Roboam, M., and Fox, M.S., (1990), "Intelligent Networking: Towards Integrating the Manufacturing Enterprise", Proceedings of The Fourth International Conference on Expert systems in Production and Operations Management, Hilton Head Island, SC.

INTELLIGENT NETWORKING: TOWARDS INTEGRATING THE MANUFACTURING ENTERPRISE

Michel Roboam[1] and Mark S. Fox[2]

Center for Integrated Manufacturing Decision Systems
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

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 paper we introduce the concept of Intelligent networking 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 Intelligent 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. In this paper, we describe the first three layers.

1.0 Introduction

This paper presents the architecture, the elements and the organization of an Intelligent Network (IN) to support the integration of the Manufacturing Enterprise. Optimizing manufacturing can only be achieved by greater integration of activities throughout of the production life cycle. Integration must not only address the issues of shared information and communication, but how to coordinate decision and activities throughout the firm.

If we look back in the history of the automation of Manufacturing systems, we can distinguish four major steps [4]:

To 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.

To manufacture what will be sold: to reduce the inventories, companies tried to determine market demand by means of forecasting. Then, the manufacturing activity was planned accordingly. 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.

---

[1]Michel ROBOAM is currently visiting scientist in the Center for Integrated Manufacturing Decision Systems and is sponsored by the AEROSPATIALE Company (France).

[2]This research has been supported, in part, by the Defense Advance Research Projects Agency under contract #F30602-88-C-0001.

Appeared in the Proceedings of the Fourth International Conference on

Integrated manufacturing: the manufacturing enterprise must be viewed more broadly. We must include in this activity all the stages between the raw material acquisition up to the distribution and servicing of the finished products. Large 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, ... They constitute the MRP II systems. These systems provide little flexibility.

To manufacture what is sold: the CIE (Computer Integrated Enterprise) concept increases the response capacity of the 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 in order 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.

An intelligent network is viewed 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 provided by the Intelligent Network:

- **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: transfering information to groups where the sender and the receivers are pre-defined.
  - Dynamic routing: transfering 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 IN will support the providing of feedback in two modes:
  - Pre-define 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 INP 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 Intelligent Network is divided into six levels:

---

6. Market Layer

5. Coordination Layer

4. Organization Layer

3. Information Layer

2. Data Layer

1. Network Layer

The *Network Layer* provides for the definition of the network architecture. At this level, the nodes (IN-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").

The *Data Layer* provides for queries and responses to occur between nodes in a formal query language patterned after SQL [1] [2].

The *Information Layer* provides "invisible" access to information spread throughout the IN. 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.

The *Organization Layer* provides the primitives and elements (such as goal, role, responsibility and authority) for distributed problem solving. It allows automatic communication of information based upon the roles a node plays in the organization. Each IN-node knows its responsibilities, its goals, and its role in the enterprise organization.

The *Coordination Layer* provides the protocols for the problem solving activities between the IN-nodes of the decentralized system. It defines the protocols for coordinating the activities of the IN-nodes through negotiation and cooperation mecanisms.

The *Market Layer* provides the protocols for coordination among organizations in a market environment. It supports the distribution of tasks and the negotiation of change and the strategies to deals with the environment. This layer allows to answer this question: "how do we deal with the enterprise environment (such as the subcontrator, the consumer, the suppliers, ...)?"

In the rest of the paper, the information routing capabilities of the IN are described, starting with a description of the architecture of a network node (IN-node), and followed by a a description of the first three layers of the IN. (The implementation of this architecture and of the communication system is described in [7]).

2.0 Intelligent Network Node

The Intelligent Network links together two or more application nodes (IN-node) by providing the "glue" that integrates the Manufacturing enterprise through architectures and mechanisms to support decision making at all levels of the organization. For example, the CORTES system [3] is composed of an uncertainty analyser, a planner, a scheduler, a factory model and dispatchers responsible of several machines (figure 2-1). Each is defined as an IN-node.

