The Metadatabase Approach to Integrating and Managing Manufacturing Information Systems*

Cheng Hsu[1], Gilbert Babin[2], M'Hamed Bouziane[3], Waiman Cheung [4],
Laurie Rattner[5], Alan Rubenstein[6], and Lester Yee[2]

* A preliminary version is presented to the
Third International Conference on
Computer Integrated Manufacturing, May 1992

January 1992

Revised June 1992

1 Associate Professor, Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute Troy, NY 12180-3590.
2 Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute Troy, NY 12180-3590.
3 Digital Equipment Corporation, Nashua, NH 03060.
4 Lecturer, Graduate School of Business Administration, Chinese University of Hong Kong, Shatin, NT, Hong Kong.
5 Assistant Professor, Anderson School of Management, University of New Mexico, Albuquerque, NM 87120.
6 Director, Computer Integrated Manufacturing Program, Center for Manufacturing Productivity and Technology Transfer, Rensselaer Polytechnic Institute Troy, NY 12180-3590.

Abstract

Integration has become a self-evident goal in today's manufacturing enterprises. Since the late 1970's, numerous major efforts have been launched worldwide to develop, employ, and deploy information technology for integration as a strategic weapon to compete in the global marketplace. Important results have been obtained, including Computer Integrated Manufacturing (CIM), and Concurrent Engineering/Re-engineering. The field is poised to tackle some fundamental barriers of integration and thereby effect a new quantum jump in overall productivity, where information will become a fourth basic factor of production along with land, labor, and capital. Among the barriers are major gaps in information technology regarding multiple systems operating concurrently over different geographical regions.

This paper discusses a unique approach to the integration problem regarding information resources management, global query and concurrent systems administration. This approach is based on the metadatabase model which incorporates both data and knowledge in order to accomplish information integration across manufacturing functions. Basic concepts, methods, and techniques have been developed for the approach and verified via a prototype metadatabase system. The system's primary components and major functionalities are discussed and illustrated in detail through the use of an extended example of a pilot manufacturing system comprising order processing, process design, and shop floor control. It will also provide the reader with a simulated system demonstration.

Keywords: Systems Integration, Manufacturing Information Management, Multiple Data and Knowledge Systems.

1. Manufacturing Information Integration

It is well known in the industry that information accounts for a major part of the integration problem for manufacturing, especially when adaptiveness and short cycle time are paramount requirements. To achieve information integration, however, new basic results are needed concerning data and knowledge engineering, enterprise information management, and global control of multiple systems. For example, the traditional model of global synchronization for controlling multidatabases or other distributed systems does not address the needs of large volume, real-time processing and communication in computerized manufacturing. Similarly, today's manufacturing environment is characterized by heterogeneous computer hardware and a variety of local software application systems, which also goes beyond the conventional domain of computing architecture. The metadatabase model represents a novel approach to getting control of corporate information resources in order to keep pace with the ever changing and increasingly complex enterprise environment. The research thrust is the development of a new metadatabase technology [7], which contains information about enterprise data combined with knowledge about how this data is used. The metadatabase uses this knowledge to assist in bringing about information integration.

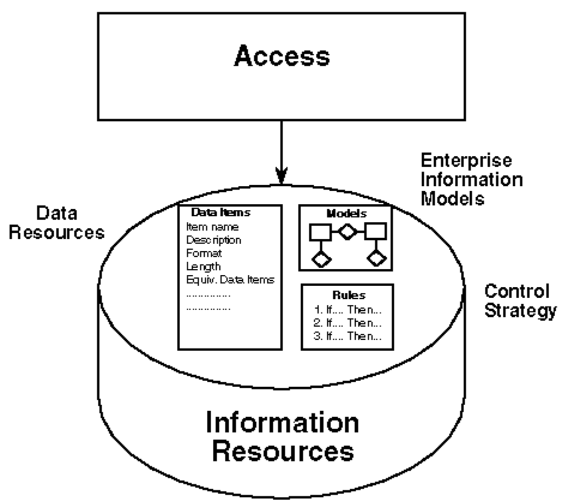

Figure 1 : Integrated Information System

An integrated information system will provide access to the organization's information resources and will use this information to assist in designing, implementing and controlling the enterprise [7]. As shown in Figure 1, this information includes an enterprise information model describing the data resources of the local subsystems and their control strategy and tactics in the form of rules. The information model also includes knowledge about the dynamics of information transfer such as what and how information is shared among local systems and under what circumstances it is used [8].

The metadatabase model is aimed at effecting such an integrated environment. The model has been implemented in a prototype system at Rensselaer's Computer-Integrated Manufacturing Program (through June, 1992) and the Adaptive Integrated Manufacturing Enterprise (AIME) Program (since June, 1992) and has been successfully demonstrated to industrial sponsors (see Acknowledgment at the end of the paper). In the Rensselaer Metadatabase system, data, along with knowledge in the form of rules, are unified and used to manage the complex and distributed environment. The union of data and knowledge is believed to be the cutting edge of research in information integration [2, 3, 14]. A major benefit of this union to (end)users is that the system makes it possible to both query and control the information in the enterprise without knowing the details of the individual subsystems or individual information flows [9].

Prior approaches to integration included use of common databases for several application systems, use of a single manufacturer's hardware, and for heterogeneous systems, use of a common data model, which controls the several local databases through a single logical manager (Figure 2a) [17]. These approaches all face severe limitations in practice. Only recently has the importance of including knowledge in the metadata started to be widely recognized. We believe that the work at Rensselaer has fostered the philosophy of combining both data and knowledge about information interactions in the metadata .

In addition, previous distributed or multiple databases either only support the mere retrieval of information or require a hierarchy of central controller to assure that updates to one database would be reflected in all other databases [11, 12, 13, 15, 16, 17]. The Rensselaer metadatabase architecture, on the other hand, allows concurrent, automatic update of all databases in a distributed environment through shells which accommodate the rules for information interaction (Figure 2b). A new correctness criterion focusing on event/usage consistency is developed and employed, as opposed to the conventional instances consistency.

