The Metadatabase for Manufacturing Systems Integration

Cheng Hsu[†1], Gilbert Babin[2], M'hamed Bouziane[3], Waiman Cheung[2],
Laurie Rattner[4], Lester Yee[2]

Department of Decision Sciences and Engineering Systems
Rensselaer Polytechnic Institute
Troy, New York, 12180-3590
(518) 276-6847 Bitnet: USERCHBY@RPITSMTS

[1] Associate Professor of Decision Sciences and Engineering System, Rensselaer Polytechnic Institute, Troy, New York
[2] Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, New York
[3] Digital Equipment Corporation, Nashua, New Hampshire
[4] Assistant Professor of Management Information Systems, University of New Mexico, Albuquerque, New Mexico

Abstract

The Metadatabase project is a multi-year research effort at Rensselaer Polytechnic Institute. Sponsored by industry (ALCOA, Digital Equipment Corporation, GE, General Motors, IBM and others) through Rensselaer's Computer Integrated Manufacturing Program, this project seeks to develop novel concepts, methods and techniques for achieving information integration across major functional systems pertaining to computerized manufacturing enterprises. Thus, the metadatabase model encompasses the generic tasks of heterogeneous, distributed and autonomous databases administration, but also includes information resources management and integration of concurrent (functional) systems. The model entails (1) an integrated data and knowledge modeling and representation method; (2) an on-line kernel (the metadatabase) for information modeling and management; (3) metadatabase assisted global query formulation and processing; (4) a concurrent architectural whereby global synergies are achieved through (distributed) metadata management rather than synchronization of (distributed) database processing; and (5) a theory of information requirements for integration. A metadatabase prototype was recently demonstrated to the industrial sponsors. The basic concept of the metadatabase model is discussed through the prototype in this paper.

1. The Concept of Metadatabase

To fully implement the vision of integrated manufacturing [see e.g. 1,3], new results in the areas of information integration and data and knowledge resources management are necessary [4,5,6,7,8]. A key element in these requirements is what we shall call enterprise metadatabase [3,5,6]. This concept, in its own right, is an extension to previous metadata systems such as the Information Resources Dictionary System (IRDS) of the National Institute of Standards and Technology (formerly NBS) [2], and the latest Repository systems developed respectively by IBM and some other corporations. However, the metadatabase also serves as the kernel of systems integration and thereby leads to a new approach to effecting synergy for heterogeneous, distributed environments.
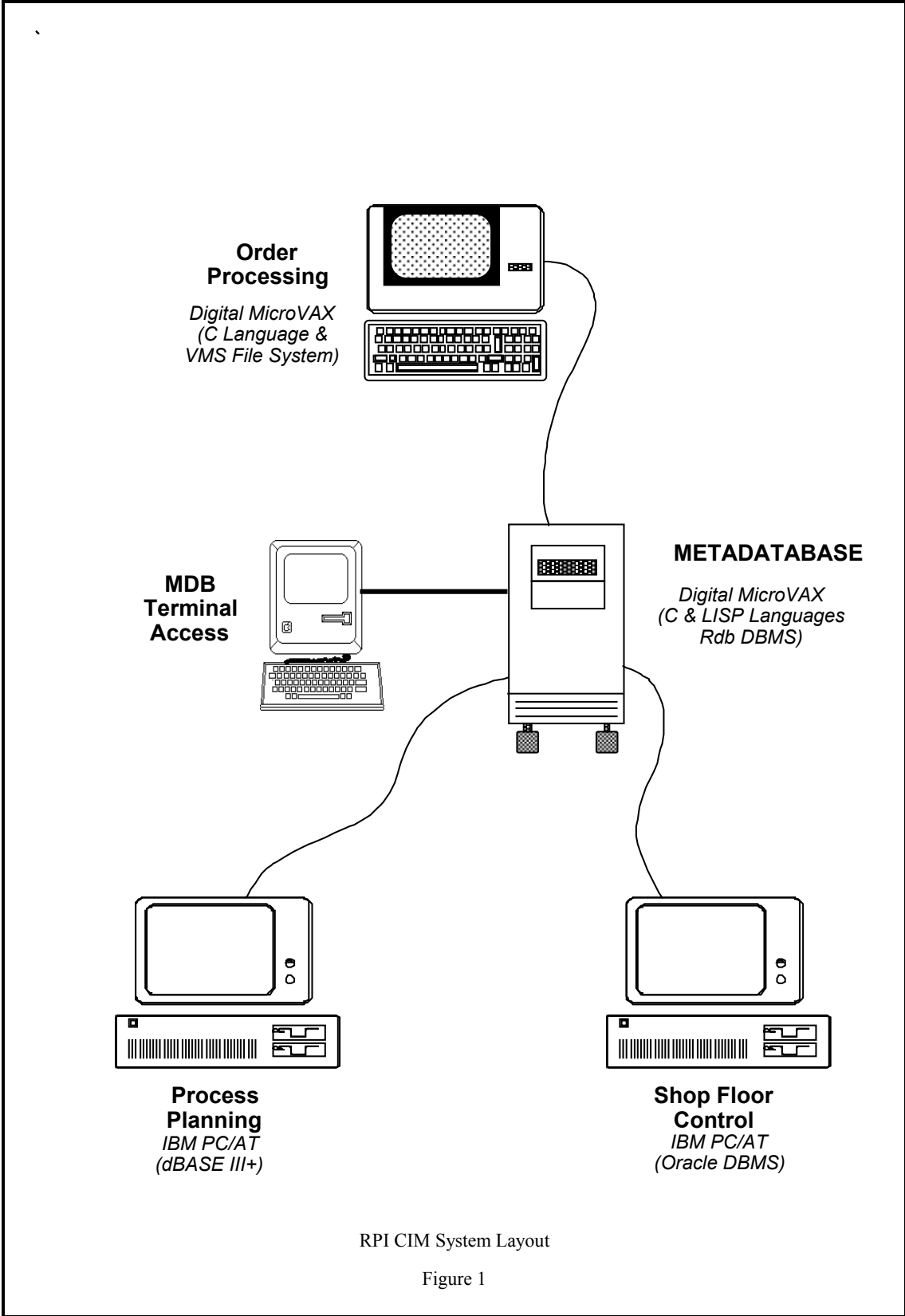
The scope of metadata in this approach is extended from representing data systems per se to including knowledge resources as well. These knowledge resources not only comprise the various computerized enterprise, but more importantly represent the functional contexts in which individual information (sub-) systems operate towards enterprise-wide synergy, or information integration (see [4,6] for detailed discussions of the structure and design of the metadatabase.)

