	Restriction:	
Updated-schemata	*Value:*	type information*
	Restriction:	
Dictionary	*Value:*	type (information string)*
	Restriction:	
Local-address	*Value:*	type local-EMN node-name
	Restriction:	

Figure 17.9
Communication

17.4.2 Data Layer Example

As we can see, the objects for a more sophisticated communication capability have been added in each EMN node. These capabilities will be used at the upper levels to support negotiation but also for the information search and updating.

The elements we add at this layer concern the internal structure of an EMN node. If we implement these new elements in our example, figure 17.10 becomes:

17.5 Information Layer

17.5.1 Communication Functions and Protocols

The information layer of the manufacturing architecture provides the functions, rules and schemata to support information exchange among EMN nodes. The Information Layer provides two services for the EMN node:

- Automatic access to information distributed throughout the network, and
- Automatic information management.

Automatic acquisition of information provides an EMN node with the capability to acquire at any moment and without knowledge of its location in the EMN the schemata needed by their Problem solving subsystem but not available in their knowledge base subsystem. We can identify at this level two functions:

- The message sending sequence (information search sequence) (we define its algorithm on Figure 17.11), and
- The message receiving (answer sequence) (figure F7.12).

Information acquisition occurs automatically when an EMN node's problem solver attempts to access information that does not exist in it's knowledge base. Four methods are used by the knowledge base manager to acquire the information. First, the owner of the schema in which the information is referenced may be another EMN node. Therefore it is reasonable to believe that EMN node may also have the desired information. Secondly, the schema taxonomy, defined below, contains pointers to EMN nodes that maintain schemata of a particular type. Third, the EMN node maintains a list of other EMN nodes that it corresponds to regularly and may query them. Fourth, as a last resort the request may be broadcast to all EMN nodes for which it has a channel to.

We have defined at the network layer some primitives for security purpose: the locking and the blocking mechanism. Each of the two information acquisition sequences can use either.