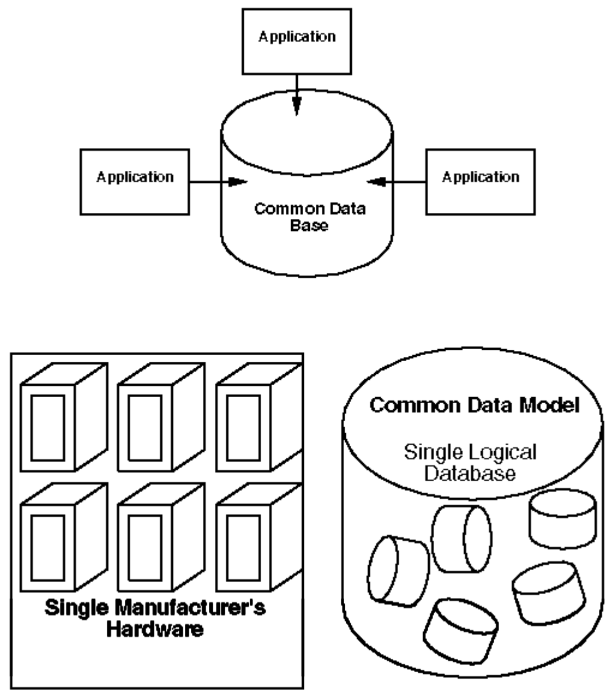

Figure 2a : Prior Approaches to Distributed Databases

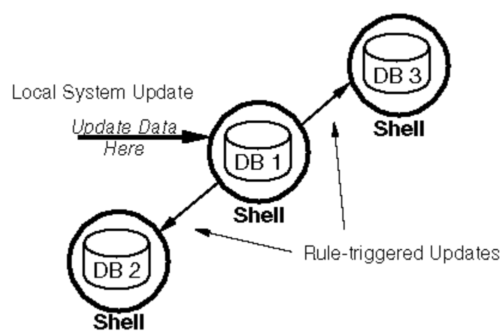

Figure 2b : Metadatabase Approach to Distributed Databases

Thus, there are two advanced distributed information system concepts which we have implemented in our metadatabase research; (1) combining information about enterprise data with knowledge about how this data is used and (2) the capability to automatically update several remote databases simultaneously when a local database is updated. Furthermore, the model has accomplished a new, much-needed goal: the ability to accommodate legacy systems, to change local systems/shells, and to add new systems, all without requiring redesign or recompilation of the system.

These results are synthesized for the first time in this paper from a user's perspective. As the objective, it provides a detailed illustration of the major concept, design, and elements of the metadatabase system, so that its general promises to intelligent manufacturing can be assessed and, ultimately, utilized by researchers and practitioners in the field. The remainder of the paper is organized as follows: 2. Metadatabase Research, providing a brief overview of the scope of the effort; 3. Metadatabase Approach, a discussion of the prototypical architecture and components of the metadatabase environment; 4. Metadatabase System Functionality, a brief discussion of the three modes of operation: metadata management, global query, and concurrent administration of multiple systems; 5. Detailed Presentation of the Three Modes, and 6. Conclusions.

2. Metadatabase Research

At Rensselaer, the information integration problem is being approached in a manner that reflects the entire scope of required technology, from information modeling methodology to integration theory (Figure 3). First, an enterprise information modeling method satisfying the unique requirements of information integration has been developed [6]. This method combines user's information SUBJECTs with their CONTEXTual knowledge and globally consolidates them into normalized ENTITIES and RELATIONSHIPS. As such, it encompasses the process-oriented functional modeling (e.g., IDEF and Data Flow Diagram) and data-oriented semantic modeling (e.g., Object-oriented and Entity-relationship) in a way that automates the mapping from high-level conceptual models to logical systems designs while connecting data with knowledge. It also provides linkages and translations with commonly used models such as CODASYL, relational, and object-oriented systems.

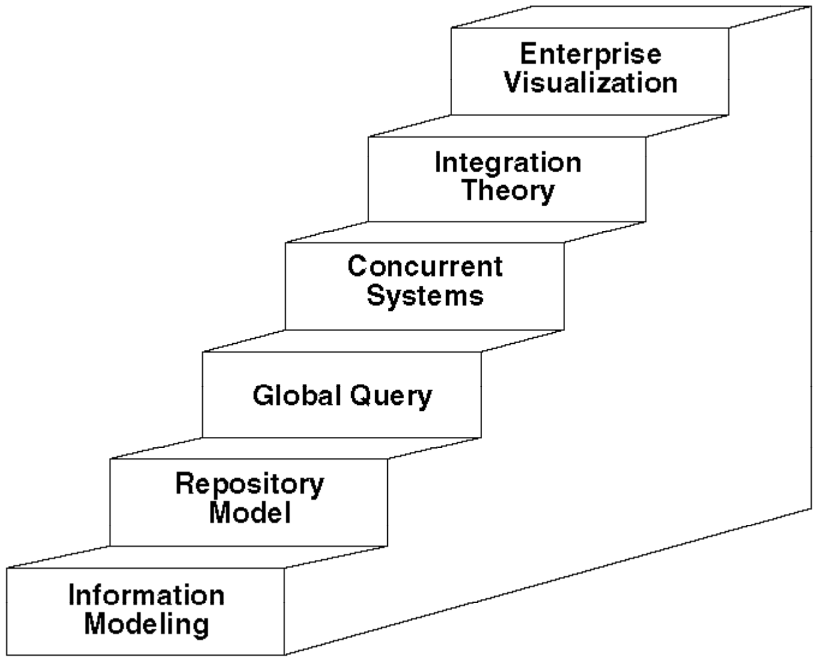

Figure 3 : A Stepwise Approach to Information Integration

A new repository model was then originated to extend prior government and industry results (IRDS [1, 4, 5]). These results tended to focus only on data dictionary and software design functions, whereas the new repository model includes substantial knowledge about the interaction of data [2, 10]. Since the structure is based on the information modeling method, it is generic and largely independent of particular applications. Through the use of this new repository model as a knowledge base, advanced global query capabilities have been provided to end users and programmers. First, the users are assisted with the on-line knowledge, thus both query formulation and processing can be performed directly in terms of information models with which they are familiar, without either requiring detailed technical knowledge of the local systems, nor relying on integrated schemata which tend to impose changes on local systems [3, 9]. Similarly, using the metadatabase for managing concurrent systems, methods have been developed to make heterogeneous application databases work together concurrently and synergistically, without necessitating a central computer to control them directly [8, 9]. The key to this is the metadatabase-driven concurrent architecture mentioned above, which is discussed further in the next section.

Finally, the research also develops a theory of integration that incorporates the new techniques in order to guide the planning, analysis and design of integrated systems [14]. The theory provides an information requirements model containing fundamental elements of production and their data classes and knowledge classes needed to achieve parallel interactions among decision processes. It is being further extended to providing microeconomical assessment of systems through the concept of transaction costs. All of these results also contribute to the user-oriented information using visualization technologies. Research on graphical user interface and beyond for all phases of information management are underway.

3. Metadatabase Approach

The metadatabase system (Figure 4) can be viewed as comprising the metadatabase itself, the Metadatabase Management System (MDBMS), the Information Base Modeling System (IBMS) and the Rule-Oriented Programming Environment (ROPE) which operationalizes the concurrent architecture. All of these are supported by the Two-Stage-Entity-Relationship (TSER [6]) model and its attendant metadata algorithms.