A prototype metadatabase system for information integration is currently under development at Rensselaer. This system is implemented for the Computer Integrated Manufacturing Laboratory. The order entry, process planning and shop floor control systems are employed as depicted in Figure 1. The order entry system is implemented as a VAX/VMS file system, the process planning system developed as a dBASE III+ application on an IBM PC/AT, and the shop floor control system runs under PC/Oracle DBMS. The metadatabase is the fourth system and resides on a Digital MicroVAX running Rdb through the host language C. A heterogeneous hardware and software environment was intentionally developed to help convey how the metadatabase approach can help with information integration.

The metadatabase operates in three modes providing three levels of functionality:
1) Repository/Enterprise Information Resources Model (Passive Mode)
2) Global Query (Semi-active Mode)
3) Systems Integration (Active Mode)

In the passive mode, the metadatabase functions as an on-line logical kernel whereby users throughout the enterprise can gain access to an integrated model of the enterprise. The second level, semi-active mode, builds on the passive mode and provides model-assisted global query capabilities retrieving actual data from local systems. Finally, the third level, active mode, serves as the knowledge base supporting the control of system interactions and updates to local systems.

**Order
Processing**

*Digital MicroVAX
(C Language &
VMS File System)*

**MDB
Terminal
Access**

METADATABASE

*Digital MicroVAX
(C & LISP Languages
Rdb DBMS)*

**Process
Planning**
*IBM PC/AT
(dBASE III+)*

**Shop Floor
Control**
*IBM PC/AT
(Oracle DBMS)*

RPI CIM System Layout

Figure 1

To best illustrate how the metadatabase achieves information integration, some examples of its operation are developed below, using a scenario for this environment. Key features and functions of the metadatabase are introduced through these examples.