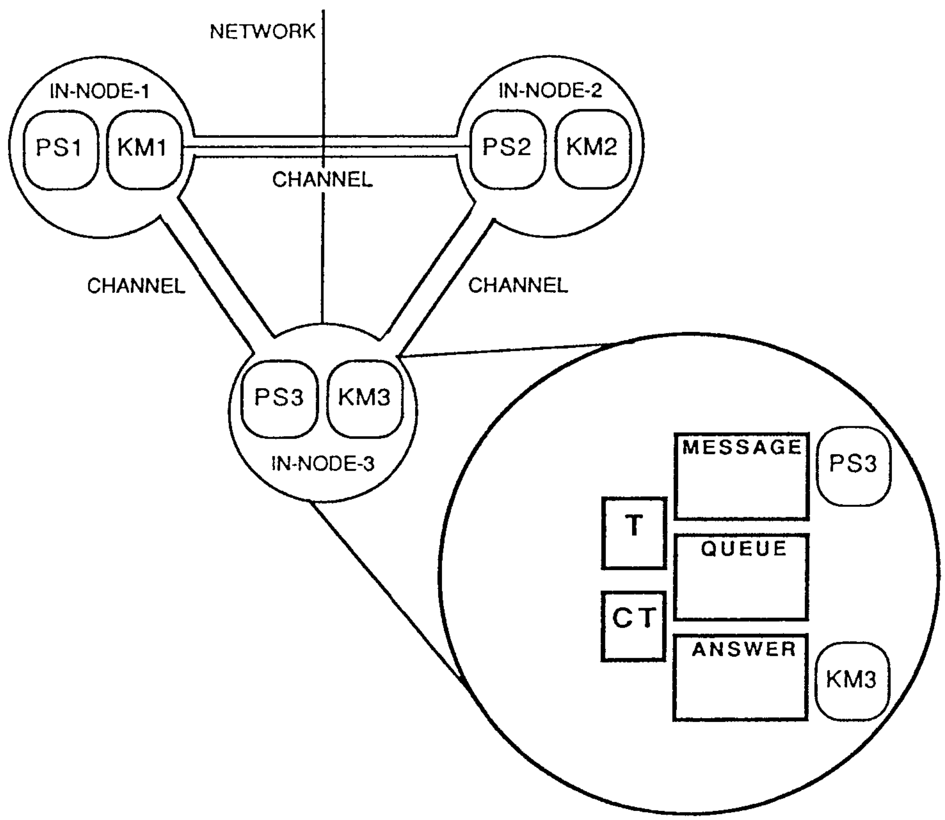

Figure 17.10
Data Layer implementation example

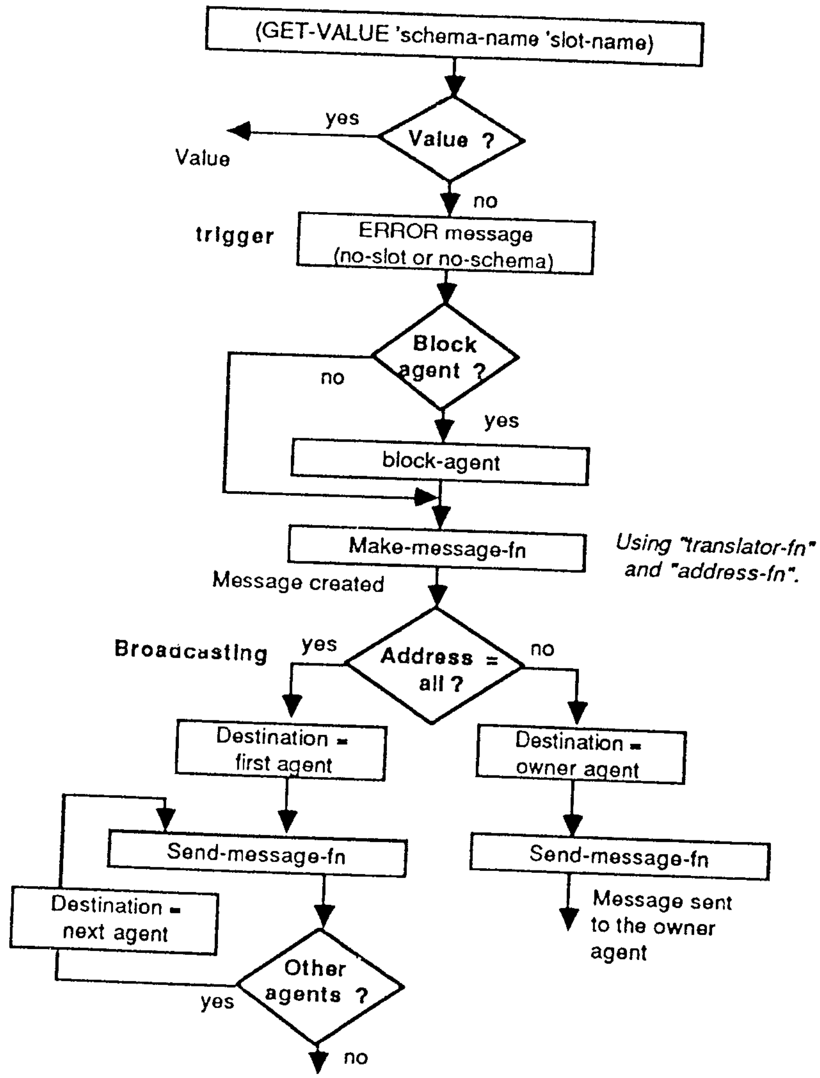

Figure 17.11
Message Sending Sequence.