Metadatabase

The metadatabase is a repository of information about the structure and functions of the enterprise's multiple local application programs, their functional and information models, their databases, their interactions, and the information dynamics of the enterprise. The metadatabase is structured according to a unified metadata representation method [10]. Since this method is generic, so does the metadatabase structure. Therefore, new information models or changes to existing ones can be incorporated into the metadatabase through ordinary metadata transactions, without triggering redesign or recompilation. This is both a basic class of metadata independence [2] and a foundation to truly open systems architecture.

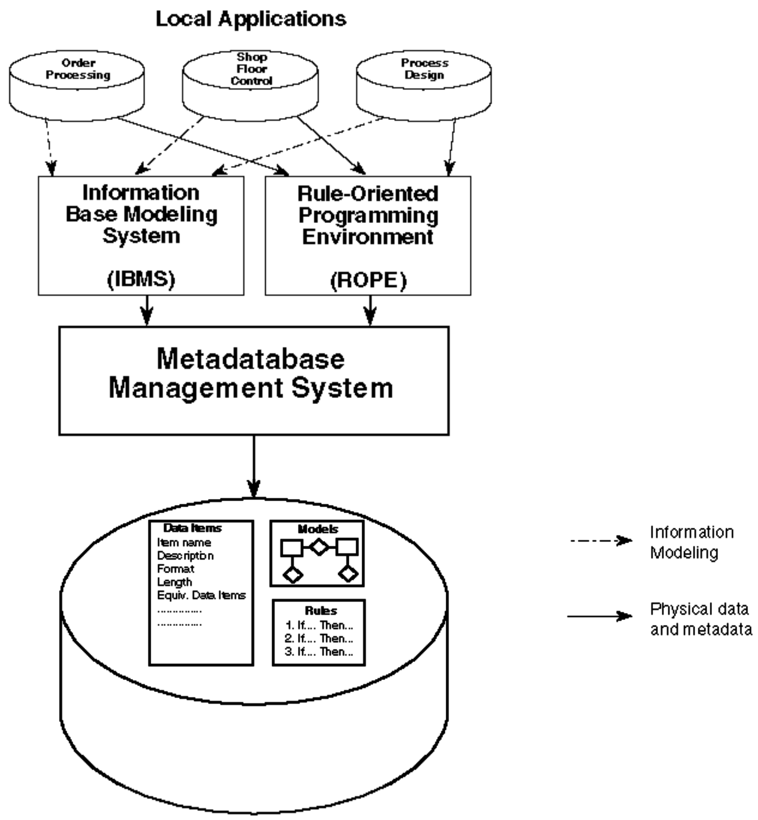

Figure 4 : The Metadatabase System

Metadatabase Management System (MDBMS)

The metadatabase management system is the user interface to the metadatabase and the processor which makes it possible to create, maintain and utilize the information in the metadatabase. MDBMS consists three basic elements: systems integrator, global query manager, and metadata manager [9]. The metadata manager, in turn, features a rulebase processor, a routine manager, and a meta-relation manager [2].

The metadatabase management system has been implemented on a microVAX platform using Digital's Rdb as the database engine for the metadatabase. A new version using the IBM AIX RS/6000 workstation and Oracle DBMS has recently been completed to provide a multi-platform and distributed metadatabase environment. To facilitate user and program interactions with the metadatabase, a shell has been developed in the C language. Currently, enterprise users interact via a menu (in the VAX version) or an X-Windows/Motif® (in the AIX version) interface, while other systems interact through a metadatabase query language application program interface (API).

Information Base Modeling System (IBMS)

The Information Base Modeling System is a computer-aided software engineering tool that assists users in designing an enterprise information system and creating a consolidated dictionary of information resources. The dictionary is the source of metadata in an automated environment, while metadata could also be provided to the metadatabase directly. Individual applications can be created using IBMS, or mapped into the global model through IBMS; but otherwise, IBMS is not required by MDBMS. To facilitate usability, however, model translators have been developed to reverse-engineer EXPRESS (PDES/STEP) products schemata into the information model, as well as to create Oracle and Rdb (both are commercial relational database management systems) schemata.

Rule-Oriented Programming Environment (ROPE)

Rule-Oriented Programming Environment [9] is a software environment which makes it possible to implement the knowledge about information interactions among the several subsystems. It is a layer between the Metadatabase Management System and the local application systems. It creates, maintains, and manages the distributed rule-based shells of the concurrent architecture according to (the changes to) the metadatabase. Furthermore, ROPE also monitors local systems behavior and effect communications among them. Thus, it essentially masks the local systems from global users and provides a logically uniform behavior of the multiple systems as a whole without intervening local operations.

The Concurrent Architecture

The basic structure of the entire metadatabase environment is characterized with a concurrent architecture depicted in Figure 5a. Each of the local subsystems and the metadatabase system has a software shell around it (Figure 5b) designed according to ROPE. The shells are responsible for monitoring events that are significant for the enterprise, for executing the rules assigned to it and for communicating with other shells. Shells can be written for each local application without disturbing the existing local system code. Figures 5a-b illustrates the flow of metadata through the shells to and from the metadatabase system and the flow of actual data among the local applications. This way, local systems interact directly with each other in parallel according to their own localized rulebase shells, while the metadatabase controls, and only controls, these local shells. Local autonomy, open systems architecture, and global adaptiveness are accomplished.

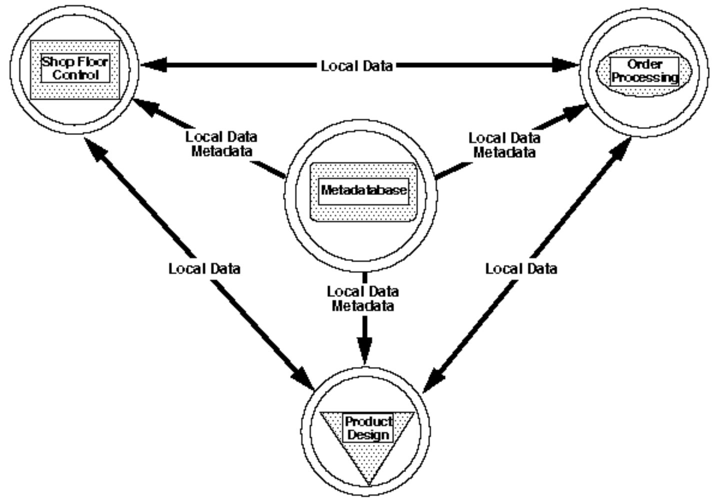

Figure 5a : Rule-Oriented Programming Environment (ROPE)

