SRC TR 89-71

Integration of Product and Process Design with Manufacturing Resource Management

By

G. Harhalakis, and C.P. Lin

Integration of Product and Process Design with Manufacturing Resource Management

G. Harhalakis
Department of Mechanical Engineering and Systems Research Center
University of Maryland

C.P. Lin
Department of Mechanical Engineering and Systems Research Center
University of Maryland

ABSTRACT. There is a critical need for establishing CIM at the factory level, to complement the research done in manufacturing integration, which has concentrated so far on flexible manufacturing cells, robotics and other fabrication and material handling devices. This paper identifies the application modules that clearly lend themselves to an integrated information flow in a controlled manner: Computer aided design and computer aided process planning constitute the product and process design centers of the proposed system respectively. Manufacturing resource planning undertakes the management of production plans to satisfy the market demand. The functional design of the system was derived from expertise in manufacturing management. The modeling and analysis are formalized with the use of generalized Petri-Nets. The implementation strategy recognizes the existence of application tools whose characteristics must be retained and subjected to synergism. Hence, a prolog-based database interoperability language enabled us to construct the knowledge-base that controls the system. Extensions of this work include the incorporation of a shop floor control module, to interface with the factory level.

INTRODUCTION. Two decades ago a technical discussion on integrated manufacturing would most likely have been viewed as an unwarranted intrusion by the technologist into the activities of the manufacturing community. But events have changed those attitudes. The evolving world economic system has demanded new responses. The resulting challenge to U.S. firms is severe. As a consequence, "the whole manufacturing enterprise is undergoing fundamental change. It is becoming more science driven and both uses and produces sophisticated technology... We need to look at manufacturing as an integrated system and optimize it as a total process", [1]. The technical tools used to design both the product and the manufacturing system will be increasingly critical to achieving the desired competitiveness. Creating these tools, however, requires a thorough understanding of the principles of the manufacturing process as a system. It is an unfortunate commentary on the state of development of manufacturing technology that this basic understanding has yet to be achieved. Traditionally, the first step in treating the system as a whole has been to start from considering specific independent sub-systems, that fall within one's specialty, such as design, processes, materials handling and the like. Clearly, this approach has been successful in creating stand-alone modules with little to no synergism.

Although modularity in the design of hardware may be (and has been) as useful as it is in computer software, it creates interfaces that can inhibit the transfer of data or the exchange of information. Old patterns of limited interaction between elements of the manufacturing enterprise must give way to new patterns emphasizing communication and teamwork. We can no longer afford the disruptions and the severe inefficiencies resulting when one unit throws a design, analysis, or test "over-the-wall" to another unit.

Given that the high cost of pilot-scale manufacturing systems often precludes the pilot operation and testing of alternative designs for manufacturing systems, the validity of new system designs becomes critically dependent on the adequacy of the models and the consistency of data. In recognition of the lack of such a mechanism, this paper presents the functional design modeling and implementation of an integrated factory-level manufacturing system, involving three well defined and tested modules: Computer Aided Design, Computer Aided Process Planning, and Manufacturing Resource Planning. The term "factory" here is used to denote the super-set of various manufacturing divisions, which in turn comprise a variety of manufacturing cells. With reference to Gershwin's recent work on a multi-level hierarchical approach, our work is placed one or two levels above the bottom-most fabrication unit. The modeling and analysis of this highly complex system has been effected by using Petri Nets, a proven technique, long used for hardware configurations. We also present the implementation tools, involving an AI data base interoperability language, which provides for data retrieval and updates in a consistent manner.

RESEARCH METHODOLOGY.

1. Phase 1: Functional Design

This system is intended to operate in a discrete-parts, make-to-stock environment, which is the commonest type in the manufacturing sector. With minor modifications it will be able to operate for custom-made products, if necessary.

The first task, which is the development of the detailed functional model of CAD/CAPP/MRP II integrated system, will be based on the similarity of functions and the commonality of data among these three application modules. More specifically, the common elements to be considered are the following:

-- Common elements in CAD, CAPP, and MRP II

* Part Specification
* Bills of Material
* Engineering Changes

-- Common elements in CAPP and MRP II

* Production Routings (Process Plans)
* Work Centers

The common data records maintained among these systems are shown in figure 1.