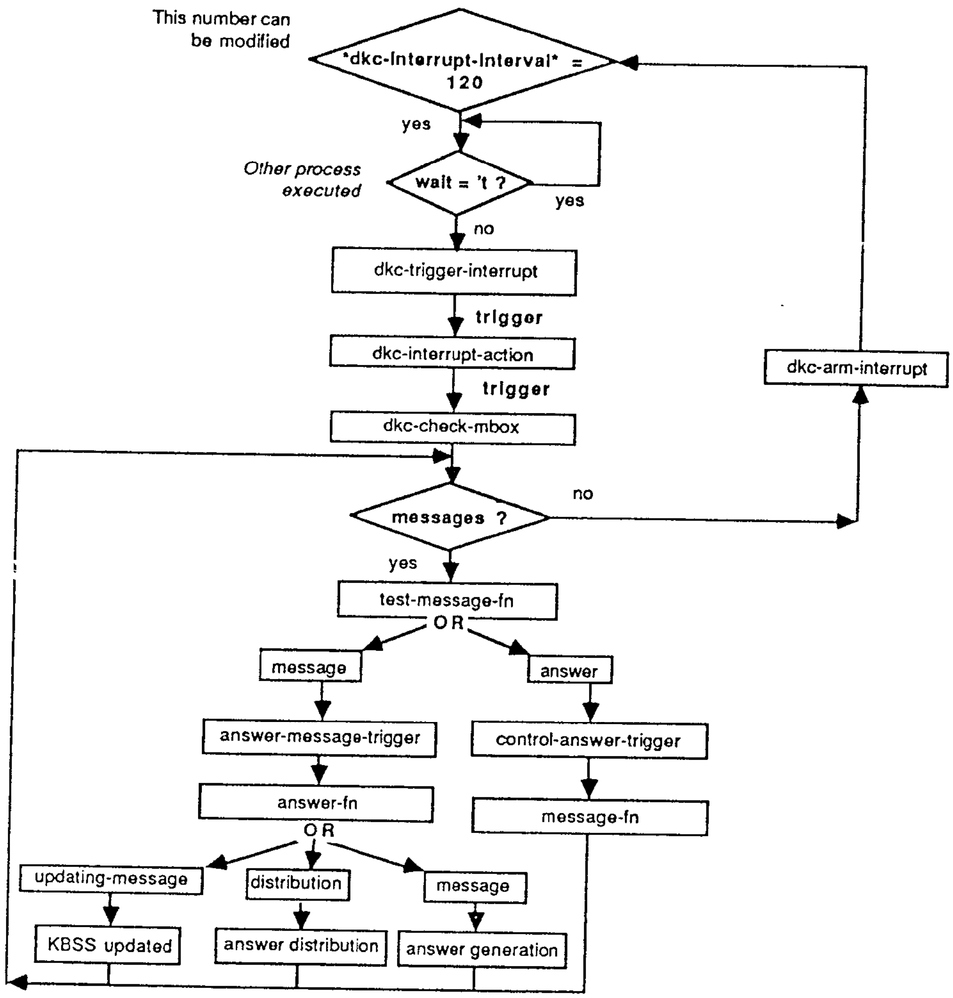

Figure 17.12
Message Receiving Sequence.

- *Blocking:* During the search for information or generation of an answer, the node's activity can be suspended until the end of this search or answer. The block is released when the action is completed. (The primitive function for blocking and unblocking has been defined at the network layer.)
- *Locking:* For shared schemata, i.e. updated globally by several EMN nodes, it is possible to lock their content in order to maintain consistency. This mechanism is triggered when an EMN node wants to read and then modify a shared schema's content. The schema lock is released once the read and modifications have been done through an updating message by the first EMN node requesting the schema. When a shared schema is locked, nodes that subsequently request the schema recieve an answer that the schema is locked. The node may then request an earlier version of the schema if desired.

Message generation is automatic. It can be synchronous or asynchronous depending on the EMN node implementation. Each message (for either updating, distribution or information search) or answer (in response to an information search message sent previously) is stored in the mail box of the EMN node. Periodically, this mail box is checked and its content evaluated. According to the nature of the messages (the four types enumerated previously), processes are triggered (figure 17.12).

Automatic information management maintains the consistency of an EMN node's knowledge throughout the EMN. We can identify two functions:

- Information updating (figure 17.13), and
- Information distribution (figure 17.14).

Information updating maintains consistency of schemata in the EMN. When the owner of a schema—or an EMN node authorized to modify a schema it doesn't own but shares—makes a modification, the owning EMN node generates a message to inform the users of the modification.

Each schema has a single owner, but may be read by other schemata if they possess the appropriate access rights. The schema owner *grants* access rights of other EMN nodes. Four kinds of rights exist: select, insert, delete and update. This is true for all schemata except those *shared* by two or more EMN nodes. Shared means they can be read and modified by several EMN nodes, but update messages are generated by the single owner.

The information distribution function sends new schemata to EMN nodes that are potentially interested. The questions are what, when and to whom should schemata be distributed? To solve this problem, we define a user callable function: distribution-function. This function identifies, according to the types of the new schemata created,