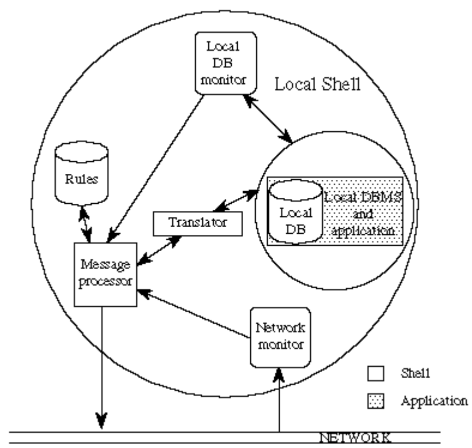

Figure 5b : ROPE Shell Architecture

The Metadatabase-Integrated Enterprise

In large-scale systems, the metadatabase system can be distributed as illustrated in Figure 6, where several of the application programs are represented in a mini-metadatabase. This mini-metadatabase then can be represented in the main metadatabase system such that the mini-metadatabase becomes the gateway to the application programs it represents. The total model shown in Figure 6 signifies the vision of the metadatabase approach.

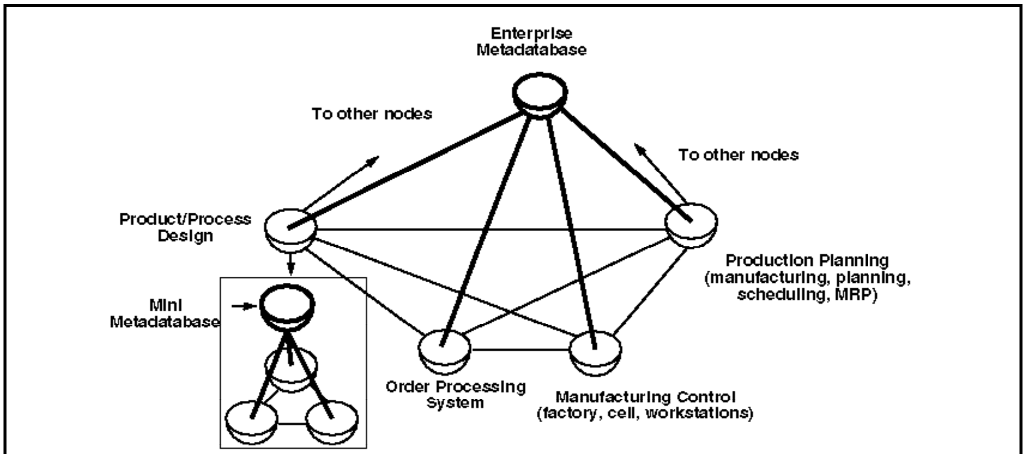

Figure 6 : Enterprise Integrated Through Distributed Metadatabase System

4. Metadatabase System Functionality

The following discussion is intended to assist in developing an understanding of the core functionality of the metadatabase system. For the purpose of this discussion, the Rensselaer CIM facility consists of three application systems; the Process Planning System, the Shop Floor Control System and the Order Processing System (Figure 5a). This state-of-the-art functionality is defined in three modes: passive, semi-active and active modes.

In the passive mode, the metadatabase system acts as a repository of information about the information in each of the several application systems. The metadatabase can be perused to get information about the location and design of data resources in the enterprise, about commonly shared data items, about the functional and information models of the applications or about the control knowledge. This information can be used, for instance, by a systems analyst who is developing a new application to be integrated into the enterprise. The information flow in Figures 5a-b shows that the individual application programs communicate with their own databases, and the metadatabase management system communicates with the metadata in the metadatabase. In the passive mode, the metadatabase and the individual application programs do not have to be connected.

In the semi-active and active modes, the metadatabase management system can interact with the applications to provide global query and systems integration functionality.

Semi-active mode functionality is implemented with a global query system that can obtain data from the application systems through the information stored in the metadatabase. The metadatabase management system can generate queries to the individual databases and join the responses when the required information is in several different application databases. The global query system has the capability of on-line assistance for query formulation.

In the active mode the metadatabase management system communicates with the metadatabase and with the individual applications in the enterprise through the shells, which are represented as circles around the applications and the metadatabase system. The shells are built on top of the individual applications so that the applications themselves do not have to be modified to operate with the metadatabase system. A shell is also built around the metadatabase system to coordinate with the shells around the applications. In the active mode, when a local update is made, the rules incorporated in the shells control the updating of other local databases either through the local data and metadata commands or through the shell-to-shell local data commands.

5. Detailed Presentation of the Three Modes

5.1. Passive Mode

We now take a closer look at the three operational modes of the metadatabase system. Several screens built into the prototype will be used to illustrate the scope and functionality of the different modes.

Figure 7 provides an overview of the system, in which we can see part of the metadatabase management system menu tree. The main menu corresponds to the five elements at the top level. MANAGE THE METADATABASE and PERFORM METADATA QUERY are used in the passive mode. MODEL LOCAL SYSTEMS is used to input information into the system. ACCESS LOCAL SYSTEMS is the Global Query System and MANAGE SYSTEM INTEGRATOR provides the facilities for Active Mode Operations in coordination with the Rule Oriented Programming Environment.

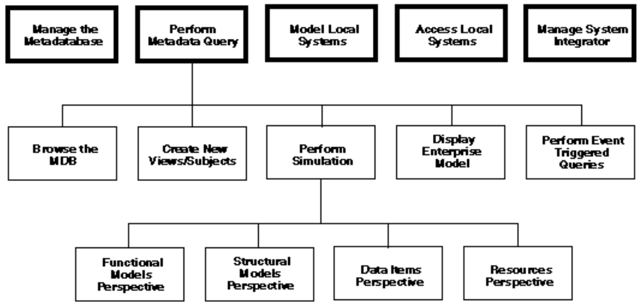

Figure 7 : Partial MDBSS Menu Tree

We will first examine the passive mode functionality (see [2] for technical details of the model, methods, and techniques developed for this mode of operation). The example shows a typical path taken by an analyst who is about to modify the Shop Floor Control Application or who is planning to develop a new application. In either case, the analyst needs to know what information resources and requirements are currently available that can be or have to be consolidated into the new effort.

To do this we will select PERFORM METADATA QUERY and choose PERFORM SIMULATION from the next level, then DATA ITEMS PERSPECTIVE (Figure 7).

The screen presented in Figure 8 displays individual data items represented in the metadatabase and allows the user to select the data items to be studied. Item #55 in the list, PART_ID from application Shop Floor Control, is selected.

Some detailed information about the data item PART_ID is given in Figure 9. In this screen, for instance, its description and format are shown.