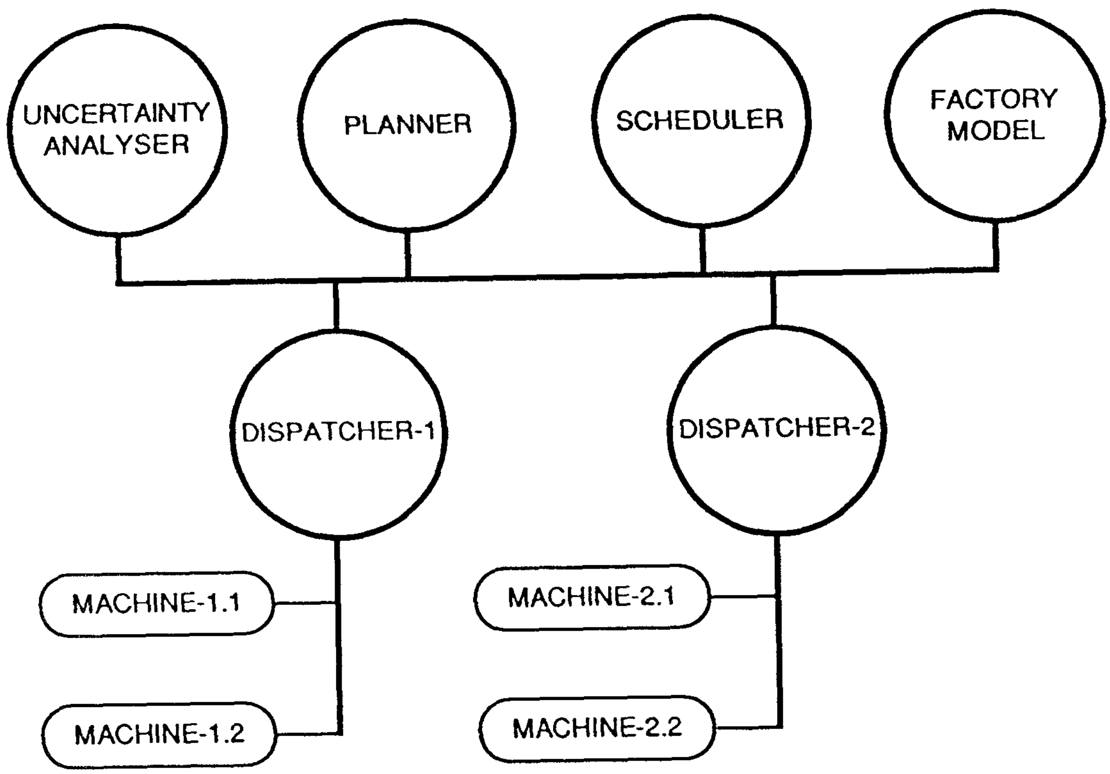

Figure 2-1: Example of decentralized system

Each IN-node consists of the following subsystems[3]:

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

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

Each IN-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 CRL schemata [5].

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 IN-nodes. In the Intelligent 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 IN-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 IN-node and a copy is stored locally.

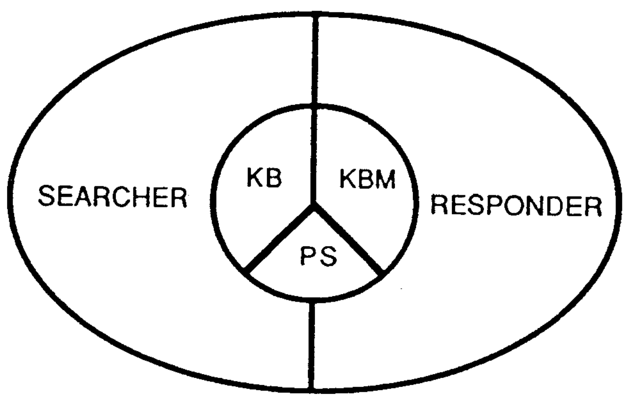

Figure 2-2: The elements of an IN-node

A problem that arises in supporting the exchanges between the problem solving subsystem and the knowledge base, is the inavailability of schemata locally. The problem solver often refers to knowledge that cannot be found locally, but may be found in another IN-node's knowledge base. At the time of reference, the problem solver may or may not know where in the intelligent 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 IN-nodes. To accomplish this, the IN-node has as part of it a **Communication Manager**. It manages the both the search for information in the IN and responds to like requests from other IN-nodes. To perform these

---

[3]Currently implemented in CommonLISP

activities, the Communication Manager has two modules:

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

Figure 2-2 summarizes the structure of an IN-node.