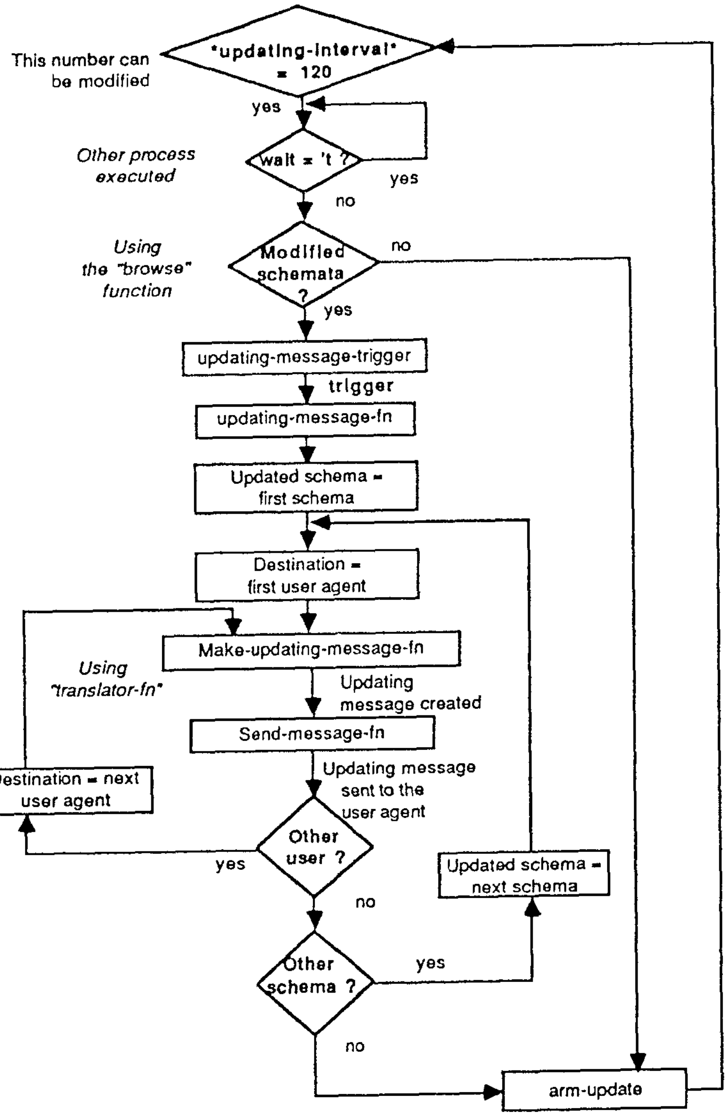

Figure 17.13
Updating Message Generation Sequence.

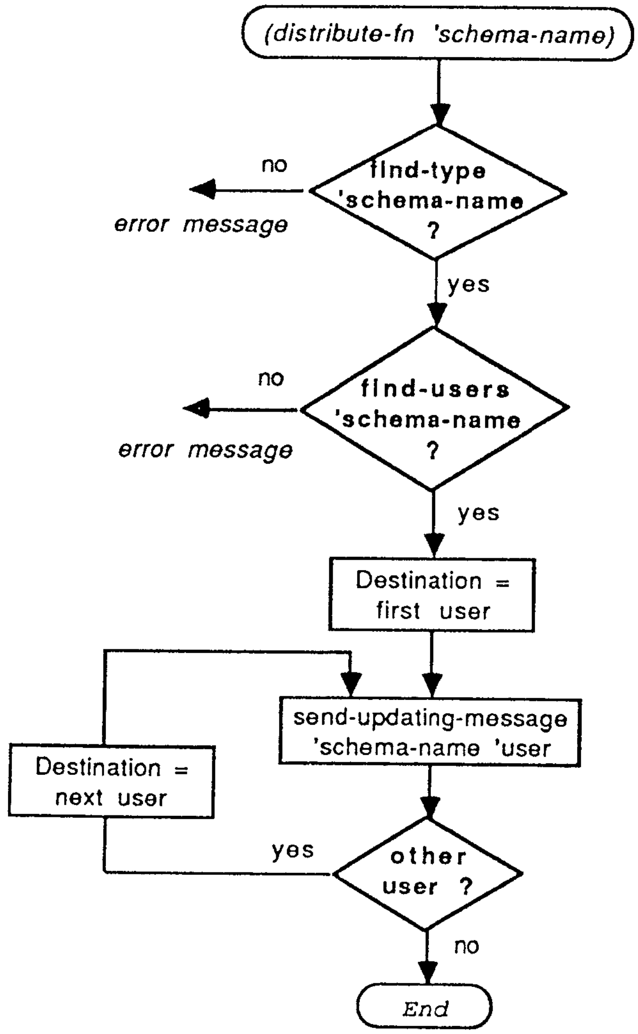

Figure 17.14
Information Distribution Sequence.

who are the "potential" recipients. Types of information of interest to EMN nodes are maintained in a taxonomic hierarchy. For each class, a set of keywords are used to define the class; that is, schemata that match the keywords are members of the information class. Each class also has a list of EMN nodes that are interested in the information class. If a recently created schema matches a particular information class, the schema is distributed to the EMN nodes interested in the class and to any other EMN nodes interested in classes above it in the hierarchy.

17.5.2 Information Layer Example

At this level, the main elements which are added on the previous figure 17.10 are the communication protocols and functions (figure 17.15):

17.6 Organization, Coordination and Market Layers

The first three layers of the EMN have been implemented and tested. In this section we briefly describe our goals for the organization, coordination and market layers. Investigations are underway at each of these levels (e.g., [35]).

17.6.1 Organization Layer

To ensure coherence and coordination of decisions and actions within the manufacturing system, we must consider the goals and the roles of each EMN node. The Organization Layer provides the modeling primitives for defining EMN node's structure, goals, roles, authority, etc.

There are many ways in which an organization can be modeled [23, 10, 27]. The 3 main aspects are:

- Physical: All the physical resources of the organization such as the machines, the personnel, the tools, etc.
- Decisional: All activities related to the control of the physical activities which need decisions. The main caracteristic of these activities is the multiple choices possibility.
- Informational: The control of activities by the decisional system is done by exchanging information between these 2 systems and also by internal communication. This aspect also includes all activities performed automatically (for example the Material Requirement Planning).

