SRC TR 87-188

An Expert Systems Approach to Integrating Product Design and Manufacturing Execution Modules

by

G. Harhalakis, L. Mark, C.P. Lin, and B. Cochrane

AN EXPERT SYSTEMS APPROACH TO INTEGRATING PRODUCT DESIGN AND MANUFACTURING EXECUTION MODULES

By

G. HARHALAKIS[1], L. MARK[2], C.P. LIN[1], B. COCHRANE[2]

[1]Department of Mechanical Engineering, University of Maryland
[2]Department of Computer Science, University of Maryland

ABSTRACT. To build an interface between product design and manufacturing execution modules is the first phase to achieving Computer Integrated Manufacturing for the factory of the future. A model of the Computer Aided Design (CAD)/Manufacturing Resource Planning (MRP II) integrated system, detailing the logical interaction between the systems in the areas of part specification maintenance, product structure data maintenance, and engineering changes, has been developed. An expert system based on multi-database interoperability, which uses a formal language, called "Update Dependencies", based on an AI production system, is being developed, and has been tested against the major part of the model design specification.

INTRODUCTION. Since 1980, a large number of computerized manufacturing systems have been widely used in manufacturing industries, thanks to an extensive array of available computer hardware and softrware, as well as the notable reduction of the computer price/performance ratio. Those systems were developed based on relatively new technologies and techniques which include Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Manufacturing Resource Planning (MRP II), Flexible Manufacturing System (FMS), Group Technology (GT), Computer Aided Process Planning (CAPP), Just In Time Inventory Control (JIT), Automated Material Handling (AMH), and many more. When properly planned, managed, and implemented, most of these systems have proven beneficial to their specific area of application. However, a fully operational CIM system has not yet been developed due to the lack of communication or, better, integration between these individual tasks. The result has been a proliferation of heterogeneous hardware, software, and data models, with each application being optimized locally. Instead of firms independently automating as many as 50 different functional areas [1], often using unique hardware and software for each, it is time to adopt a systematic approach to implementing and integrating the various technologies as a means for achieving the productivity gains required [2].

In order to coordinate all of these activities, MRP II may effectively serve as the "hub" of the CIM system [3], as depicted in Figure 1, the proposed functional model of CIM. The links between the various systems are determined by the common information required and the logical rules to regulate the data flow.

Although the concept of CIM has been loosely defined, neither a thorough system architecture nor a requisite integration vehicle is yet available. There are many problems and issues that must be resolved before CIM is possible, among which the database architecture is of utmost importance [4]. There are two primary approaches of designing a database architecture; the first is the "general database", which comprises a single database accessible to all system functions and maintains all system data. The second alternative is "multiple databases" where separate databases are maintained for each function and use some sort of interoperational capabilities between functions. We support, and furthermore endorse, the separate database approach, which facilitates a gradual evolution toward CIM, carries the promise of software individuality, and preserves invested capital and expertise.