2. Enterprise Information Resources Model (the Passive Mode)

The passive mode metadatabase serves primarily for enterprise users, information managers and system developers as a stand-alone enterprise information resources repository. A variety of information ranging from summary views of an enterprise to data-element details is provided by the metadatabase in this mode.
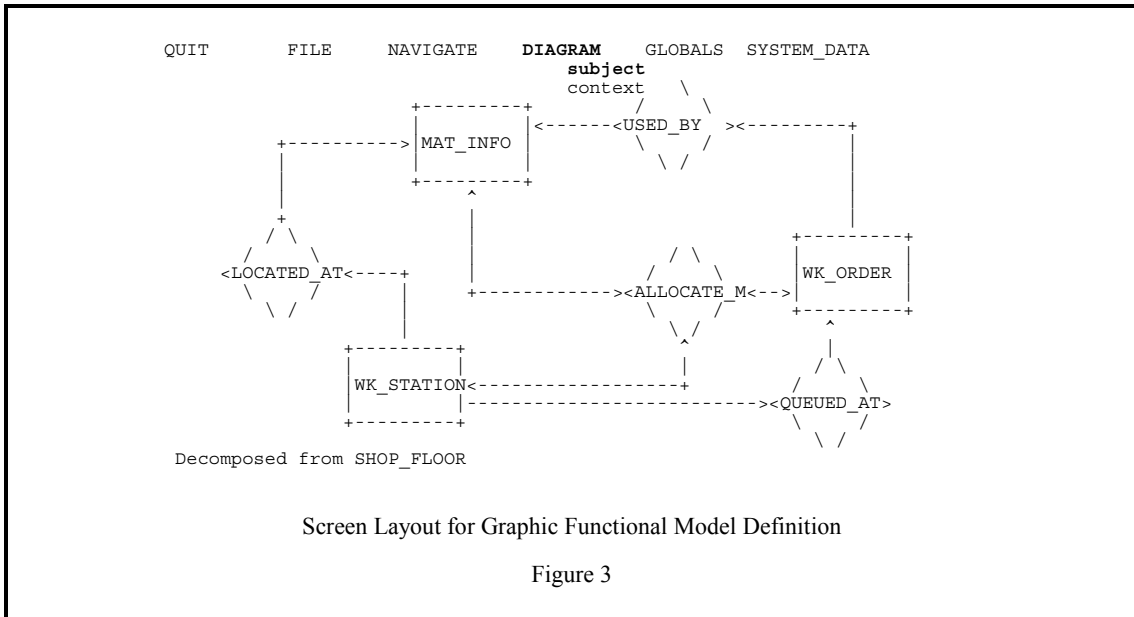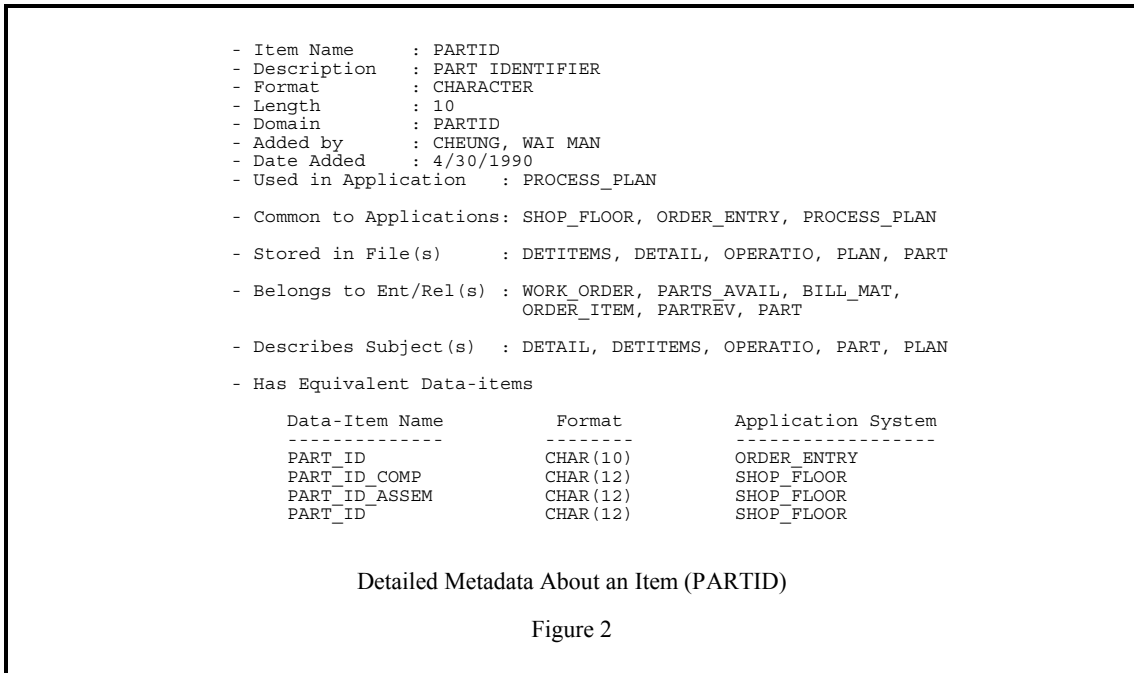
There are four unique features of the passive metadatabase. First, it contains the contextual knowledge about data resources and combines both in a unified representation for easy retrieval. Second, the metadatabase supports life-cycle representation of models ranging from planning and analysis to design and implementation. Third, the passive metadatabase allows users to obtain overviews as well as perform "What if?" types of analysis on proposed new applications or changes to existing enterprise information resources. Finally, the passive mode metadatabase supports the integration of information modeling (using current functional views in the enterprise) with metadata management itself. These features of the passive mode are based solely on the content of the metadatabase. All the information necessary for information integration can be retrieved and then presented in the appropriate format to the user.