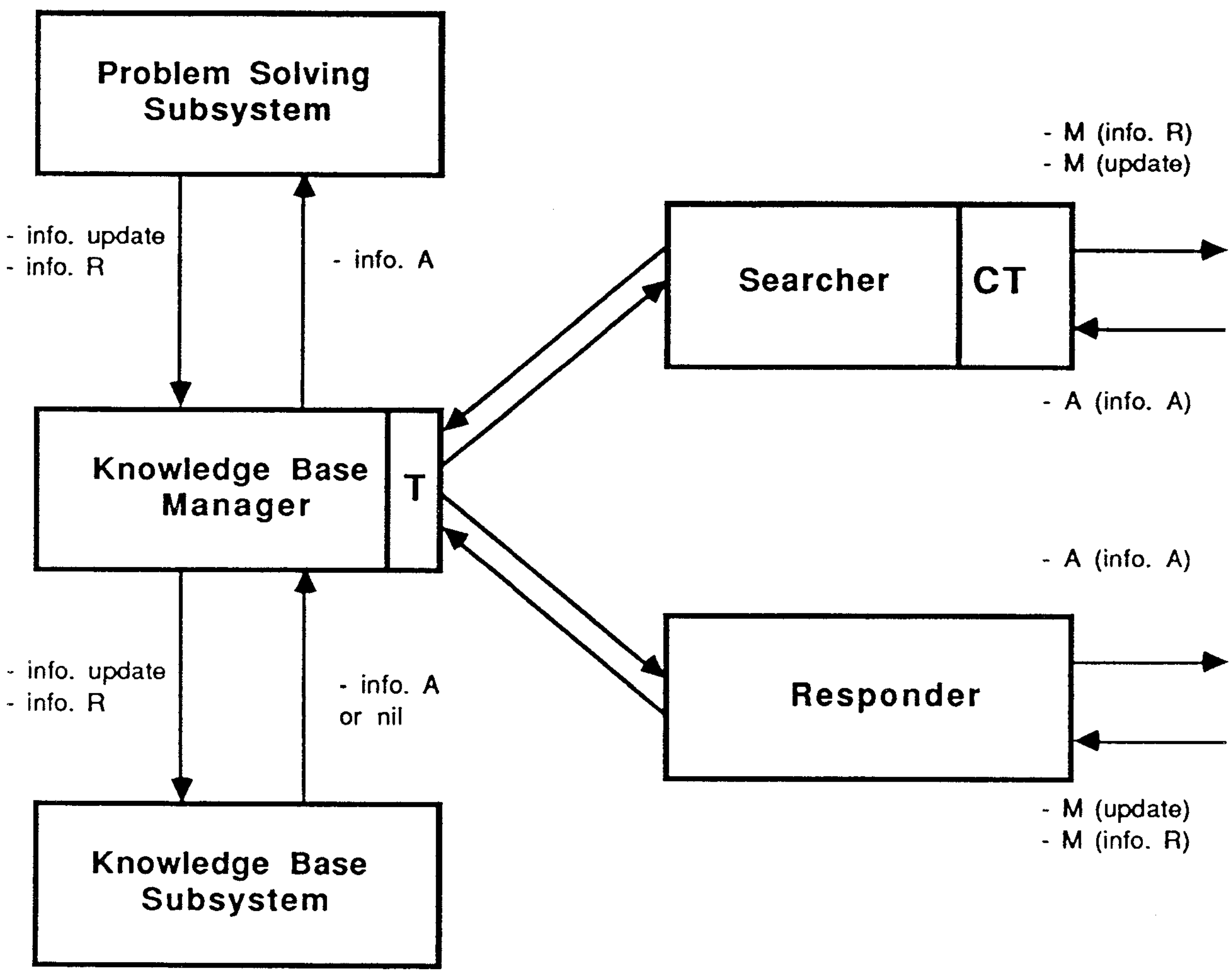

Figure 2-3: Information exchanges overview

The communication manager manages four types of interaction:

- ***Triggering***: information that trigger the node's processing.
- ***Dynamic retrieval***: Requests for information not available in itsKnowledge base and necessary to perform its task. These information needs appear during the internal activity (processing) of a IN-node.
- ***Updating information***: When an IN-node, owner of some schemata, modifies these schemata, the searcher dispatches the modifications to other IN-nodes that have local copies of these schemata. The responder may or may not update a local copy depending

on the usage at the receiving IN-node. Owner of a schema means, the IN-node is the only one allowed to modify globally the content of a schema. But each IN-node having 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 IN-node in the figure 2-3. This figure shows the different 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 Correspondance Table). We will discuss in the next sections the content of these transactions.

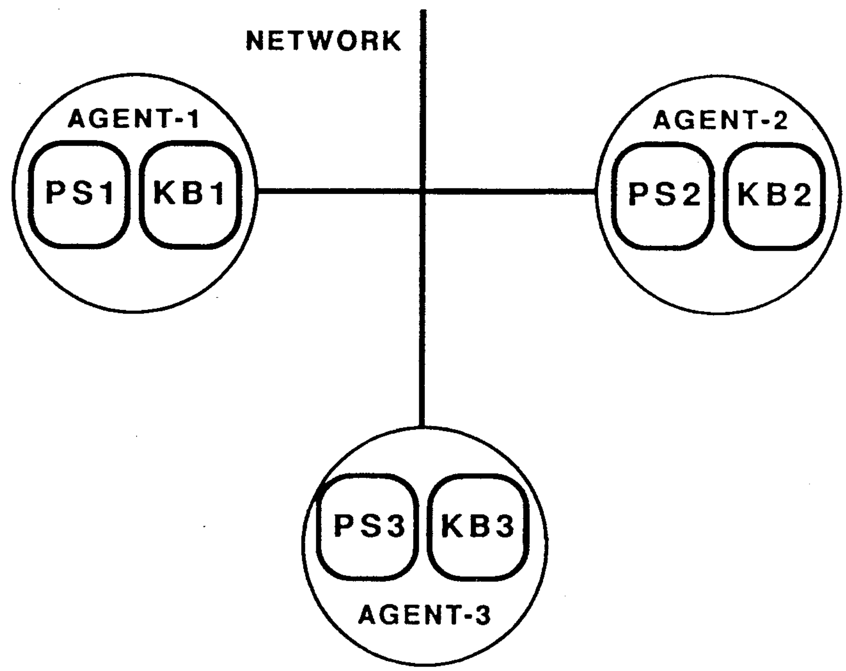

Figure 2-4: Decentralized system example

The first three layers of the Intelligent Network architecture are defined in the remaining sections. Each layer, provides further detail on the functionality and operation of IN-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) (figure 2-4). We describe in this first figure, an empty decentralized system, e.g., without the Intelligent Network. We will use this example by adding at each level specific elements, functions and protocols.

3.0 Network Layer

The architecture presented in this document covers only the higher levels of a network protocol. We assume the existence of lower levels provided by a network such as DECNET[4].

The Network Layer defines the network structure:

[4]DECNET is a registered trademark of Digitel Equipment Corporation.