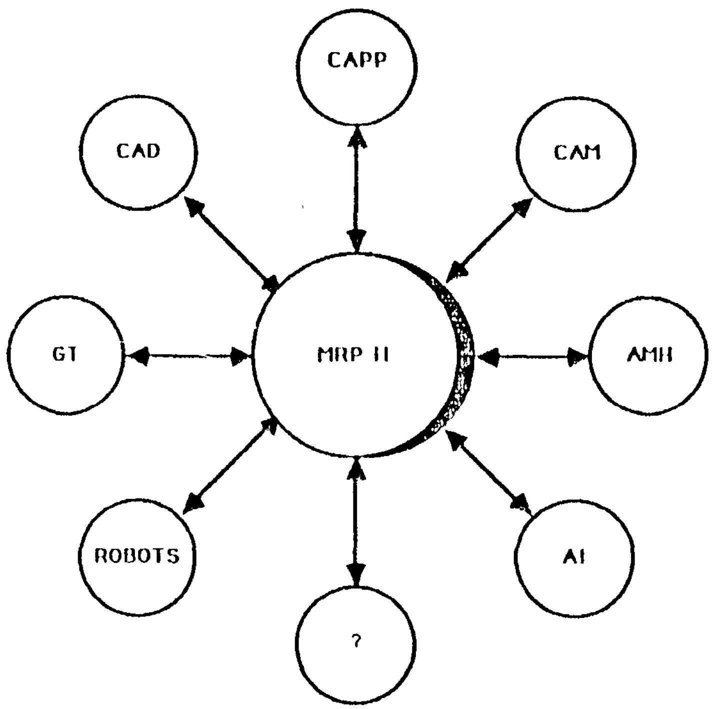

Fig. 1 Functional CIM Model with MRP II as the "Hub"

For a true CIM architecture, a database interoperability system linking all separated databases together must be actively involved in the management of data; i.e., it needs to not only retrieve and update data passively on behalf of application programs, but also to actively control the flow of data and warrant data integrity. The suggested database management system based on this model must be instrumental in decision making and initiate appropriate action to maintain the consistency and integrity of all the databases. Therefore it can be best described as a "knowledge based multi-database interoperability" system which merges artificial intelligence (AI) and database management technologies through extensions of databases.

The following sections give the description of the integrated system model, the Update Dependencies language, a brief program demonstration, and a discussion of our results.

MODEL DESIGN SPECIFICATION. The functions under consideration for this starting point of our CIM model are Computer Aided Design (CAD) and

Manufacturing Resource Planning (MRP II). While neither of these two systems can be called fully mature, their overlap is well established: product definition.

More specifically, the elements common to MRP II and CAD addressed by the integration are the following:

- -Part Specifications.
- -Bills of Material.
- -Engineering Changes.

The functional model of the CAD/MRP II integrated system is based on the similarity of functions and the commonality of data between these two systems, and is intended to operate in a discrete parts, make-to-stock environment.

The part specification data maintained by each system is shown in Figure 2. General part data is maintained for each part and is retrieved by part number; in addition to this data, the effectivity start and end dates and status code (different for each system) for each revision are maintained.

For Each Part Number

CAD	MRP II
Part Number	Part Number
Drawing Number	Drawing Number
Drawing Size	Drawing Size
Description	Description
CAD Unit of Measure	BOM Unit of Measure
	MRP (Purchasing) Unit of Measure
	UOM Conversion Factor
	Source Code
	Cost
	Leadtime
Supersedes Part Number	Supersedes Part Number
Superseded by Part Number	Superseded by Part Number

For Each Revision Level

CAD	MRP II
Part Number	Part Number
Revision Level	Revision Level
Effectivity Start Date	Effectivity Start Date
Effectivity End Date	Effectivity End Date
CAD Status Code	MRP II Status Code
Drawing File Name	

Fig. 2 Part Specification Data Maintained By Each System

Product structures are represented and recorded by the individual parent-component relationships comprising them. The product structure related data maintained by each system are shown in Figure 3.

CAD	MRP II
Parent Part Number	Parent Part Number
Parent Revision Level	Parent Revision Level
Item Number	Item Number
Component Part Number	Component Part Number
Quantity Per Assembly	Quantity Per Assembly

Fig. 3 Product Structure Data Maintained by Each System

It is assumed that no data exists in either system when the integration is established, so that data consistency can be ensured.

The functioning of the model can be represented by examining the status codes associated with each part and revision. The status codes associated with each system are different as shown below.

<u>CAD Part Status</u>

W - "Working"; not a completed drawing, used prior to approval, and not transmittable to MRP II.
R - "Released"; an active part.
H - "Hold"; under review, pending for approval, possibly with a new revision level. Part should not be used by either system.
O - "Obsolete"

<u>MRP II Part Status</u>

R - "Released"; active part.
H - "Hold"; not to be used by MRP II.