To show the capability of the passive mode, consider a scenario in which we are developing a new application to better support management's need to track customer orders. This example illustrates how the passive metadatabase can support both information management and systems development. The following four information requests illustrate the multiple perspectives of the systems planning, analysis, and design life-cycle for developing a new application system: an order tracking system.

      (1)   Investigate what information resources we have.
      (2)   Investigate what information we have about orders and parts.
      (3)   Develop an information model for the order tracking system.
      (4)   Investigate implementation details for the order tracking system.

What follows are excerpts from an actual run of the passive mode to provide answers to the above requests. For the first request, the metadatabase presents the existing information resources that are available — classified according to the major enterprise subsystems. We have the ability to delve into many levels of detail on any of the systems included in the metadatabase. (i.e. System Level $\varnothing$ Functional Model $\varnothing$ Structural Model $\varnothing$ Data-element Level $\varnothing$ Physical Implementation Level)

The second request embeds the knowledge that the data-element "part" would be involved in the order tracking system, and, as such, the query seeks all information relating to part (e.g., "PARTID"). The metadatabase presents this information in Figure 2.

```
              - Item Name       : PARTID
              - Description     : PART IDENTIFIER
              - Format          : CHARACTER
              - Length          : 10
              - Domain          : PARTID
              - Added by        : CHEUNG, WAI MAN
              - Date Added      : 4/30/1990
              - Used in Application   : PROCESS_PLAN

              - Common to Applications: SHOP_FLOOR, ORDER_ENTRY, PROCESS_PLAN

              - Stored in File(s)     : DETITEMS, DETAIL, OPERATIO, PLAN, PART

              - Belongs to Ent/Rel(s) : WORK_ORDER, PARTS_AVAIL, BILL_MAT,
                                        ORDER_ITEM, PARTREV, PART

              - Describes Subject(s)  : DETAIL, DETITEMS, OPERATIO, PART, PLAN

              - Has Equivalent Data-items

                  Data-Item Name         Format           Application System
                  --------------         --------         ------------------
                  PART_ID                CHAR(10)         ORDER_ENTRY
                  PART_ID_COMP           CHAR(12)         SHOP_FLOOR
                  PART_ID_ASSEM          CHAR(12)         SHOP_FLOOR
                  PART_ID                CHAR(12)         SHOP_FLOOR
```

Detailed Metadata About an Item (PARTID)

Figure 2

```
       QUIT       FILE      NAVIGATE    DIAGRAM    GLOBALS  SYSTEM_DATA
                                        subject
                                        context    \
                            +---------+       /  \
                +---------->|MAT_INFO |<------<USED_BY  ><---------+
                |           |         |       \   /     \  /       |
                |           +---------+        \ /                 |
                |                ^                                 |
                +                |                   +---------+
               / \               |                   |WK_ORDER |
              /   \              |        / \         |         |
          <LOCATED_AT<----+      |       /   \        +---------+
             \   /        |   +----------><ALLOCATE_M<-->  ^
              \ /         |      \   /      \   /           |
          +---------+     |       ^          \ /          / \
          |         |     |       |           |          /   \
          |WK_STATION<----------------+        |       <QUEUED_AT>
          |         |-----------------------------><QUEUED_AT>
          +---------+                                  \  /
                                                        \ /
       Decomposed from SHOP_FLOOR
```