1.1 Status Codes

To enable better representation and control of the functioning of the model, status codes for each entity in the system will be used, to control the information flow and the entity status changes. Sample status codes related to parts are listed and explained below:

PART RECORD

CAD	MRP II
Part Number	Part Number
Description	Description
Drawing Number	Drawing Number
Drawing size	Drawing size
B.O.M. Unit of Measure	B.O.M. Unit of Measure
—	Purch./Inv. Unit of Measure
—	U.O.M. Conversion Factor
—	Source Code
—	Standard Cost
—	Lead Time
Supersedes Part Number	Supersedes Part Number
Superseded by Part Number	Superseded by Part Number

ROUTING RECORD

CAPP	MRP II
Routing Number	Routing Number
Part Description	Part Description
Unit of Measure	Unit of Measure
Operation Number	Operation Number
Operation Description	Operation Description
Work Center ID Number	Work Center ID Number
Set Up Time	Set Up Time
Machining Time	----
Handling Time	----
Run Time	Run Time
Feed	----
Speed	----
Depth of Cut	----
Number of Passes	----
----	Resource Code
----	Begin Date
----	End Date
Status Code	Status Code

PART REVISION RECORD

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

WORK CENTER RECORD

MRP II	CAPP
ID Number	ID Number
Description	Description
Capacity (HR.)	—
Rate Code	—
Resource Capacity	—
Dispatch Horizon	—
Effectivity Start Date	Effectivity Start Date
Effectivity End Date	Effectivity End Date
Status Code	Status Code
—	Horse Power
—	Speed Range
—	Feed Range
—	Work Envelope
—	Accuracy
—	Tool Change Time
—	Feed Change Time
—	Speed Change Time
—	Table Rotation Time
—	Tool Adjustment Time
—	Rapid Traverse Rate

COMPONENT RELATIONSHIP

MRP II	CAD
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. 1. Common data records maintained among CAD/CAPP/MRP II systems.

PART REVISION STATUS

CAD

W - "Working"	: At a conceptual or preliminary stage, prior to approval, and not transmittable to MRP II.
R - "Released"	: An active part, whose design has been finalized and approved.
H - "Hold"	: Under review, pending approval, possibly with a new revision level. The part should not be used by any other system.
O - "Obsolete"	: The part is obsolete; not to be used by MRP II.

CAPP

W - "Working"	: At a conceptual or preliminary stage, prior to approval.
R - "Released"	: An active part, whose design has been finalized and approved.
H - "Hold"	: Under review, pending approval, possibly with a new revision level.
O - "Obsolete"	: The part is obsolete.

MRP II

R - "Released"	: An active part, whose purchase or manufacture can be initiated in MRP II.
H - "Hold"	: Under review, not to be used by MRP II.

Similar status codes were developed for routings and work centers.

1.2 Development of Expert Rules

By using these status codes and human expertise, many scenarios were and analyzed to cater for all possible transactions between the application modules involved. The entire set of scenarios constitute the working rules of the system. To better illustrate our proposed approach, a sample scenario (Creation of a new part in CAD) is outlined below. CAD is assumed to control the creation and modification of design data, by originating all engineering changes, and is one of the centers from which new parts are introduced. It is also assumed that all part numbers are assigned and controlled solely by CAD for consistency.

When a new part design is first initiated in CAD, the following data are needed to trigger action in CAPP and MRP II.

PART RECORD	REVISION RECORD
* Part Number	* Part Number
* Description	* Revision Level
* Unit of Measure	* Drawing File Name
	* CAD Status Code

With these data, the system performs a series of consistency checks to ensure that the part number has not been assigned by CAD to another part. Initially, the CAD part revision status code is set to "W". The effectivity dates are left unknown to be decided by MRP II users. At the same time, a skeletal routing record is automatically created in CAPP. This will let CAPP know that this part is being worked on, and CAPP will be called upon shortly to interact with design, to assess manufacturability, and to finalize the product structure (for assembly parts). After CAD and CAPP have worked on the part, and the design has been finalized, it will be released ("R") in CAD. The following actions are now initiated:

1. A skeletal part master record for the new part is automatically established in MRP II. Those data fields maintained in MRP II, but not in CAD, are initiated as "unknown" until supplied by MRP II users.