****** LIST OF THE ITEMS IN THE METADATABASE ****** *

NO.	ITEM NAME	DESCRIPTION
45	NUM_SCRAPPED	NUMBER OF SCRAPPED WORK ORDERS
46	OD_STATUS	CUSTOMER ORDER STATUS
47	OI_STATUS	ORDER ITEM STATUS
48	OPDESC	OPERATION DESCRIPTION
49	OPID	OPERATION IDENTIFIER
50	ORDER_ID	CUSTOMER ORDER IDENTIFICATION NUMBER
51	PARTDESC	PART DESCRIPTION
52	PARTID	PART IDENTIFICATION IN PROCESS_PLAN SYSTEM
53	PARTREV	PART REVISION
54	PART_ID	PART IDENTIFICATION IN ORDER ENTRY SYSTEM
55	PART_ID	PART IDENTIFICATION IN SHOP FLOOR SYSTEM
56	PART_ID_ASSEM	ASSEMBLY PART IDENTIFIER
57	PART_ID_COMP	COMPONENT PART IDENTIFIER
58	PLANDATE	PLAN CREATION DATE
59	PLANNER	PLANNER NAME

<PRESS RETURN KEY TO CONTINUE>

*****PLEASE ENTER ITEM NAME/NUMBER, (OR
-- TYPE (L) TO LIST ALL THE ITEMS
KNOWN TO THE METADATABASE.
-- TYPE (Q) TO QUIT THIS QUERY) : 55

Figure 8 : Data Item Selection

****** ITEM DEFINITION METADATA ******

- Item Code : ITEM_64
- Item Name : PART_ID
- Description: PART IDENTIFICATION IN SHOP_FLOOR SYSTEM
- Format : CHARACTER

<PRESS RETURN KEY TO CONTINUE>

Figure 9 : Data Item Metadata

Figure 10 illustrates other information resources in the enterprise that are related to “part”. Note that the same object “part” has three different names and two implementation formats. They all are recognized by the metadatabase and presented to the user as three equivalent forms.

The information resources displayed here include rules, files, and database tables. This list of resources is derived from the knowledge about the enterprise model included in the metadatabase. The analyst will use this information to review the possible ways of incorporating “part” into the new design, or assess the impact of changing the data item PART_ID.

```
***** Any Changes to this Item may affect :

    – Application(s) :
       -- PROCESS_PLAN
       -- ORDER_ENTRY
       -- SHOP_FLOOR

    – Database Tables :
       -- PARTREV          -- PART
       -- BILL_MAT         -- ORDER_ITEM
       -- WORK_ORDER       -- PARTS_AVAIL

    – Rules :
       -- SFCRULE_11       -- SFCRULE_16
       -- SFCRULE_17       -- SFCRULE_12
       -- SFCRULE_13       -- SFCRULE_3

    – Data Files :
       -- PARTS_AVAIL
       -- WORK_ORDER

    – Equivalent Data Items :
       Data-Item Name      Format           Application System
       --------------      ----------       ------------------
       PART_ID             CHARACTER(5)      SHOP_FLOOR
       PART_ID             CHARACTER(5)      ORDER_ENTRY
       PART_ID_COMP        CHARACTER(5)      SHOP_FLOOR
       PART_ID_ASSEM       CHARACTER(5)      SHOP_FLOOR
       PARTID              CHARACTER(10)     PROCESS_PLAN

                  <PRESS RETURN KEY TO CONTINUE>
```

Figure 10 : Metadata Related to a Data Item

5.2. Semi-Active Mode

Recall that in the active and semi-active modes the metadatabase management system communicates with the individual applications in the enterprise (Figure 5a). The interface circles around the individual applications are called shells and are built on top of the applications. This is so that the applications themselves do not have to be modified to operate with the metadatabase system. A shell is also built around the metadatabase system to manage the shells around the applications.

In the semi-active mode, the interaction between the metadatabase management system and the local systems consists of queries for local data and responses to the queries. The semi-active mode has been implemented with a Global Query System that can obtain data from the application systems by using information stored in the metadatabase. The metadatabase management system can generate queries to the individual databases and join the responses when the required information comes from several different application databases. The Global

Query System has the capability of on-line assistance for query formulation.

To illustrate the semi-active mode operation (see [3] for technical details of the model, methods, and techniques developed for this mode of operation), we look at how the Global Query System deals with a query which requires the retrieving of data from three distributed application systems. The sample query is:

> "Find the Customer Order, PartID, Part Description, Quantity and Quantity Completed for Jane Doe's Customer Order which has a desired date of 10/25/91."

While it is possible to write out this query with an appropriate syntax, the Global Query user interface can be used to specify the data items required for the query and thus have the appropriate queries generated automatically.

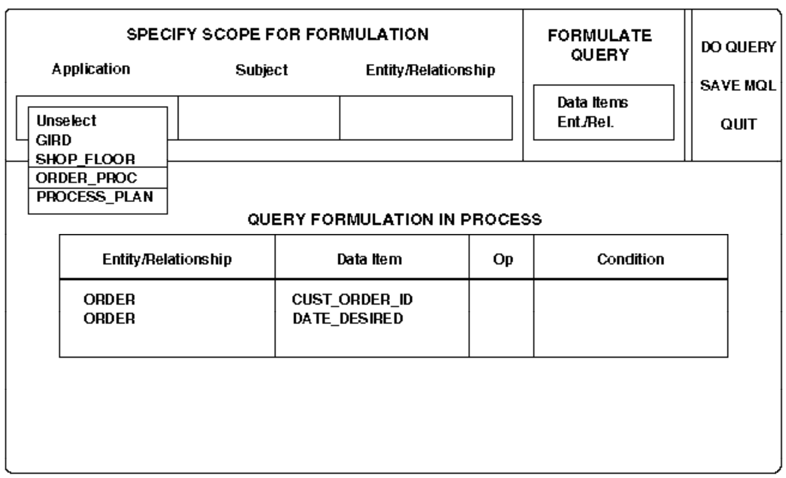

Figure 11 : Global Query System Interface

The user interface provides pop-up menus whose items are generated dynamically from the contents of the metadatabase (Figure 11). The user needs only to choose the appropriate data items in order to formulate the query. Prior to the formulation, and throughout the formulation, the scope of data items from which to choose can be narrowed as desired. Notice that the initial pop-up menu shown here allows for selecting a particular application, say Order Processing. The scope may also be set, using a pull-down menu, to a particular subject within the application, and similarly an entity or relationship within the subject may be set.

The data items that we select will appear on the bottom half of the screen. These data items may later be deleted or may have conditions placed on them.

Figure 12 illustrates a path through the data items that could be followed to generate the sample query. This figure does not depict the user interface but rather shows how related tables are traversed to identify the various data items.