Screen Layout for Graphic Functional Model Definition

Figure 3

For the third request, the metadatabase presents a graphical diagram to help an analyst determine whether an existing model component can be reused in developing the model of the new order tracking system. If the new model cannot be built from components of any existing model, the

metadatabase modeling facility is also available to develop it. This new model, when completed and approved, will be integrated with the present global enterprise model.The graphical representation of a part of a model in the metadatabase is shown in Figure 3 and is generated by reverse modeling directly from the contents of the metadatabase (i.e., Figure 3 is *computed* from metadata and not a stored graphical file).

```
***** Any Changes to this Item may affect :
        -- Application(s) :
            --- SHOP_FLOOR
            --- ORDER_ENTRY
            --- PROCESS_PLAN

        -- Subjects :
            --- DETAIL
            --- DETITEMS
            --- OPERATIO
            --- PART
            --- PLAN

        -- Entity/Relationships :
            --- WORK_ORDER           --- PARTS_AVAIL
            --- BILL_MAT             --- ORDER_ITEM
            --- PARTREV              --- PART

        -- Rules :
            --- RULE_9               --- RULE_8
            --- RULE_7               --- RULE_6
            --- RULE_30              --- RULE_29
            --- RULE_28              --- RULE_27
            --- RULE_26              --- RULE_25
            --- RULE_24              --- RULE_23
            --- RULE_22              --- RULE_21
            --- RULE_20              --- RULE_19
            --- RULE_18              --- RULE_17
            --- RULE_16              --- RULE_15
            --- RULE_14              --- RULE_13
            --- RULE_12              --- RULE_11
            --- RULE_10

        -- Data Files :
            --- DETITEMS
            --- DETAIL
            --- OPERATIO
            --- PLAN
            --- PART

        -- Other Data Items :
            --- PART_ID              ===> of Application : ORDER_ENTRY
            --- PART_ID_COMP         ===> of Application : SHOP_FLOOR
            --- PART_ID_ASSEM        ===> of Application : SHOP_FLOOR
            --- PART_ID              ===> of Application : SHOP_FLOOR
```

What-If Analysis

Figure 4

The fourth request presents what model components will be affected if a data item is redefined or deleted. This is the metadatabase sensitivity analysis capability that a systems developer would utilize. Clearly, to involve "PARTID" in the new order tracking system one must consider the implications on existing uses of that data item as shown in Figure 4

From reviewing the results shown in Figure 4 one can see that "PARTID" is involved in all three applications: Shop Floor Control, Process Planning, and Order Entry. It is also involved in a number of rules and physically implemented in a number of files. This data item is logically equivalent to four other data items (with different names and types) from other systems.

These four types of query highlight some of the functions that the passive mode provides to support managers' and developers' needs for integrated information. They also establish a foundation upon which global queries can be based.

3. Global Query System (the Semi-active Mode)

Enterprise users often want to retrieve information from local subsystems without being burdened with technical modeling or implementation details. Toward that end, the semi-active mode supports end-users' need for local data irrespective of where such data are stored. This semi-active mode provides a global query system (GQS). The global query capability simplifies access to data from multiple (and even heterogeneous) local systems. Users interact with the single integrated global enterprise model instead of manually searching, reconciling, and consolidating information from multiple databases and file systems.

The GQS processes queries that involve any number of local systems without requiring the users to know much about the existence or contents of specific local systems. Three features make the global query system distinct from other distributed query systems. First, the system supports a syntax-free interface for end-users; i.e. the system helps and guides query formulation by applying information contained in the metadatabase. For example, GQS automatically uses the metadata concerning global equivalence among data items to build queries across local systems. Second, end-users are spared many tedious processing details since the global query system generates cptimized sub-queries, interacts with any number of local systems, and consolidates results automatically in the background. Third, GQS has an adaptive interface for metadata retrieval; that is, the menu items provided to users for query formulation are *computed* based on the current contents of the metadatabase and the menus automatically change whenever metadatabase contents change.