2. Second, a revision record is also established in MRP II by using the part number and revision level from the CAD revision record. The status of the part revision in MRP II is set to "H", since many of the fields required by the MRP II system have been initialized to "unknown", and have to be completed before MRP II can consider the part to be active. MRP II also has to wait for CAPP to release the routing to generate lead time information for manufactured parts. CAPP can release the routings only after CAD releases the part, because finalized process plans are normally completed only after the design is finalized. Also, MRP II must have the part established in its part master record before it can accept the routing from CAPP.

3. Third, CAD checks for a value in the "supersedes part number" field of the CAD part record. If it finds a valid part number in this field, the part number of the new part is inserted into the "superseded by part number" field in the part record of the superseded part. The MRP II part master record is likewise modified to reflect the supersession. The latest revision of the part being superseded, which may have a released status, or which may be on hold, is then made obsolete. If all the preceding steps are successful, the status of the new part revision in CAD is changed to "R".

When CAPP finalizes its process plan, the routing record is given a released status. Once again a series of consistency checks are initiated before the release can be successful:

1. A check is made to make sure the part for which the routing is to be released has a released status in CAD. If it does not, a message to this effect should be generated in CAPP, and the release is not possible.

2. A check is made to make sure that the effectivity dates of all the work centers used in the routings are at least six months (or some other applicable time period) ahead. This ensures that the routings are still applicable when the product must be produced.

3. Finally, it is checked that all the data fields in the CAPP routings file are complete. If all these checks are successfully made, then the routing is assigned an "R" status, and is immediately transferred to the routing module of MRP II with a status of "H".

MRP II now starts working on the part record of the new part in the system. It fills in whatever information required is available, including lead time, by using either the production routing information from CAPP or the vendor information from the purchasing module. The final step in this chain of events is the release of the part and the routing in MRP II, to be used in production.

2. Phase 2: Modeling of The Logic Rules

2.1 PETRI-NETS: Introduction and Basic Notions

Since the pioneering work of Petri who originated his nets in his dissertation "Communication with Automata", [2], Petri-Nets have been increasingly developed and used for modeling a wide variety of systems. This paper introduces their modeling capabilities in the field of Computer Integrated Manufacturing.

Notations

(1) $P = \{p_1,\ldots,p_n\}$ denotes the set of places (represented graphically as circles)

(2) $T = \{t_1,\ldots,t_m\}$ denotes the set of transitions (represented graphically as bars)

(3) A marking is represented by a n-vector of non-negative integer $M = [m_j]_{nx1}$, where the j-th component m_j denotes the number of tokens on place p_j. M^o denotes the initial marking.

In addition to the graphical representation of the Petri-Net, the structure of the net can be expressed in the form of a matrix called an incidence matrix, an example of which is shown in figure 3. It is defined as follows:

$$C = [C_{ij}]nxm \quad \text{where} \quad \begin{array}{l} C_{ij} = -1 \text{ if } P_i \text{ is an input place of } t_j \\ C_{ij} = \ \ 1 \text{ if } P_i \text{ is an output place of } t_j \\ C_{ij} = \ \ 0 \text{ otherwise} \end{array}$$

2.2 The Model

In this section, the scenario of creating a new part in CAD, presented in section 1.2, is modeled using Petri-Net theory, as shown in figure 2. The modeling of this scenario involves a mapping of the logical rules which regulate the data flow between the three application modules, onto a Petri-Net. It is a transformation of logic expressed in written words into a graphical and mathematical form, suitable for analysis. Obviously, the value of the analysis depends on the correctness of this conversion, so it is important there be a formal means of validating the accuracy of the mapping. It will be shown in section 3.1 how net invariants can be used to accomplish this task.

The Petri-Net graph is composed of 33 places and 16 transitions grouped into three sections, corresponding to the three application modules. This physical layout provides a clear graphical representation of the dependencies and relations between the three application systems (CAD, CAPP, MRP II).

The interpretations assigned to the places and transitions are given in figure 3. Due to space limitation, we will not describe the mapping of the entire net. Rather, we will outline some of the results that can be obtained through the analysis of it.

3. Phase 3: Analysis and Evaluation of The CIM Model

After modeling the system with a Petri-Net, the Petri-Net must be analyzed in order to validate the model and to gain insight into the behavior of the modeled system. The use of three analysis techniques (invariants, reachability trees, and behavioral nets) will be described below, always with reference to the sample scenario presented in section 1.2.