Schema 3-1: IN-node

IN-node		
SLOT	FACET	VALUE
Name	*Value:* *Restriction:*	type string
Network	*Value:* *Restriction:*	type network-name*
Domain-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*

- ***IN-nodes*** represent problem solving agents. They include the basic communication objects: mail box, semaphore box, queues and low level message. To support the IN-node initialization, we use different schemata such as the IN-node schema (schema 3-1).
- ***Channels*** define communication links between IN-nodes. Each channel is defined as an instance of the channel schema.
- ***Messages*** can be sent along channels between IN-nodes. During information transfer, an IN-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 IN-nodes.

Figure 3-2 defines the message sending process at the Network Layer.

A message can be sent from one IN-node to another only if the corresponding channel already exists, and the receivers mailbox is not locked. If the mail box is locked, the sending IN-node waits until this mail box is unlocked. Once the mailbox is unlocked, the following is performed:

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 function is provided: message reception (figure 3-2). This function allows the content of the mail box to be read. When it is triggered, it checks at first the mail box status. If it is locked, it waits until it is unlocked. The second steps of this function is performed in three phases: lock mail box, read mail box, and unlock mail box.

The example depicted by figure 2-4 is extended by adding the network layer on this empty structure. The main modifications, in bold, which occur on the structure defined in figure 3-3 are:

- The initialization of the IN-node, this includes the definition, for each agent of the decentralized system, of a name and of the mail box and semaphores.

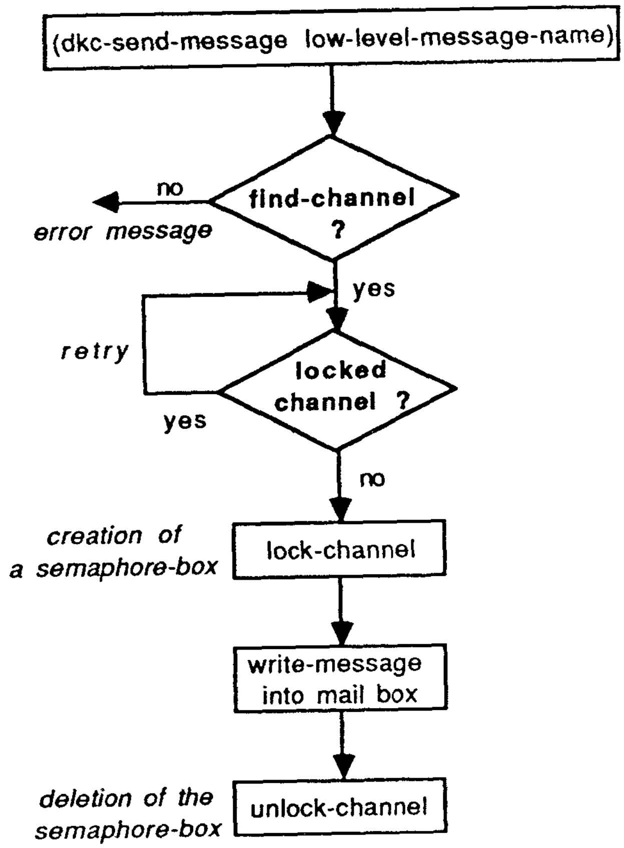

Figure 3-1: Message passing algorithm

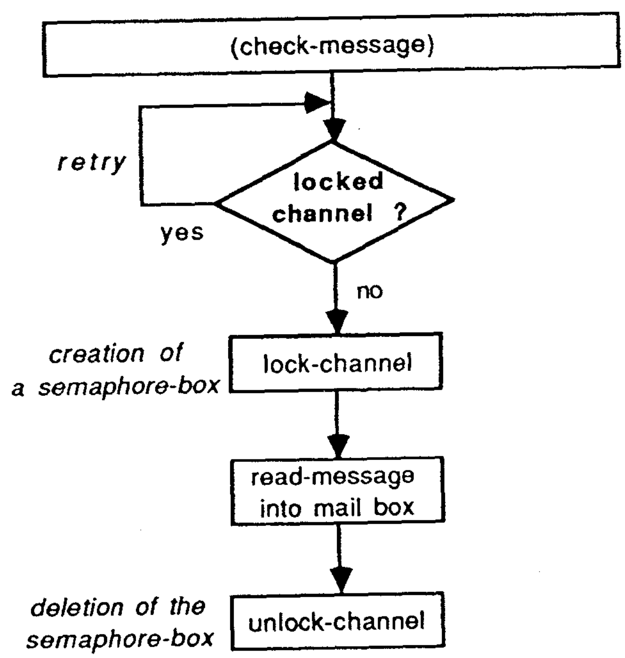

Figure 3-2: Checking mail box algorithm

• The creation of links between the IN-node through the utilization of channels. These channels also include the basic primitives for the message passing activity.