The modeling of an organization can be done by the modeling of these systems. For

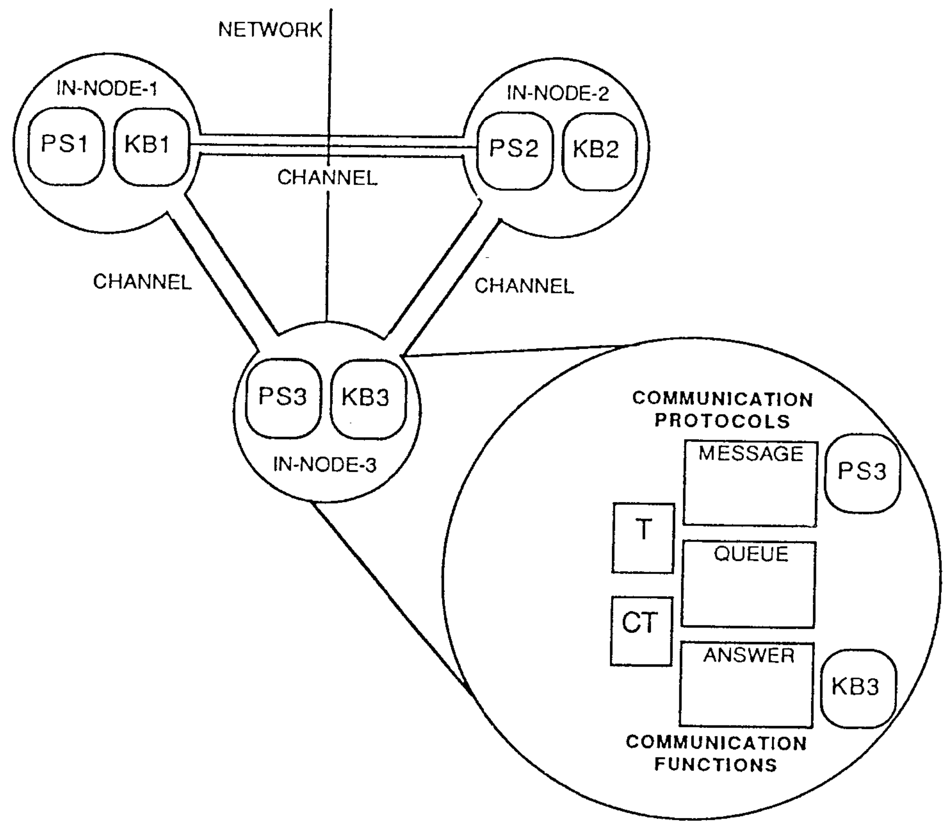

Figure 17.15
Information Layer Implementation Example.

that purpose, we can use some existing graphical tools such as the GIM tools (GRAI-IDEF0-MERISE) [28, 29].

To describe a manufacturing organization, we use the GIM graphical tools. These graphical tools support the modeling of different aspects of a manufacturing system. We use a grid as the primary graphical technique. This graphical tool is a two dimensional array which has on its x-axis different production management functions and on the y-axis a temporal decomposition using the horizon/period parameters. The horizon is the valid duration of a decision (for example: I make a schedule for 3 weeks, the horizon is 3 weeks). The period defines the recurrence of the decision (for example: I make this schedule each Monday, the period is one week).

Each "box" of the grid represents a "decision center" which could be an EMN node. On this grid, we indicate the interactions between the EMN nodes such the informational links (exchanges of information between the EMN nodes) and the decisional links (decision frame) which describe the hierarchy and goals transmission between the EMN nodes. A decision frame contains the goals, the decision variables (elements the EMN node can modify to perform its activity, for example: quantities, resources, time) and the rules to perform specific decision activities. All these elements are the basis for supporting the organizational layer. These elements are described in [28].

17.6.2 Coordination Layer

The Coordination Layer provides the protocols for the problem solving activities between the EMN nodes of the decentralized system. It defines the protocols for coordinating the activities of the EMN nodes through negotiation and cooperation mechanisms [9, 24, 18, 7, 14, 8, 2, 4, 33, 34].

17.6.3 Market Layer

The Market Layer allows for the incorporation of different kinds of distributed architectures. It supports the distribution of tasks and the negotiation of change. This layer allows for the answer of questions such as: how do we deal with the enterprise environment? (such as subcontractors, consumers, suppliers, etc.). It provides the strategies for dealing with the environment.