3.1 Invariants

Net invariants have been useful in analyzing the Petri-Net model in terms of verifying system properties and locating modeling errors, [3], [4]. An invariant is formally defined as a n-vector, X, of non-negative integers such that:

$$X^TC = 0 \qquad (1)$$

where C denotes the incidence matrix and T denotes the matrix-transpose. The set of places, whose corresponding components in X are strictly positive, is called the support of X and is denoted by $||X||$.

The fundamental property of invariants is the following: X is invariant if and only if for any initial marking M^o and for any reachable marking M:

$$X^TM = X^TM^o \qquad (2)$$

The interpretation of (2) is as follows: the total number of tokens in the set of places $||X||$ (weighted by the components of X) is invariant by any transition firing.

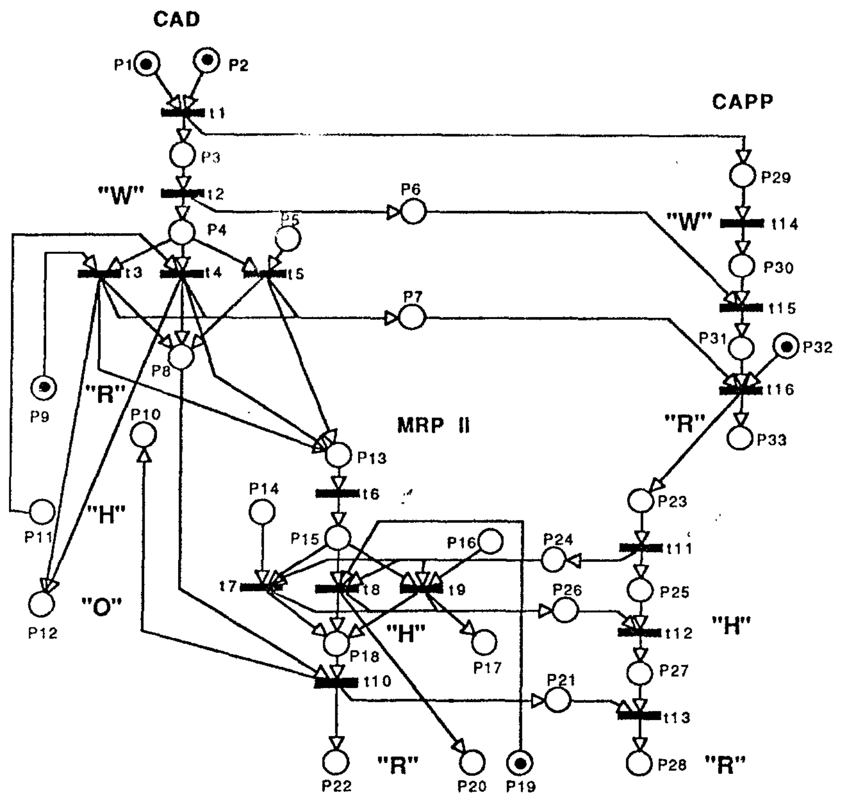

Fig. 2. Petri Net model with initial marking.

PLACES INTERPRETATION

p_1:New part is ready to be entered in CAD.
p_2:New part # has not existed in CAD before.
p_3:New part record and revision record with "W" status in CAD. Part design has not been completed yet.
p_4:Completed part design and completed part record and revision record with "W" status in CAD.
p_5:No superseded part in CAD.
p_6:Completed part design and completed part record and revision record with "W" status in CAD.
p_7:New part record and revision record with "R" status in CAD.
p_8:New part record and revision record with "R" status in CAD.
p_9:Superseded part with "R" status in CAD.
p_{10}:New part record and revision record with "R" status and completed effectivity start date in CAD.
p_{11}:Superseded part with "H" status in CAD.
p_{12}:Superseded part with "O" status in CAD and completed superseded by part #.
p_{13}:Skeletal part record downloaded from CAD.
p_{14}:No superseded part in MRP II.
p_{15}:Skeletal part record and revision record with "H" status in MRP II.
p_{16}:Superseded part with "H" status in MRP II.
p_{17}:Superseded part with "H" status in MRP II and completed m effectivity end date and superseded by part #.
p_{18}:Completed part record and revision record with "H" status in MRP II.
p_{19}:Superseded part with "R" status in MRP II.
p_{20}:Superseded part with "R" status in MRP II and completed effectivity end date and superseded by part #.
p_{21}:Completed part record and revision record with "R" status in MRP II.
p_{22}:Completed part record and revision record with "R" status in MRP II.
p_{23}:Skeletal routing record downloaded from CAPP.
p_{24}:Skeletal routing record with "H" status in MRP II.
p_{25}:Skeletal routing record with "H" status in MRP II.
p_{26}:Completed part record and revision record with "H" status in MRP II.
p_{27}:Routing record with "H" status in MRP II is complete.
p_{28}:Completed routing record with "R" status in MRP II.
p_{29}:Skeletal routing record downloaded from CAD.
p_{30}:Routing record with "W" status in CAPP.
p_{31}:Completed routing record with "W" status in CAPP.
p_{32}:Work center effectivity end dates are at least six months ahead.
p_{33}:Completed routing record with "R" status in CAPP.