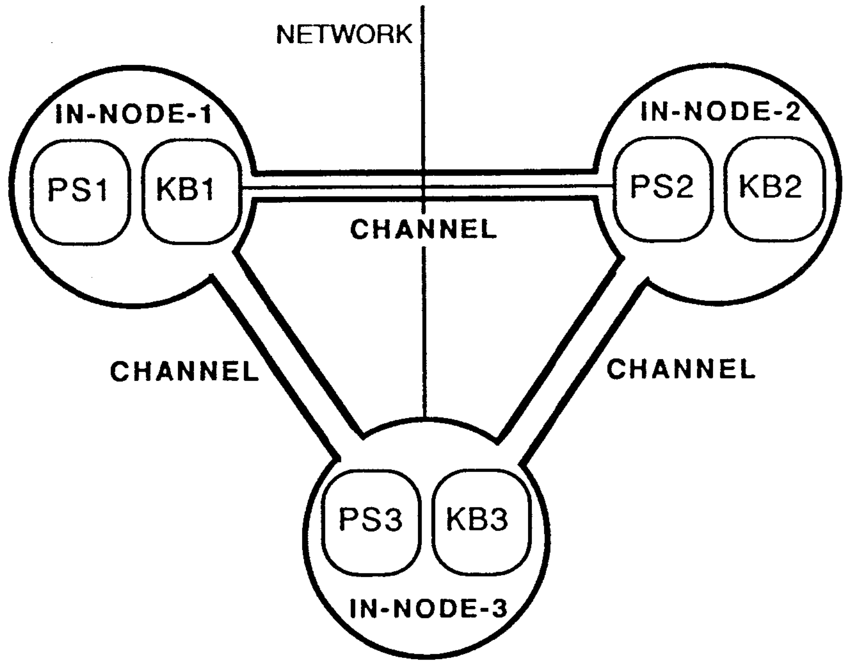

Figure 3-3: Network Layer implementation example

4.0 Data Layer

The Data Layer provides IN-nodes with the capability to explicitly request and send information, in the form of schemata, from/to other IN-nodes. The protocol for requesting and asserting information between IN-nodes is based on a subset of SQL [1, 6]. In this version, schemata correspond to tables, and slots correspond to fields. Protection is provided at the schema level; access to schemata may be locked and the requesting IN-node blocked until the schema in unlocked. Following is a simple example of slot retrieval query.

Example:

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

Result:

```
SELECT capacity
FROM machine-1;
```

```
{{ MACHINE-1
      CAPACITY: 100-p/h }}
```

As we can see, the objects for a more sophisticated communication capability has been added in each IN-node figure 4-1. These capabilities will be used at the upper levels for negotiation information distribution.