The basic functions of the system are described with the aid of status code diagrams, showing the flow of information and the status of each part in both systems during a given activity. In the following, the basic operations using Part Master Records and Product Structures between CAD and MRP II are described through the presentation of appropriate scenarios.

<u>Part Master Records.</u>

In Figure 4, a brand new part is first created by a CAD user as a working drawing with status code "W". At this point, no information about the part exists in MRP II. Upon completion and approval within CAD, the part is released by a CAD user with a CAD status code "R".

If the part supersedes another, the status of the superseded part is immediately changed in CAD to obsolete with a status code 'O', regardless of whether the part previously had an "R" status or an "H" status. In MRP II, the changeover to the superseding part is performed automatically, by virtue of the effectivity start date of any higher level assemblies calling for the new part as part of a revision change, handled by an Engineering Change procedure.

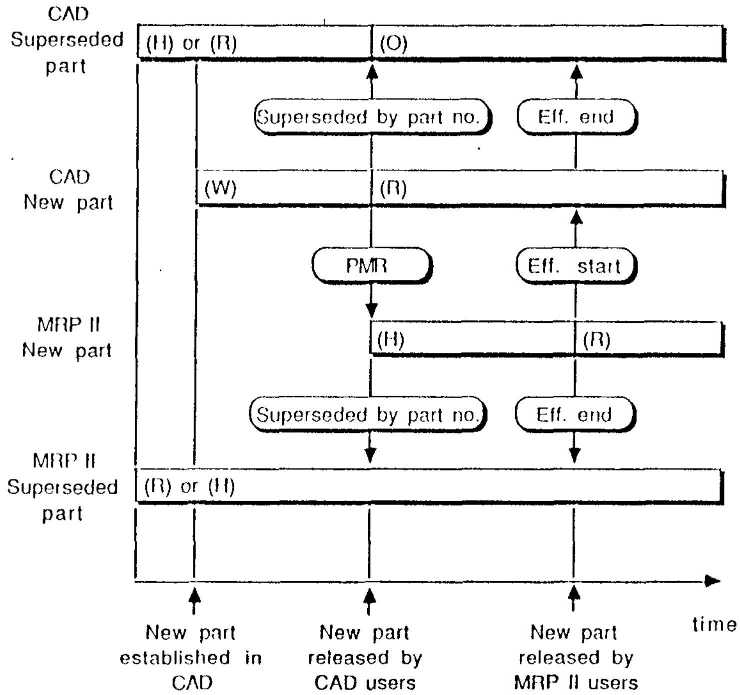

Fig. 4 Status diagram for the Creation of a New Part

The release of the new part within CAD triggers the establishment of a skeletal Part Master Record (PMR) in MRP II using the CAD Part Specification data. Because the PMR is not complete, and to give manufacturing time to plan for the procurement or the manufacture of the part (e.g. to search for vendors or to develop routings), the part is given a status of "H" in MRP II. When an MRP II user completes the PMR, the part can be released within MRP II. If the need arises, due to a machine break down or vendor problems, for example, an MRP II user can place a local hold on the part without affecting CAD. Once held, the MRP II user can again release the part.

Similar diagrams have been developed for part obsolescence, deletion, and changes of revision codes.

Product Structures.

The basic interoperability functions required to maintain consistent assembly or Bill of Material information have also been developed. Typcally CAD is the source of the firat product structure for a given new assembly, as well as the origin of engineering changes requiring the modification of an assembly.

If the addition of a component calls for either a revision change or a new parat number at the assembly level [5], the new revision or part should be created first, and the addition made to the new Bill of Material. This happens regardless of whether the addition is being made via CAD or MRP II. If the new revision or part is not created prior to the operation, the system allows the user to create it as a part of the operation.

The indicated revision of the parent part, or assembly, may have a CAD status of either "working", "hold", or "released"; it cannot have an "obsolete" status, however. Typically, designers will create the product structure of an assembly before it is finalized, i.e., while it has "working" status.