We choose first the table ORDER, and select the two data items that have been darkened in the diagram. CUSTOMER_ORDER_ID and DATE_DESIRED are selected since they are both data items needed to answer the query or needed for a condition in the query. Notice that DATE_DESIRED with a value of "10/25/91" is a condition in the query. When we have selected all data items that are needed from a table, a pop-up menu allows us to select a related table that could provide further data items that we need.

For this query, we choose CUSTOMER, which is related to ORDER due to the common field CUSTOMER_ID. We select the data item CUSTOMER_NAME since it is a condition in the query that customer name be "Jane Doe". We then return back to the related table ORDER to find more related tables.

We choose the related table WORK ORDER, which has moved us from the Order Processing system to the Shop Floor Control system. The two tables are related by virtue of the field ORDER_ID, which is equivalent to the field CUSTOMER_ORDER_ID. In this database, the Order ID in Shop Floor does not actually have the same value as the Customer Order ID in Order Processing, although there is a correspondence between them, so that when tables from these two applications are joined later, the Global Query System will have to use conversion rules from the Metadatabase to do the join. We select the data items PART_ID, WORK ORDER QUANTITY (WO_QUAN), and NUMBER_COMPLETED.

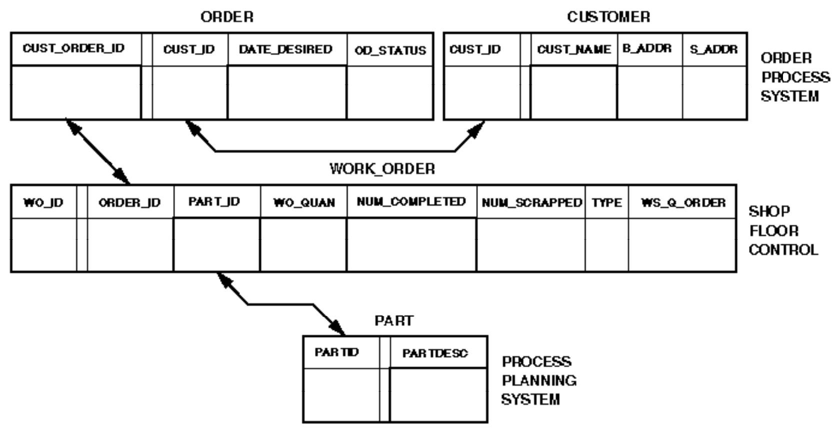

Figure 12 : Query Formulation

Again moving to a related table, we choose PART, which is in the Process Planning system. This is related due to the equivalent field PARTID. We select the data item PART DESCRIPTION.

Now we have specified all data items for our query. We move to the bottom half of our user interface and attach conditions on CUSTOMER_NAME and DATE_DESIRED. Since this terminates our query, we indicate so by selecting the DO QUERY button on the upper right corner of the screen (Figure 11). Then, after waiting for requests over the network to be satisfied and then joined together, we have the answer to our query (Figure 13).

CUST_NAME	DATE_DESIRED	CUST_ORDER_ID	PART_ID	WO_QUAN	PART_DESC	NUM_COMPLETED
Jane Doe	10/25/91	18790_25	PZ57	5	Burr Puzzle	3
Jane Doe	10/25/91	18790_25	B	1	Burr Piece	0

Figure 13 : Query Answer

5.3. Active Mode

The Active Mode is the level of Metadatabase functionality where systems integration takes place. The approach is unique in that knowledge is incorporated in the form of rules to control the systems' information interactions. The following are details and examples of how the metadatabase system achieves this.

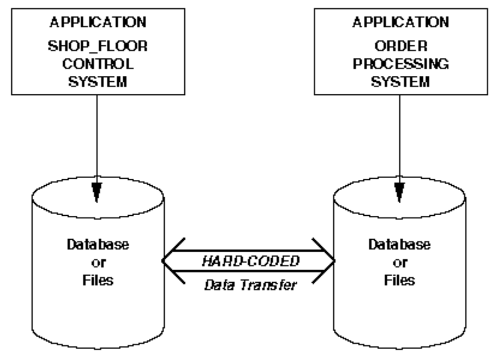

Figure 14a : Traditional Approach to System Integration

Many integration efforts focus on integrating systems at the data-level; that is, determining what pieces of data are used at what systems and writing the code to move shared data between applications. If the task is to transfer information between a Shop Floor Control System and an Order Processing System, many programmers would choose to obtain the requirements from each and hardcode the data extraction and upload between these systems (Figure 14a).

The metadatabase approach incorporates the control logic through a model based methodology. Instead of hard-coding the data links between systems, knowledge of information interaction is derived from the model, and rules are generated for moving information (Figure 14b). This rule based architecture is adaptive since a change in the model can automatically change the rules.

The active mode component of the metadatabase architecture includes facilities to: (1) generate data integration rules from models, (2) model operational rules from users, (3) implement rules automatically, and (4) manage the modification of existing rules.

The prototype system will demonstrate these functions with examples. First we will generate data integration rules from models. Beginning with the Metadatabase Management System main menu (Figure 7), we select MANAGE SYSTEM INTEGRATOR for the management of the global control rules.

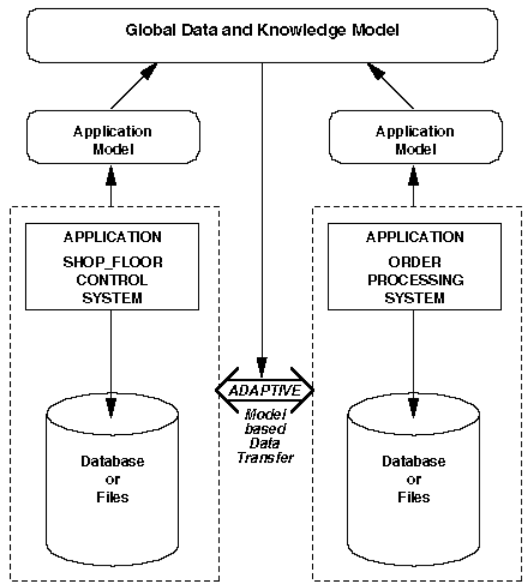

Figure 14b : Metadatabase Approach to System Integration

In order to view the global control rules, a local application system, Shop Floor Control System is selected.

We then get to the GLOBAL DATA MANAGEMENT MENU FOR APPLICATION : SHOP_FLOOR (Figure 15). The active-mode menu shows three groups of selections. The first set, DATA MANAGEMENT REPORTS provides a description of the data management rules. The second set, DATA MANAGEMENT RULE PROCESSING FACILITY presents the data management rules in a rule format appropriate to local information system managers. The third set, PROCESS GLOBAL RULES distributes the global data management rules to the local systems.