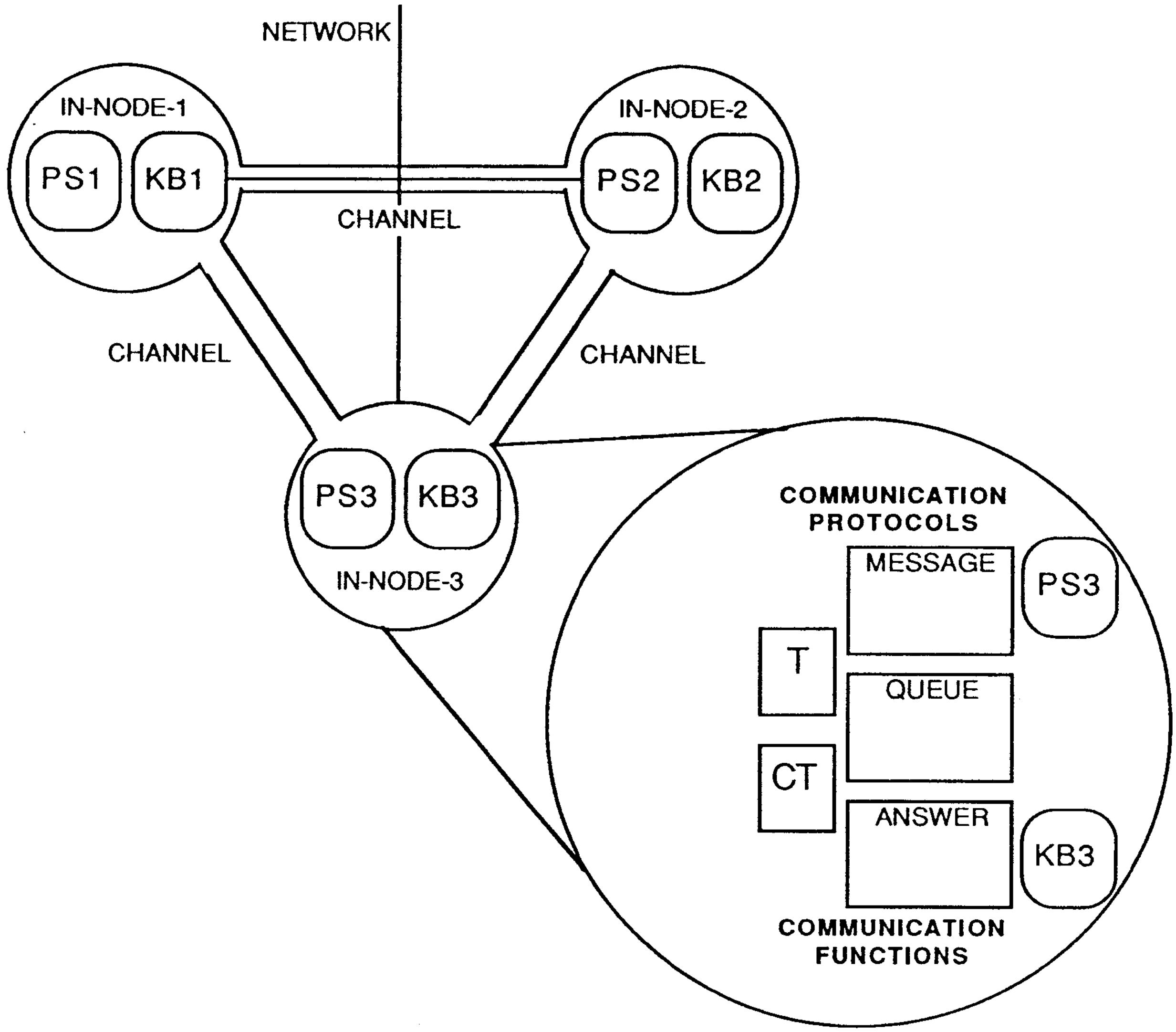

Figure 5-3: Information Layer implementation example

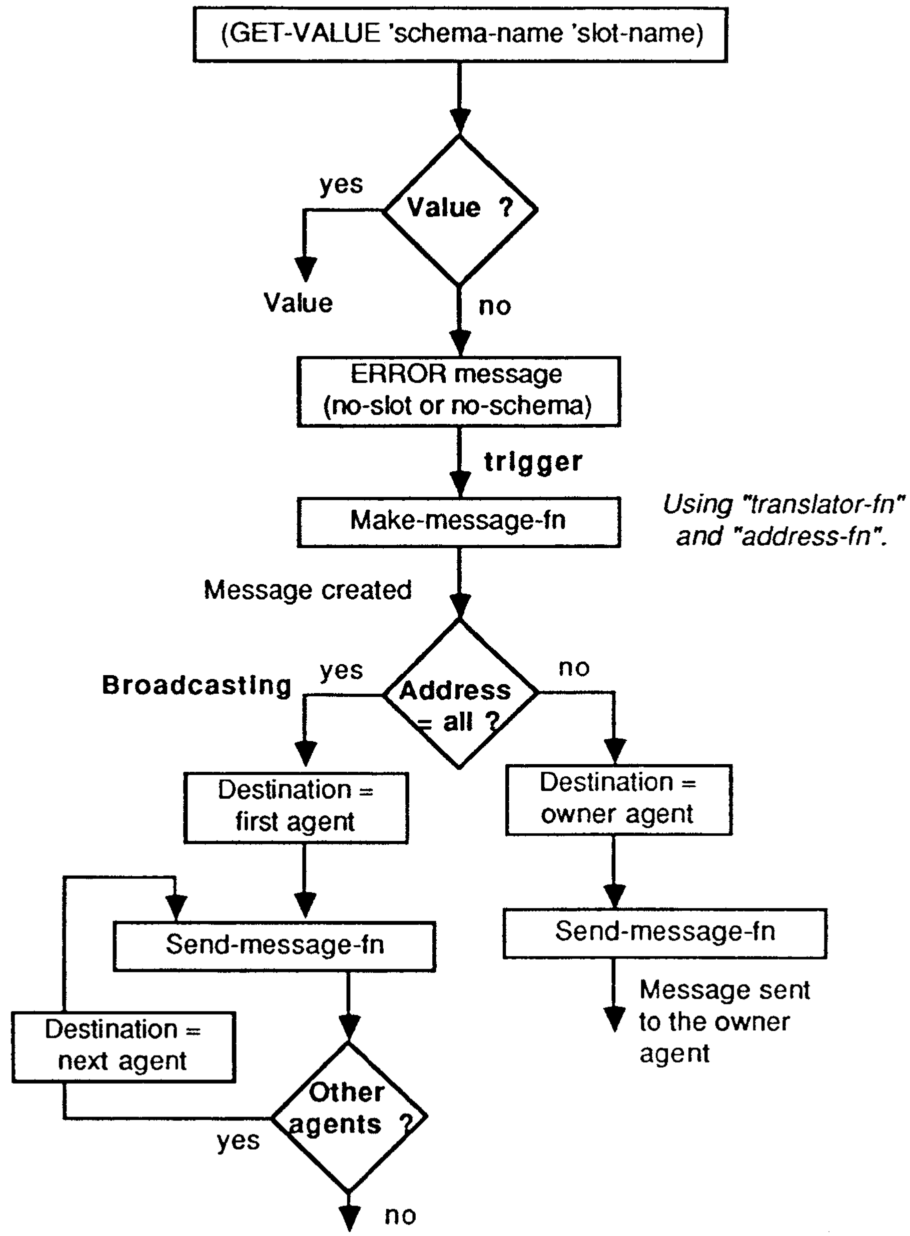

Figure 5-1: Message generation sequence

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

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

- Information updating (figure 5-2), and
- information distribution.

Information updating maintains consistency of schemata in the IN. When the owner of a schema, or an IN-nodes authorized to modify a schema they don't own but share, make a modification, the owning IN-node generates an 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 IN-nodes. Four kinds of rights exist: select, insert, delete and update. This is true of all schemata except those *shared* by two or more IN-nodes. Shared means they can be read and modified by several IN-nodes, but update messages are generated by the single owner.