Consider this global query example:

> "Find the customer order ID, part ID, part description, and quantity completed for Gilbert Babin's order which has a desired date of 10/25/90."

This request requires that we access data from all three application systems: order entry, shop floor control and process planning. (Note: The user posing this query need not to be aware that the query traverses multiple systems.) The user engages the system through the model-assisted dialog menus to formulate the query and marks the data fields needed along with any conditional statements. (In this query example, "Date desired=10/25/90" and "Customer name=Gilbert Babin.") Once this formulation is completed, GQS decomposes the query into an optimized set of local-bound subqueries that are determined by the system using the metadatabase. Then, each subquery is sent across the network to the respective local systems to be serviced; GQS disseminates these subqueries using the native query languages of the local systems involved (in this case, dBASE III+ for Process

Planning, Oracle/SQL for Shop Floor Control, and file system access for the Order Entry System). Each application's local shell receives a request to process the subquery. Upon reception, the subquery is executed by calling the local DBMS with the file containing the generated subquery. The results are then sent back by the local shells to the GQS where they are assembled logically and presented to the user. Figure 5 shows a model-assisted query formulation session. The user does not need to know the information model; it is presented here as an illustration of the model network stored in the metadatabase. This network is used by GQS itself to guide the query formulation process.

The shaded lines show the particular path followed by the user to formulate this query. Note the relationship between the data items (listed at the bottom of the model) and the systems in which they are stored (shown at the top). The items tagged with asterisks have been requested in the query. Also observe that the rounded boxes represent SUBJECTs and the squared boxes the corresponding data structures (OEs and PRs) of the enterprise model.

Query Example: Model Traversal Tree

Figure 5
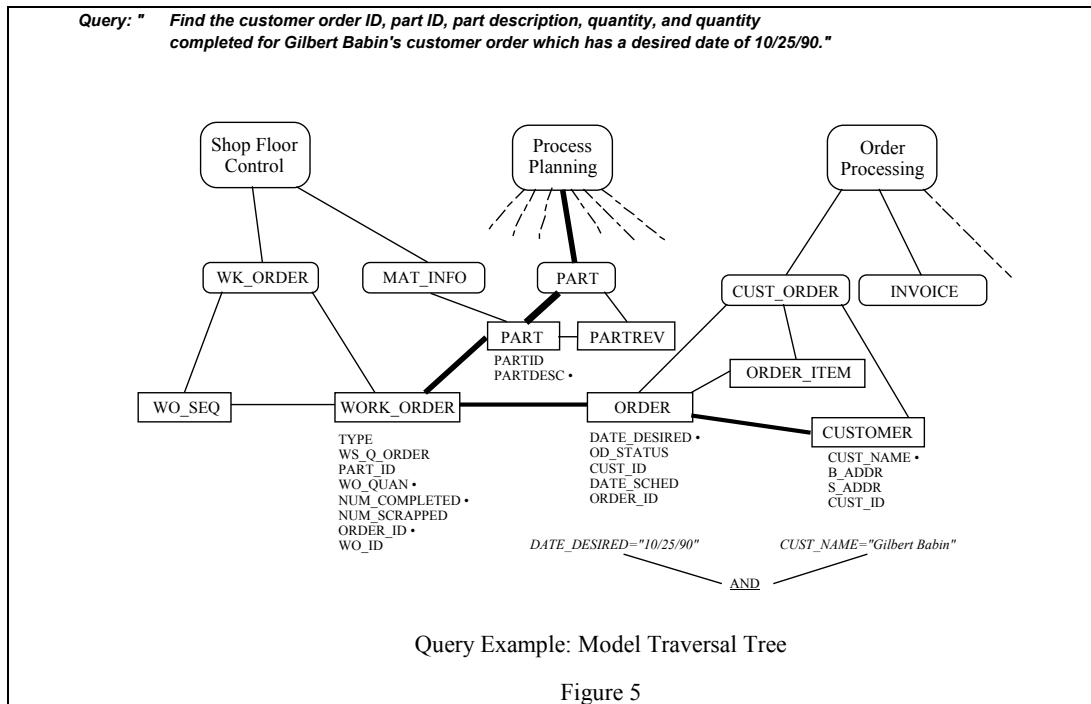
In demonstrating the passive and semi-active modes, the examples have shown how the metadatabase supports a variety of needs for integrated information. The metadatabase functionality also incorporates an active mode that manages and controls the interactions among information system beyond the previous modes. This active mode is described in the next section.