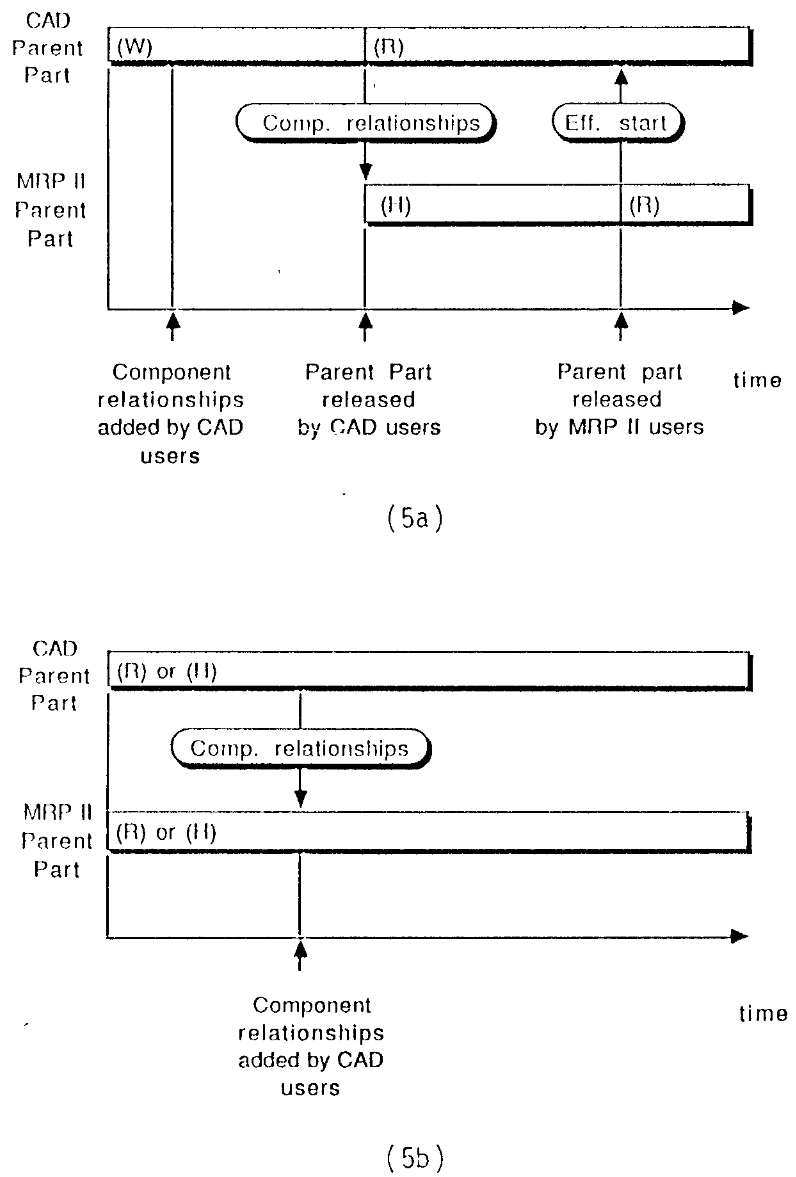

Fig. 5 Status Diagrams for the Creation of Component Relationships

The timing of the transfer of the component relations from CAD to MRP II is dependent on the CAD status of the part/revision combination. The two possible cases are shown in Figure 5. In the first case, shown in Figure 5a, the revision of the assembly being constructed begins with a working status.

This is perhaps the most common situation, as designers typically construct the Bill of Material for an assembly before it is released.

In the second case, depicted in Figure 5b, the revision of the assembly being formed has already been finalized and has either a "hold" or "released" status in CAD.

Similar diagrams have also been developed for deleting component relationships, substituting components in relationships, and modifying component quantities.