The information distribution function send new schemata to IN-nodes that are potentially interested. The questions are what, when and to whom should schemata be distributed? To solve this problem, we defined a user callable function: distribution-function. This function identifies, according to the nature of the new schemata created, who are the "potential" recipients Types of information of interest to IN-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 IN-nodes that are interested in the information class. If a recently created schema matches a particular information class, the schema is distributed to the IN-nodes interested in the class and to any other IN-nodes interested in classes above it in the hierarchy (subsumed).

At this level, the main elements which are added on the previous figure 4-1 are the communication protocols and functions (figure 5-3):

6.0 Conclusion

The Intelligent Network is designed to facilitate the integration of a distributed set of heterogeneous functions. 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 distrubtion, coordination and negotiation of tasks and outcomes.

As described in this paper, the Data layer provides the ability to perform "standard" SQL-like queries across the network. The Information layer provides a node the ability to "invisibly" access and distribute information anywhere in the network without explicitly referring to its location nor its retrieval.

The specification and implementation of the three lower levels of this architecture has been implemented and are described in [8] and [7].

7.0 Acknowledgements

Katia Sycara, Chuck Marshall, and the rest of the members of the CORTES and CARMEMCO projects have contributed to their comments to the development of the network.

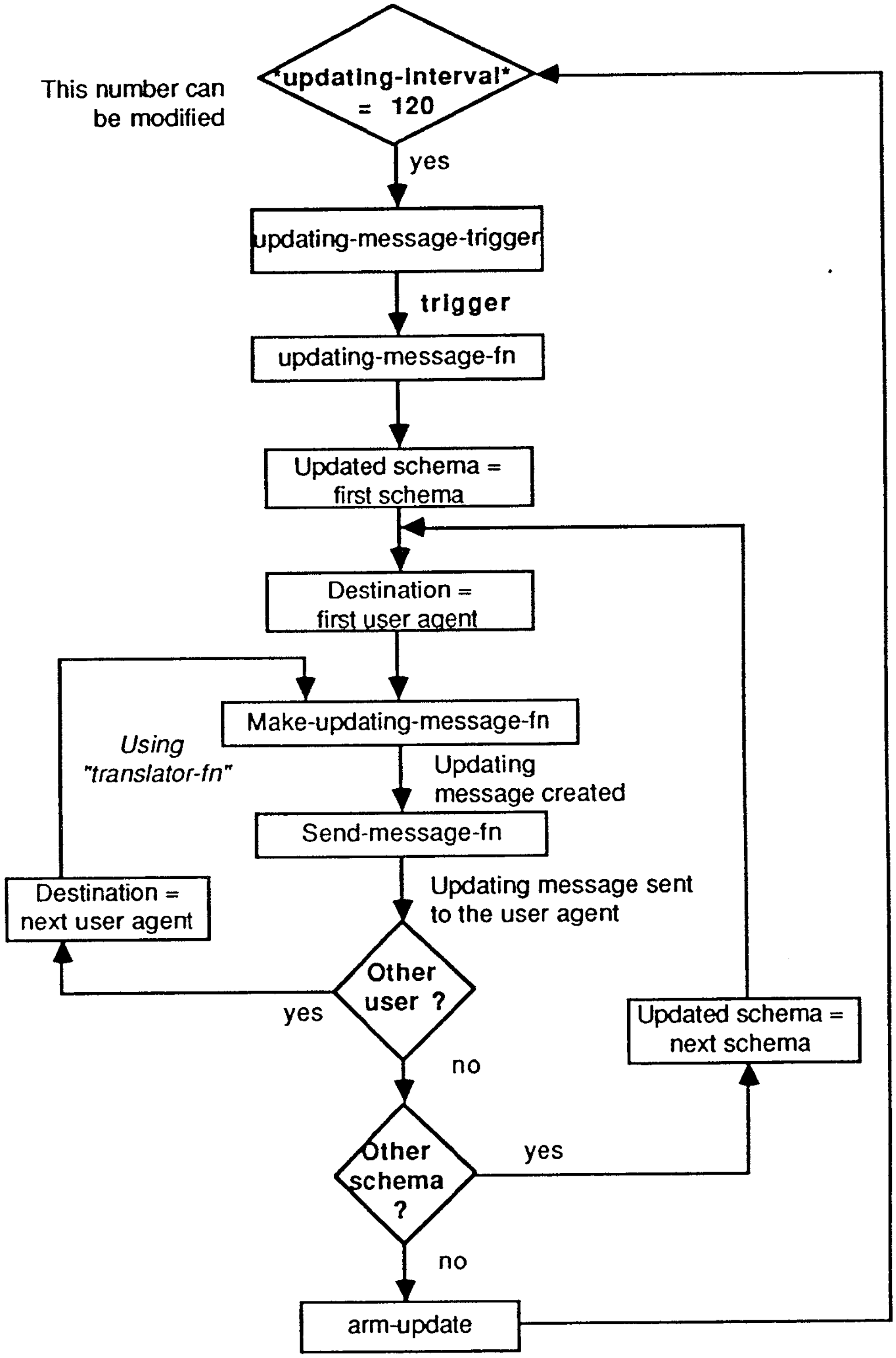

Figure 5-2: Updating message generation sequence

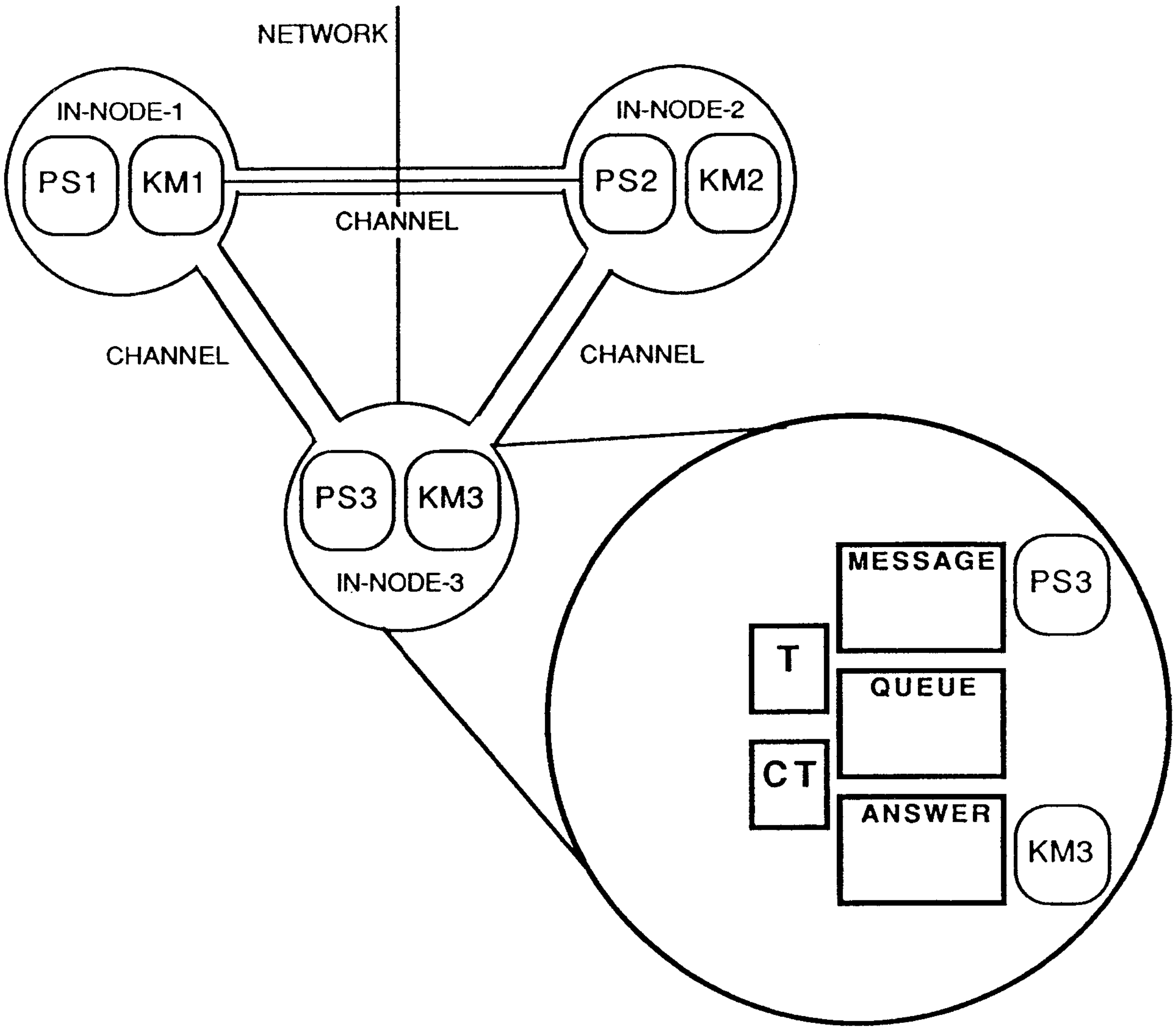

Figure 4-1: Data Layer implementation example

5.0 Information Layer

The Information Layer provides two services for the IN-node:

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

Automatic acquisition of information provides an IN-node with the capability to acquire at any moment and without knowledge of its location in the IN 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 generation sequence (information search sequence) (we define its algorithm on figure 5-1), and
- The message and answer reception (answer sequence).

References

[1] Date, C.J.
An introduction to data base systems.
Addison-Wesley Publishing Company, Massachusetts, 1981.

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

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

[4] P.M.Gallois.
De la tenue des stocks au pilotage synchronise de l'ensemble des flux.
Presentation aux Journees AUGRAI, 1988.
Universite de Bordeaux I, Bordeaux, France.

[5] **Knowledge Craft**
Carnegie Group Inc, 650 Commerce Court, Station Square, Pittsburgh PA 15219, 1985.

[6] Lusardi, F.
The database Experts' guide to SQL.
Mc Graw Hill, New York, NY, 1988.

[7] Roboam, M., and Fox, M.S.
Distributed communication system: user manual.
Technical Report, CIMDS, Carnegie Mellon University, PITTSBURGH, PA, 1990.
In preparation.

[8] Roboam, M., and Fox, M.S.
The CORTES Project - Intelligent Manufacturing Architecture.
Technical Report, CIMDS, Carnegie Mellon University, PITTSBURGH, PA, 1990.
In preparation.