7

4. System Interactions (the Active Mode)

Maintaining control of interactions among multiple systems is quite complex since individual systems tend to evolve separately over time and may be distributed physically and logically. Global control is necessary, however, to achieve logically complete information integration for an enterprise. The metadatabase facilitates this complex and difficult task by serving as a knowledge-base for the enterprise and thereby opening up many possibilities for simplification in the design of the environment. In particular, the active mode operation of the metadatabase employs a concurrent architecture incorporating distributed rule-oriented shells for local systems.

Concurrent Systems Integrator Architecture
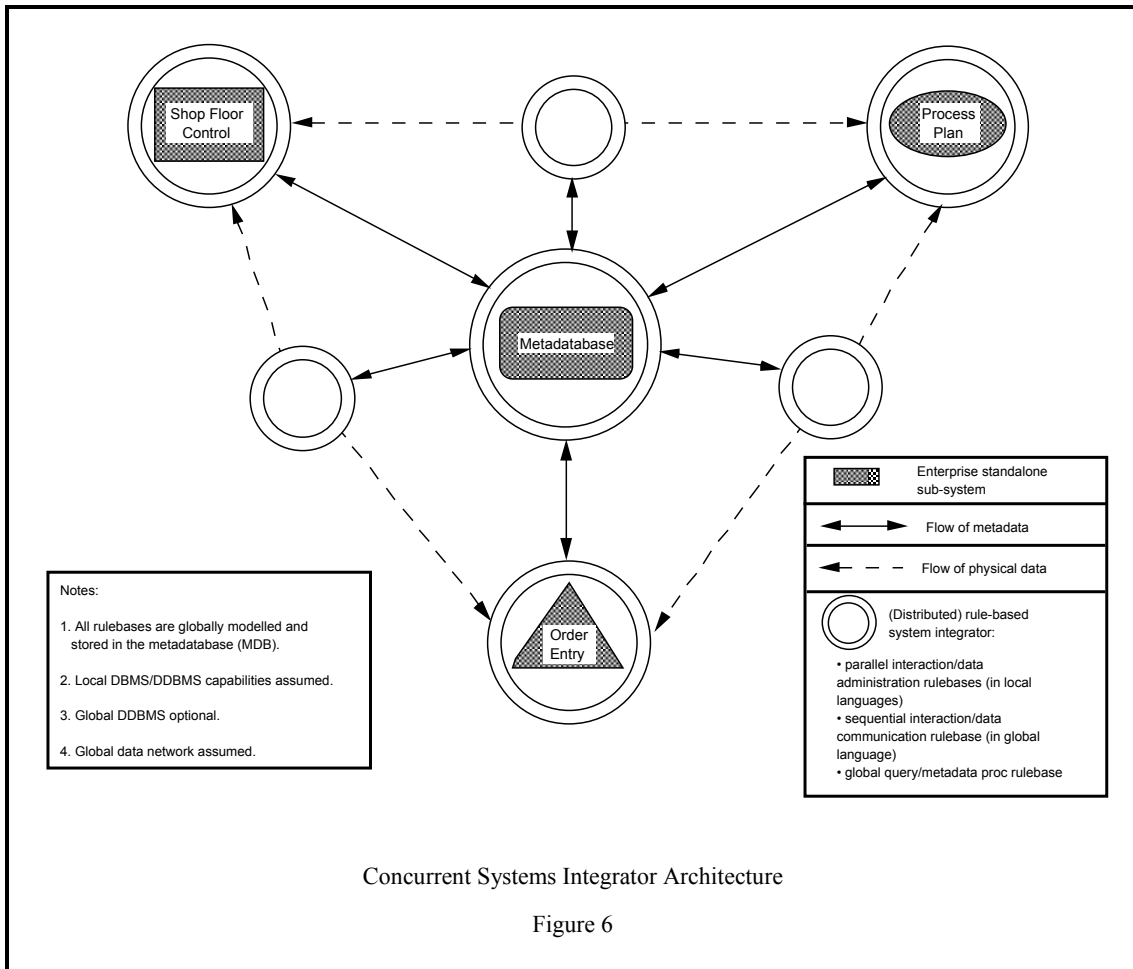
Figure 6

In this mode, the operating, control, and decision knowledge that has been modeled in the metadatabase become operational. The individual shells (Figure 6) process these rules for their respective application.