UPDATE DEPENDENCIES LANGUAGE. To implement and demonstrate the integrated MRP II/CAD system, the concept of database interoperability is being utilized. This section presents this concept, which is based on the Update Dependency language currently under development in the department of Computer Science at the University of Maryland [6] as a tool for achieving interoperability. The language is being applied to the current problem as a means to critically analyze the design of the integrated system as well as to analyze the effectiveness of the language in specifying such a system. Therefore, a rule set is constructed for the integrated system, called update and retrieval dependencies, which controls inter-database consistancy, through inter-database operation calls. This rule set is used here to enforce the functionality of the integrated MRP II/CAD system.

A relation R is update (or retrieval) dependent on relation S if there exists an update (or a retrieval) on relation R that succeeds only if one or more implied update on relation S succeed.

Update and retrieval dependencies have the following structure:

$op_1(R) \rightarrow cond;$

$op_2\ (S);$

$op_3\ (T);$

$op_4\ (R).$

This is called a compound update operation, and the meaning of it is as follows: operation op_1 on relation R is said to succeed if and only if the condition, "cond", evaluates to be true and the operation op_2 on S, op_3 on T, and op_4 on R all succeed. The operations on the right-hand side may be primitive operations or they may themselves be compound update operations.

The implied primitive operators are: 'add' for adding a new tuple in a relation, 'remove' for eliminating one, 'write' and 'read' for retrieving data by and from the user, 'new' for creating a unique new surrogate, and 'break' for temporarily stopping the system to do some retrieval before giving the control back to the system.

DEMONSTRATION. A sample interactive section demonstrating some of the basic interoperability functions of the MRP II/CAD system is shown in Figures 6-8.

Initially, both the MRP II and CAD databases are empty. The CAD and MRP II part and revision records contain the information specified in Figure 2 and the product structure records contain the information specified in Figure 3.

```
ud> insert(cadpart(Pnum,Dnum,Dsize,Des,Buom,
            Spnum,Sbnum)).

Part Number?
|: 12345.
Description?
|: deluxe_widget.
Unit of Measure?
|: each.
New Revision Level?
|: 1.
Drawing File Name?
|: 'widget.prt'.
Revision Has Been Added
Part Has Been Added

ud> listing(cadpart).

cadpart(12345,unknown,unknown,deluxe_widget,
        each,unknown,unknown).

ud> listing(cadrevs).

cadrevs(12345,1,unknown,unknown,w,'widget.prt').

ud> listing(mrppmr).

ud> listing(mrprevs).
```

Fig. 6 Adding a New Part to CAD

```
ud> releasework(cadrevs(Pnum,Rev,Estart,Eend,
                Cstat,Dfname)).

Part Number?
|: 12345.
Revision Level?
|: 1.

Revision Has Been Released

ud> listing(cadrevs).

cadrevs(12345,1,unknown,unknown,r,'widget.prt').

ud> listing(mrppmr).

mrppmr(12345,unknown,unknown,deluxe_widget,
       each,unknown,unknown,unknown,
       unknown,unknown,unknown,unknown).

ud> listing(mrprevs).

mrprevs(12345,1,unknown,unknown,h).
```

Fig. 7 Releasing a New Part

In Figure 6, a new part is inserted into the CAD database; the user is prompted for the data required to establish a record for the new part and one for its first revision level. The remaining fields, if not specified by the user, are given the value "unknown", as shown in the subsequent listing of the CAD part and revision records. The absence of the MRP II part and revision records verifies that no information has been transferred to MRP II yet.

```
ud> listing(cadpart).

cadpart(12345,unknown,unknown,assemblyA,each,unknown,unknown).
cadpart(12346,unknown,unknown,component1,each,unknown,unknown).
cadpart(12347,unknown,unknown,component2,each,unknown,unknown).

ud> listing(cadrev).

cadrev(12345,1,unknown,unknown,w,file1).
cadrev(12346,1,unknown,unknown,r,file2).
cadrev(12347,1,unknown,unknown,r,file2).

ud> insert(cadcomponet).

Does this change require a new assembly part number? no.
Does this change require a new assembly revision level? no.
Parent part number? 12345.
Parent revision level? 1.
Item Number? 1.
Component part number? 12346.
Quantity per assembly? 1.
in insert cadcomp 1Component relationship has been added to CAD

ud> listing(cadcomponent).

cadcomponent(12345,1,1,12346,1).

ud> listing(mrpcomponent).
```