The first two categories of selections are further broken down into three classes of data management rules. These three classes of rules represent : (1) common data items, which are globally shared data elements of the same name and type, (2) equivalent data items, which are globally shared data elements that have different names or types, but that have the same meaning, and (3) integrity constraints, which are direct relationships between pieces of data among systems. Classifying them in this manner allows for convenient management of these rules. In our tour of the active mode functions, only one representative example will be viewed.

```
**********************************************************
  GLOBAL DATA MANAGEMENT MENU FOR APPLICATION : SHOP_FLOOR
**********************************************************
DATA MANAGEMENT REPORTS:
  1. View Common Data Items
  2. View Equivalent Data Items
  3. View Integrity Constraints
DATA MANAGEMENT RULE PROCESSING FACILITY:
  4. Generate Rules for Common Data Items
  5. Generate Rules for Equivalent Data Items
  6. Generate Rules for Integrity Constraints
PROCESS GLOBAL RULES:
  7. Implement Rules to Local System Shells

  P. Exit to Previous Menu

 **** Please enter your selection:
```

Figure 15 : Active Mode Facility Main Menu

To show the generation of a Data Integration Rule, we will go into the Active Mode Main Menu and pick GENERATE RULES FOR EQUIVALENT DATA ITEMS (Figure 15).

Shown in Figure 16 is the rule representation of equivalent data ORDER_ID and CUSTOMER_ORDER_ID involving Shop Floor Control and Order Processing System respectively. This rule is derived from the knowledge embedded in the model. This rule will be implemented locally at the Shop Floor site. The rule itself states that if the value of ORDER_ID in Shop Floor is updated, then the value of CUSTOMER_ORDER_ID in the Order Processing System must also be updated. The local system shells will be able to process these equivalent data rules and propagate the generated updates to distributed systems. This demonstration illustrates the generation of data management rules that integrate systems. These rules were derived from a model of how the systems interact with each other.

The systems integration approach also accommodates user defined operational rules, as is described in the second component of the Active Metadatabase: MODEL OPERATIONAL RULES FROM USERS.

This component is accessed in the Metadatabase Management System Main Menu by selecting MODEL APPLICATION SYSTEMS (Figure 7).

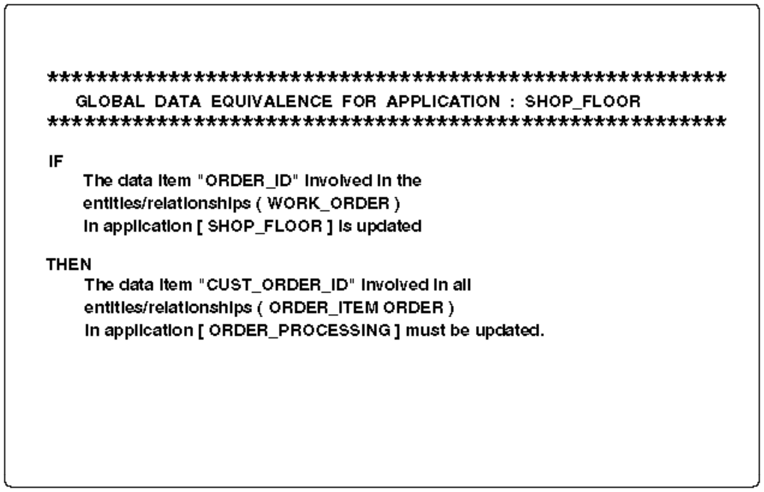

Figure 16 : Global Data Equivalence Rule

By loading part of the CIM model as displayed on the left side of the screen, rules can be entered in the INPUT KNOWLEDGE WINDOW (Figure 17). As an example, the rule entered will update the status of all orders in the Order Processing System, for all the completed work orders in the Shop Floor Control System at 2:00pm. The rule is defined by giving the rule a name, description, and type, and by typing the rule in the form of a "if-then" statement.

This is the second class of rules that the metadatabase architecture supports. These two classes of rules (model generated and user defined) are dynamically distributed and implemented at each location of the application systems. The distribution of the rules to the local systems is under the control of the Active Mode Facility.

To implement these rules, we choose IMPLEMENT RULES TO LOCAL SYSTEM SHELLS from the Active Mode Facility Main Menu (Figure 15). This will extract the data management rules for each local application system and send them over a network to the local operating shells from which they will be automatically put into effect.

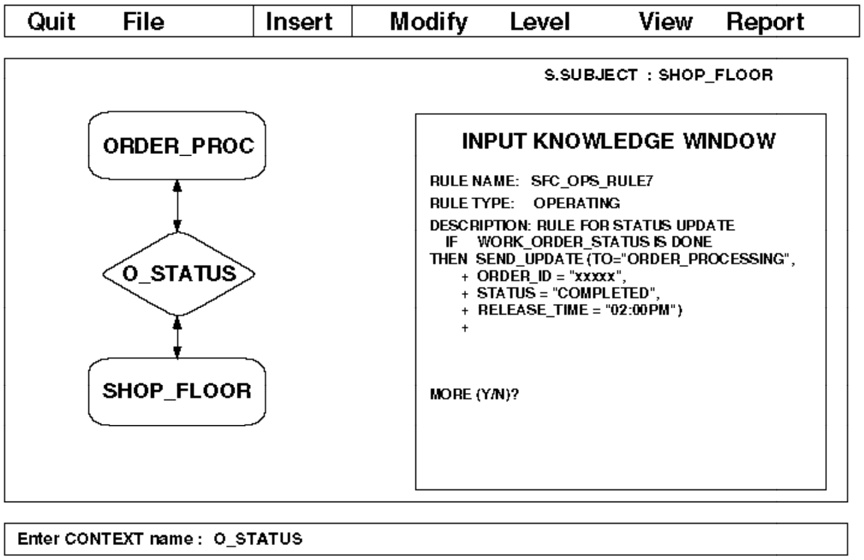

Figure 17 : Knowledge Input Window

As shown in Figure 5a, the rules are managed centrally at the Metadatabase level and are propagated to local system shells (metadata), shown as the shaded ring around each application system. At this point, each application shell will monitor the data elements based upon these data integration rules for changes and initiate appropriate updates to other application systems (Figure 5b).

It is important to note that the rules we refer to are responsible for linking systems together, as in our user defined rule for sending order status from Shop Floor to Order Processing System at a predetermined time. In Figure 18, we see that at 2:00pm, the information flow from Shop Floor to Order Processing contains needed information on order status. Furthermore, no centralized data manager is involved in this integration strategy, which leads us to be able to have many such interactions operating simultaneously.

Lastly, the active mode provides facilities to manage these rules. The management function includes the ability to manage the modification of existing rules in the metadatabase, and dynamically propagate them for update in the local system shells.