The Rule-Oriented Programming Environment (ROPE) [6] concept is implemented by using, at the present time, existing software technology. Specifically, the rules embodied in these shells are not hardcoded as would be the case with conventional techniques; instead, they are represented into a

8

separate section of the shell code according to the rulebase model of GIRD. This section is interpreted by the rest of the code for execution, and is also updated directly from outside. This way, the logic of the shells can be changed, maintained, and managed in just about any ways without necessitating the shell code to be recompiled at every change.

The following example demonstrates the significance of these concepts in achieving information integration of the stand-alone Order Entry System with the Shop Floor Control System.

When an order is entered into the Order Entry System and is committed to the order database, a trigger is activated (based upon a rule in the metadatabase). This trigger fires rules that extract the new order, convert it to the destination format, export it to the Shop Floor Control System and then execute the order. The local shell is responsible for activating the trigger; i.e., it must be able to detect that a change occured in the Order Entry System database. The knowledge for inter-system data in the metadatabase is used to determine the data elements and corresponding tables that need to be monitored for changes. This monitoring can be event-triggered or temporal-based. Once a change is detected (by comparing an old copy of the table with the current one), messages are generated and sent to the applications influenced by that change. By detecting the committing of a new order, the Order Entry System active mode shell fires a series of rules that will generate messages to update the Shop Floor Control application. Another shell (Shop Floor Control active shell) receives these messages and process them.

In summary, three levels of functionality are provided by the metadatabase for information integration. In the passive mode, users have a repository of metadata. In the semi-active mode, users have the ability for ad-hoc retrieval of information from anywhere in the enterprise through a guided, enterprise-specific global query system. Finally, in the active mode, the tasks for managing the complex interactions among systems is achieved by activating the knowledge already contained in the metadatabase. Thus, metadatabase technology is used to integrate the management and control of the enterprise information resources.

5. Future Work

The metadatabase prototype is under continual development. Research into new methods include (1) further development of the ROPE concept for implementing a concurrent architecture, (2) incorporation of the object-oriented paradigm and a modeling language for the metadatabase, and (3) reverse engineering (i.e. converting relational and other traditional schemata, semantic models, and functional or data flow models into TSER models).

References:

[1]   ESPRIT Consortium AMICE (eds.), <u>Open System Architecture for CIM</u>, Springer-Verlag, Berlin, Germany, 1989.

[2]   Goldfine, A. and P. Konig, <u>A Technical Overview of the Information Resource Dictionary System ,2nd ed.,</u> NBS Special Publication NBSIR 88-3700, National Institute of Standards and Technology, Gaithersburg, MD, January 1988.

[3]   Hsu, C. and C. Skevington, "Integration of Data and Knowledge in Manufacturing Enterprises: A Conceptual Framework," <u>J. of Manufacturing Systems</u>, Vol.6, No.4, pp.277-285, 1987.

[4]   Hsu, C. and L. Rattner, "Information Modeling for Computerized Manufacturing," <u>IEEE Transactions on Systems, Man, and Cybernetics,</u> Vol. 20, No. 4, pp. 759-776, 1990.

[5]   Hsu, C. M. Bouziane, L. Rattner and L. Yee, "Information Resources Management in Heterogeneous, Distributed Environments: A Metadatabase Approach," <u>IEEE Transactions on Software Engineering</u>, Vol. SE-17, No. 6, pp. 604-625, June 1991.

[6]   Hsu, C., G. Babin, M. Bouziane, W. Cheung, L. Rattner and L. Yee, "Metadatabase Modeling for Enterprise Information Integration," <u>J. of Systems Integration</u>, forthcoming

[7]   Krishnamurty, V., S. Y. Su, H. Lam, M. Mitchell and E. Barkmeyer, "IMDAS, An Integrated Manufacturing Data Administration System," <u>J. of Data and Knowledge Engineering</u>, Vol.3, No.2, pp. 109-131, 1988.

[8]   U.S. Air Force, <u>Integrated Information Support System (IISS) Report</u>, Integrated Computer Aided Manufacturing (ICAM), Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, OH, February 1983.