17.7 Applications

The Enterprise Management Network has been tested in a number of integration tasks within the CARMEMCO factory. CARMEMCO grew out of need to provide a single engineering and manufacturing problem that can integrate our research efforts in engineering and manufacturing decision systems. Our short term goal for CARMEMCO was to define a *virtual company,* existing solely in the computer, to support the exploration of issues in design, planning, and scheduling. Our long term goal has been to bootstrap the implementation of an on-campus factory which will serve as a development- and test-bed for research projects in robotics and automation of manufacturing. Consequently, the criteria for selecting a product that CARMEMCO will produce included:

- It would provide a testbed for primarily mechanical design, with the opportunity for electrical and electronic design, domain that involves description and manipulation of 3D objects which have interesting but not too detailed design features.
- It could be designed to be as simple or complex as desired.
- Components would have to be fabricated and assembled so that planning and scheduling research could explore both.
- Components could be made out of a variety of materials, both mundane and exotic.
- A variety of resources and processes which provide complex challenges for process planning, facility layout, and scheduling systems.
- Components could actually be fabricated at CMU or purchased externally.
- Students and faculty at CMU would want to purchase it.

Desk lamps were selected with these criteria in mind. Lamp components fit the design criterion quite well, as many are relatively simple, but all have at least a few interesting and unique features. For example, some arm components are straightforward hollow cylinders, while some base and head components are irregular polygons in 3D. With respect to the materials criterion, lamp components can be metal, hard plastic, soft plastic, wire, and foam. Some components can actually be either metal or plastic. With respect to the material and process variety criteria, lamp manufacturing requires purchasing, fabrication, assembly, subcontracting, non-destructive testing, packing, and distribution, as well as front end marketing and sales operations. The resources for these processes are large in number and type, as well as diverse in their operational and maintenance needs. Parts can be produced either in batches or on an individual basis. Major lamp components are heads, arms, and bases. Three styles of each are produced, with a standard interface between base/arm and arm/head components. The parts mix is achieved with a mix-n-match of these major components.

In the remainder of this section two applications of the EMN are described.

17.7.1 Integrating Production Planning Tasks

Within the Production Planning Laboratory of the Center for Integrated Manufacturing Decision Systems, we have used the EMN to integrate the facility layout and production simulation functions. We implemented a network containing the following nodes:

1. Facility Layout Decision Support System (DSS).
2. Production Analysis Decision Support System.
3. Production Planning model library.

The facility layout DSS provides the user with the ability to interactively lay out a production facility by iteratively allowing them to assert, refine and retract constraints (At the heart of the DSS lies a spatial planning algorithm that utilizes constraint-directed reasoning to construct equivalence classes of layouts that satisfy a set of constraints [3]). The following defines the steps in a layout design session and how the DSS utilizes EMN functions.

1. The designer specifies a set of products to be produced by the facility and a set of spatial perimeter constraints for the facility.
2. The layout DSS then attempts to determine which machines should be located near each other based on the proportion of orders that transit directly from one machine to another. The information necessary to compute this includes: process plans and forecasted orders. Since this information is not available initially in this DSS's knowledge base, the attempt to access the process plan of each order causes a fault which initiates the node's Information Management function. Since, the designer defined the products, there is no "owner" that the node may communicate. Therefore, it then searches its schema taxonomy for a node in the network that may contain the requested schemata. In this example, the planning model library is defined in the taxonomy as possessing the process planning information, and the information manager communicates directly with the model library node to get the information. The Layout DSS then computes the nearness constraints.
3. The designer specifies any additional spatial constraints.
4. At this point, the Layout DSS can proceed to lay out the factory. As the spatial algorithm executes, it tries to access machine dimensions. Again, the information is not locally available and the Information Manager in the node is initiated. Since the process plan schemata were owned by the planning model node, the Information Manager decides to ask the model node for the new information. This request also succeeds.

5. At this point the information necessary for the layout planner to proceed is available locally. The designer iterates through a number of designs until a set of designs are identified as worth retaining.
6. Using the Layout DSS interface, the designer designates a set of layouts to distribute to other nodes and initiates it automatic distribution.

The production analysis DSS uses knowledge-based simulation [26] to measure the effectiveness of a factory layout and process plans. It uses Information Layer functions in a similar manner as the Layout DSS.

17.7.2 Distribution Production Control

The second experiment with the EMN focused on supporting the exchange of information among 4 scheduling agents, an inventory agent and an agent that provides decision support to the factory manager. In this experiment, each scheduling agent was responsible for a single resource for which it maintained a schedule and dispatched work to. The inventory agent was responsible for maintaining information on orders and their process plans, and the manager's DSS was responsible for providing a global view of the factory schedule.

Given a set of pre-assigned activities for each scheduling agent, the manager's DSS would display a GANTT chart of all the resources and the time at which activities were performed on each. When the DSS attempted to construct the GANTT chart it would generate an error because it did not have a schema for each resource stored locally. The search manager would handle the error by broadcasting a message to all other nodes in the EMN requesting information for each of the resources. Since each resource was managed by a separate scheduler, it received only one response for each resource request. Upon receipt of each, it updated its tables to reflect where each resource was "owned." Once the information was received, a gantt chart display of all the resources' schedules was constructed and displayed. Now that each scheduler was aware that the manager DSS was interested in their resource schemata, they would distribute changes to it whenever they occur via the information distribution function.