Fig. 8 Adding Component Relationship to CAD

In Figure 7, the first revision of the part just created is released, and the listing of records after the part was released shows the updating that has occurred in both CAD and MRP II. In CAD, the status of the revision data is changed from working to released; in MRP II, a part master record is created for the new part, as well as a revision record with a status of "hold" for the first revision of the part. Any data not included in the CAD record is given the value "unknown".

In Figure 8, the listings show that one parent part and two component parts already exist in the CAD database; the parent part has a "working" status and the component parts are released. A new component relationship is inserted into the CAD database. The user is prompted to enter the new assembly part number and new assembly revision level, and every other field in the relationship. At this moment, no component relationship exists in MRP II database. To establish this relationship in MRP II, one has to release the parent part.

<u>DISCUSSION.</u> To remain as general as possible, our first CAD/MRP II integration system is a simplification of the activities and data exchange involved in a typical organization. At present stage, the functions of this model are tested on a specific computer without using actural CAD and MRP II systems. This forces the user to interact directly with the interoperability system. As we can see, when our research progresses, the system model will become more and more sophisticated

and practical, as will the interoperability system for a real Computer Integrated Manufacturing system.

Update dependencies provide a convenient formalism for organizing, expressing, and communicating algorithms, and have a declarative representation that allows the designer to express algorithms in a natural manner. This natural representation is appropriate for reviews and presentation since it is a concise statement of the user's notions. Furthermore, this formalism eliminates the tedious and error prone phase of translating algorithm into code.

CONCLUSION. The future of manufacturing industry largely depends upon the successful development and implementation of CIM systems. Our work, starting with MRP II as the nucleus of the integrated system and CAD as its first "satellite", is a wellposed approach through consecutive implementation stages that move from one d/b to two d/bs and eventually to two computers. The information flow between these two systems based on the commonality of part specification, product structure, and engineering changes, is proceeded through the "Multi-database logical rules has been modeled and transformed for the generation and maintenance of part master records and product structures initiated in the product engineering/design division of every typical manufacturing organization. These rules have been translated into update dependencies by using a form of a rule-based expert system, and is being tested for inconsistencies.

Although a "bridge-box" allowing user to hook up any two MRP II and CAD systems is not provided here, we do provide the knowledge needed to specify a "bridge-box" for two given systems, which can be used to integrate existing software packages. The entire scope of the model will now change as it has to encompass three systems rather than two, and issues to be addressed include the consolidation of the integration rules to an absolute minimum and the optimization of information flow between CAD, CAPP, and MRP II.

REFFERENCES.

(1) "The State of CIM," Daniel S. Appleton, Datamation, vol. 30, pp. 66-72.

(2) "The Engineering Reseach Centers as a Tool for Change in the Culture and Attitudes of Academic Engineering," David C. Hazen, Presented at the Second Meeting of the steering Group on Systems Aspects of Cross Disciplinary Engineering, August 8,1985.

(3) "MRP II Providing a Natural 'Hub' for Computer Integrated Manufacturing Systems," Kenneth A. Fox, Industrial Engineering, vol. 16, no. 10, pp. 44-50, October, 1984.

(4) "The CIMS Database: Goals, Problems, Case Studies, and Proposed Approaches Outlined," Michael Melkanoff, Industrial Engineering, vol. 16, no. 11, pp. 78-92, November, 1984.

(5) "Engineering Change for Made-To-Order Products: How an MRP II System Should Handle Them," G. Harhalakis, Engineering Management International, vol. 4, no. 1, pp. 19-36, October, 1986.

(6) "Operational Specification of Update Dependencies," Leo Mark and Roussopoulos Nick, Department of Computer Science, University of Maryland.