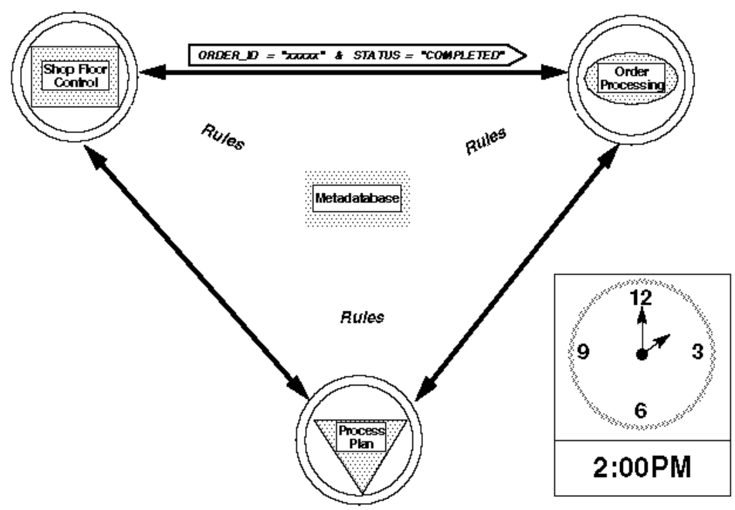

Figure 18 : Operational Rule Execution

Whenever a rule is edited through the Rule Management Facility, the facility will automatically send this revised rule to all the affected local systems and update the local rulebase shells.

6. Conclusion

Recall that the metadatabase includes information about both data and knowledge, and this unique property enables an analyst to perform a simulation and other advanced analysis and modeling tasks, as in the above examples.

The use of metadata, as managed in the passive mode, enables the semi-active and active modes of operation. More specifically, it provides the active mode with the information to perform data updates without central control.

The semi-active mode illustrates one of the key Metadatabase concepts: the combination of data and of knowledge in the form of rules. Data models in the Metadatabase are referenced in generating the global queries, and rules are employed as necessary for the joining of data from tables in different applications.

The active mode relates the two basic metadatabase concepts: the combining of data and knowledge for information exchanges in a CIM system, and the automation of updates in a distributed system without employing central database management. Both of these concepts are unique to the Metadatabase system and have been illustrated through examples from our demonstration system.

Finally, the complete model provides an important capability -- i.e., open systems architecture -- through the metadatabase residing at the global level. The metadatabase incorporates old, new, or changed local system models into its generic and integrated structure, and then implements, updates and manages the distributed shells accordingly. This capability is a new, fundamental breakthrough in the field.

Further research and development on the concurrent administration of multiple systems and the ROPE paradigm are currently underway at Rensselaer. Building on the metadatabase results, we will also investigate into the interoperability problem of multiple metadatabase/repository-based information systems (e.g., the connection of enterprise information systems that are integrated through IBM's Repository environment with systems using Digital's environments).

Acknowledgment

This work is supported in part by NSF grant DDM-9015277 and by Alcoa, Digital Equipment Corporation, GE, GM, and IBM through the Computer-Integrated Manufacturing Program and the Adaptive Integrated Manufacturing Enterprises Program at Rensselaer Polytechnic Institute.

References

[1] American National Standards Institute. (Draft proposed) American National Standard Information Resource Dictionary System: Part 1-Core Standard. ANSI X3H4, American National Standards Institute, New York. 1985.

[2] M. Bouziane, “Metadata Modeling and Management,” unpublished Ph.D. Thesis - Computer Science, Rensselaer Polytechnic Institute, June 1991.

[3] W. Cheung, “The Model-Assisted Global Query System,” unpublished Ph.D. Thesis - Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, November 1991.

[4] D. Dolk and R. Kirsch, II. "A Relational Information Resource Dictionary System," *Com-*

munications of the ACM, Vol. 30, No. 1, January 1987, pp. 48-61.

[5] ______ and ______ , *A Technical Overview of the Information Resource Dictionary System (Second Edition)*, NBS Special Publication NBSIR 88-3700, National Bureau of Standards, Gaithersburg, MD, January 1988.

[6] C. Hsu, "Structured Database Systems Analysis and Design Through Entity-Relationship Approach," *Proceedings 4th International Conference on Entity-Relationship Approach*, IEEE Computer Society, 1985, pp. 56-63.

[7] ______ and C. Skevington, "Integration of Data and Knowledge in Manufacturing Enterprises: A Conceptual Framework," *Journal of Manufacturing Systems*, Vol. 6, No. 4, 1987 pp. 277-285.

[8] ______ and L. Rattner, "Information Modeling for Computerized Manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 4, 1990, pp. 758-776.

[9] ______ , G. Babin, M. Bouziane, W. Cheung, L. Rattner, and L. Yee, "Metadatabase Modeling for Enterprise Information Integration ," *Journal of Systems Integration*, Vol. 2, No. 1, 1992, pp. 5-39.

[10] ______ , M. Bouziane, L. Rattner, L. Yee, "Information Resources Management in Heterogeneous, Distributed Environments: A Metadatabase Approach," *IEEE Transactions on Software Engineering*, June, 1991, pp. 604-625.

[11] V. Krishnamurty, S. Y. W. Su, H. Lam, M. Mitchell and E. Barkmeyer, “IMDAS, An Integrated Manufacturing Data Administration System,” *Data and Knowledge Engineering*, Vol.3, No.2, 1988, pp. 109-131.

[12] T. Landers and R. L. Rosenberg, “An Overview of MULTIBASE,” *Second Symposium on Distributed Database*, North Holland, New York, NY, 1982, pp. 311-336.

[13] D. McLeod, “Interbase: An Approach to Controlled Sharing Among Autonomous, Heterogeneous Database Systems,” *Data Engineering Bulletin*, Fall 1990, pp. 4-9.

[14] L. Rattner, “Information Requirements for Integrated Manufacturing Planning and Control: A Theoretical Model,” unpublished PhD Thesis - Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, November 1990.

[15] A. Sheth and J. Larson, “Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases,” *ACM Computing Surveys*, Vol. 22, No. 3, 1990, pp. 203-236.

[16] J. M. Smith, P. A. Bernstein, U. Dayal, N. Goodman, T. Landers, K. W. T. Lin and E. Wong, “MULTIBASE — Integrating Heterogeneous Distributed Data Systems,” *AAFIPS Proceedings*, Vol.50, 1981, pp.487-499.

[17] U.S. Air Force, *Integrated Information Support System (IISS) Report*, Integrated

Computer Aided Manufacturing (ICAM), Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, OH, February 1983.

[18] R. Wang and S. Madnick, "Facilitating Connectivity in Composite Information Systems," *ACM Data Base*, Vol. 20, No. 3, Fall 1989, pp. 38-46.