The inventory manager agent was responsible for maintaining the general information, such as due dates, quantity, etc. for each order. When changes to this information occured, the inventory manager would use the automatic distribution function to distribute the changes to the scheduling agents that need to know.

17.8 Conclusion

The Enterprise Management Network is designed to facilitate the integration of heterogeneous functions distributed geographically. Integration is supported by having the network first play a more active role in the accessing and communication of information, and second by providing the appropriate protocols for the distribution, coordination and negotiation of tasks and outcomes.

As described in this chapter, the Data layer provides the ability to perform "standard" SQL-like queries across the network. The Information layer provides a node with the ability to "invisibly" access information anywhere in the network without explicitly referring to its location or methods of retrieval.

Our design of the Enterprise Management Network has a number of characteristics:

- *Modular layered architecture* as we have defined six levels of descriptions for a decentralized system, we can implement in a specific case, a part or the entirety of the architecture.
- *High level decentralized communication system* which flexibly support cooperative decision making: our structure includes a decentralized communication system which, using the frame based structure, allows for the exchange of information (i.e., schemata) between decentralized agents.
- *User transparency*: the decentralized communication system is implemented in each agent of the decentralized system. It has the capability to provide the needed information to the agent. As this communication system is not specific to a particular agent, it has been defined as a shell. The agent need not know where to get the needed information. The communication system has the rules and capability to play this role. In our specific implementation of this communication system, the trigger of the information search is the CRL (Carnegie Representation Language (CRL) is a registered trademark of Carnegie Group Inc.) command: (GET-VALUE 'schema-name 'slot-name) (For example: (GET-VALUE 'machine 'capacity) in this case "machine" is the schema name and "capacity" is the slot name. If the value is available in the knowledge base of the agent, it is returned. If not one or several messages are generated by the decentralized communication system).
- *Declarative layer* specification provided by the frame based representation. Each agent has its own local knowledge and data bases.
- *Accessibility* of information to different parts of the organization. Each agent has translation mechanisms to enable communication with others.
- *Understandability* of information through a common communication language.

- *Awareness* of problems and communication to appropriate agents using a communication schema.
- *Focussed* information dissemination.
- *Responsiveness* of agents through rules and translation mechanisms.
- *Flexibility* of communication due to support for many types of interaction and of representation through a frame based representation.

The three lower levels of this architecture have been implemented and are described in more detail in [31, 30], and the upper layers are described in [32].

Acknowledgements

The research described herein was performed at the Center for Integrated Manufacturing Decision Systems of Carnegie Mellon University while Dr. Roboam was a visiting scientist and Prof. Fox was Director of the Center. The research was sponsored, in part, by the Aerospatiale Co. (France) and the Defense Advance Research Projects Agency under contract #F30602-88-C-0001.

Katia Sycara, Chuck Marshall, and the rest of the members of the CORTES and CARMEMCO projects have contributed to the development of the Enterprise Management Network.

References

[1] Adler, M. R., and Simoudis, E. Integrated Distributed Expertise. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence*. Bandera, Texas, 1990.

[2] Barr, A., Cohen, P. R., and Feigenbaum, E. A. *The Handbook of Artificial Intelligence, Volume 4*. Addison-Wesley Publishing Company, Massachusetts, 1989.

[3] Baykan, C. A., and Fox, M. S. Constraint Satisfaction Techniques for Spatial Planning. Intelligent CAD Systems 3: Practical Experience and Evaluation. In ten Hagen, P., and Veerkamp, P., Springer-Verlag, 1990.

[4] Bond, A. H., and Gasser, L. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, 1988.

[5] Date, C. J. *An Introduction to Data Base Systems*. Addison-Wesley Publishing Company, Massachusetts, 1981.

[6] Date, C. J. and White, C. J. *A Guide to SQL DS*. Addison-Wesley Publishing Company, Massachusetts, 1989.

[7] Decker, K, and Lesser, V. A Scenario for Cooperative Distributed Problem Solving. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence.* Bandera, Texas, 1990.

[8] Durfee, E. H., and Lesser, V. R. Using Partial Global Plans to Coordinate Distributed Problem Solvers. In *Proceedings of 10th International Joint Conference on Artificial Intelligence.* Milan, Italy, 1987.

[9] Durfee, E. H., and Montgomery, T. A. A Hierarchical Protocol for Coordinating Multiagent: An Update. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence.* Bandera, Texas, 1990.

[10] Erkes, K, and Clark, M. Public domain report number 1. Technical Report, ESPRIT Project 418, Open CAM System, April 1987.

[11] Fox, M. S. An Organizational View of Distributed Systems. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-11(1):70-80, 1981.

[12] Fox, M. S., and Sycara, K. Overview of the CORTES project: a Constraint Based Approach to Production Planning, Scheduling and Control. In *Proceedings of the Fourth International Conference on Expert Systems in Production and Operations Management,* May, 1990.

[13] Gallois, P. M. De la tenue des stocks au pilotage synchronise de l'ensemble des flux. *Prefntation aux Journees AUGRAI,* 1988.

[14] Gasser, L., and Huhns, M. N. *Distributed Artificial Intelligence,* Volume II. Pitman Publishing & Morgan Kaufmann Publishers, 1989.

[15] Gasser, L., Braganza, C., and Herman, N. MACE: A Flexible Testbed for Distributed AI Research. In Michael N. Huhns (editor), *Distributed Artificial Intelligence,* chapter 5, pages 285-310. Pitman Publishing & Morgan Kaufmann Publishers, 1987.

[16] Huhns, M. N. *Distributed Artificial Intelligence.* Pitman Publishing & Morgan Kaufmann Publishers, 1987.

[17] Huhns, M. N., Bridgeland, M. L., and Arni, N. V. A DAI Communication Aide. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence.* Bandera, Texas, 1990.

[18] Nein, M. Supporting Conflict Resolution in Cooperative Design Systems. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence.* Bandera, Texas, 1990.

[19] *Knowledge Craft* Carnegie Group Inc, Pittsburgh, 1985.

[20] Lesser, V. R. Cooperative Distributed Problem Solving and Organization Self-Design. *SIGART Newsletter* 46, October, 1980.

[21] Lusardi, F. *The Database Experts' Guide to SQL.* Mc Graw Hill, New York, NY, 1988.

[22] Mullender, S. *Distributed Systems.* ACM Press, New York, N.Y., 1989.

[23] Panse, R. CIM-OSA–A vendor independent CIM architecture. In *Proceedings of CIMCON '90,* pages 177-196. Gaithersburgh, MD, 1990.

[24] Parunak, H. V. D. Toward a Formal Model of Inter-Agent Control. In *Proceedings of 10th International Workshop on Distributed Artificial Intelligence.* Bandera, Texas, 1990.

[25] Ram, S., Carlson, D., and Jones, A. Distributed Knowledge Based Systems for Computer Integrated Manufacturing. In *Proceedings of CIMCON '90,* pages 334-352. Gaithersburgh, MD, 1990.

[26] Reddy, Y. V., Fox, M. S., Husain, N., and McRoberts, M. The Knowledge-Based Simulation System. *IEEE Software:* 26-37, March, 1986.

[27] Roboam, M., Doumeingts, G., Dittman, K, and Clark, M. Public domain report number 2. Technical Report, ESPRIT Project 418, Open CAM System, October 1987.

[28] Roboam, M. Reference models and analysis methodologies integration for the design of manufacturing systems (in French: Modeles de reference et integration des methodes d'analyse pour la conception des systemes de production). PhD thesis, Laboratoire GRAI, Universite de Bordeaux, Bordeaux, France, 1988.

[29] Roboam, M., Zanettin, M., and Pun, L. GRAI-IDEF0-MERISE (GIM): integrated methodology to analyse and design manufacturing systems. In *Computer-Integrated Manufacturing Systems,* Volume 2 Number 2, pages 82-98. Westbury House, Guilford, UK 1989.

[30] Roboam, M., Fox, M. S., and Sycara, K Enterprise Management Network Architecture—Distributed Knowledge Baf support. Technical Report CMU-RI-TR-90-21, CIMDS, Carnegie Mellon University, Pittsburgh, PA, 1990.

[31] Roboam, M., and Fox, M. S. Distributed communication system: user manual. Technical Report, CIMDS, Carnegie Mellon University, Pittsburgh, PA, 1990.

[32] Roboam, M., Fox, M. S., and Sycara, K Enterprise Management Network Architecture—The Organization Layer. Technical Report CMU-RI-TR-90-22, CIMDS, Carnegie Mellon University, Pittsburgh, PA, 1990.

[33] Sycara, K. Resolving Goal Conflicts via Negotiation. In *Proceedings of the Seventh National Conference on Artificial Intelligence* [AAAI-88]. 1988.

[34] Sycara, K and Roboam, M. Intelligent Information Infrastructure for Group Decision and Negotiation Support of Concurrent Engineering. In *Proceedings of the 24th Hawaii International Conference on System Sciences.*1991.

[35] Sycara, K, Roth, S., Sadeh, N., and Fox, M. S. Distributed Constrained Heuristic Search. *IEEE Transactions on Systems, Man and Cybernetics,* 1991. to appear.