TRANSITIONS INTERPRETATION

t_1:CAD user inserts the new part in CAD.
t_2:CAD user completes working on the part design and completes the part record and revision record.
t_3:CAD user releases the part.
t_4:CAD user releases the part.
t_5:CAD user releases the part.
t_6:Skeletal part record and revision record is established in MRP II.
t_7:MRP II user completes the part record and revision record.
t_8:MRP II user completes the part record and revision record.
t_9:MRP II user completes the part record and revision record.
t_{10}:MRP user releases the part.
t_{11}:Skeletal routing record is established in MRP II.
t_{12}:MRP II user completes the routing record.
t_{13}:MRP II user releases the routing record.
t_{14}:Skeletal routing record is established in CAPP.
t_{15}:CAPP user completes the routing and process plans.
t_{16}:CAPP user releases the routing record.

Fig. 3. Interpretation of places and transitions.

A total of 76 invariants were obtained from the Petri-Net described above. By examining these invariants, the accuracy of mapping the initial logic on the Petri-Net was checked. For example, in CAD, a new part is permitted to have a status of working (p_4) or released (p_8) but not both. This was proven to be a property of the Petri-Net model by the invariant i_1 shown in figure 4, which denotes the set of mutual exclusive places $\{p_1, p_3, p_4, p_8, p_{10}\}$.

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	i1	i2	i3
p1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
p2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p3	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
p4	0	1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
p5	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
p7	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
p8	0	0	1	1	1	0	0	0	0	-1	0	0	0	0	0	0	1	0	0
p9	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
p10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
p11	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
p12	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
p13	0	0	1	1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
p14	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1
p15	0	0	0	0	0	1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0
p16	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1
p17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
p18	0	0	0	0	0	0	1	1	1	-1	0	0	0	0	0	0	0	0	1
p19	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	1
p20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
p21	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
p22	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
p23	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	1	0	0	0
p24	0	0	0	0	0	0	-1	-1	-1	0	1	0	0	0	0	0	0	0	0
p25	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0
p26	0	0	0	0	0	0	1	1	1	0	0	-1	0	0	0	0	0	0	0
p27	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0
p28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
p29	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
p30	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
p31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
p32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
p33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Fig. 4. Incidence matrix and three invariants.

3.2 Reachability Tree

Another method used to analyze the Petri-Net, is the construction of the reachability tree as shown in figure 5. The nodes of the tree represent the reachable markings (states) of the Petri-Net while the arcs represent the firing of transitions between the markings. The numbers contained in the nodes of the tree identify the places in the net which are marked with a token. For example, the node at the top of the tree, (1,2,9,19,32), indicates that each member of the set $\{p_1, p_2, p_9, p_{19}, p_{32}\}$ is marked. Beginning with this initial marking, the reachability tree was generated by successively firing the enabled transitions until either no further transitions were enabled or a marking identical to another node was reached.

The reachability tree provides a means of discovering logical conflicts in the system. These logical conflicts can cause blockages in the tree which prevent the branches from properly terminating in the desired final state. For this scenario, the desired final state was reached in all branches of the tree.

3.3 Behavioral Net

The final analysis tool used is the behavioral net, as shown in figure 6. This net is a transformation of the Petri-Net graph (based upon the initial marking) into a form which clearly distinguishes the transitions which fire concurrently from the ones that fire sequentially. In this form, possible redundancies in the logical rules are revealed by finding the redundant places and/or links between transitions and places of the behavioral net. It means that, from a logical point of view, those redundant places and links can be suppressed. However, in the interest of obtaining more information on the state of the system, it was decided to maintain some of the redundancies.

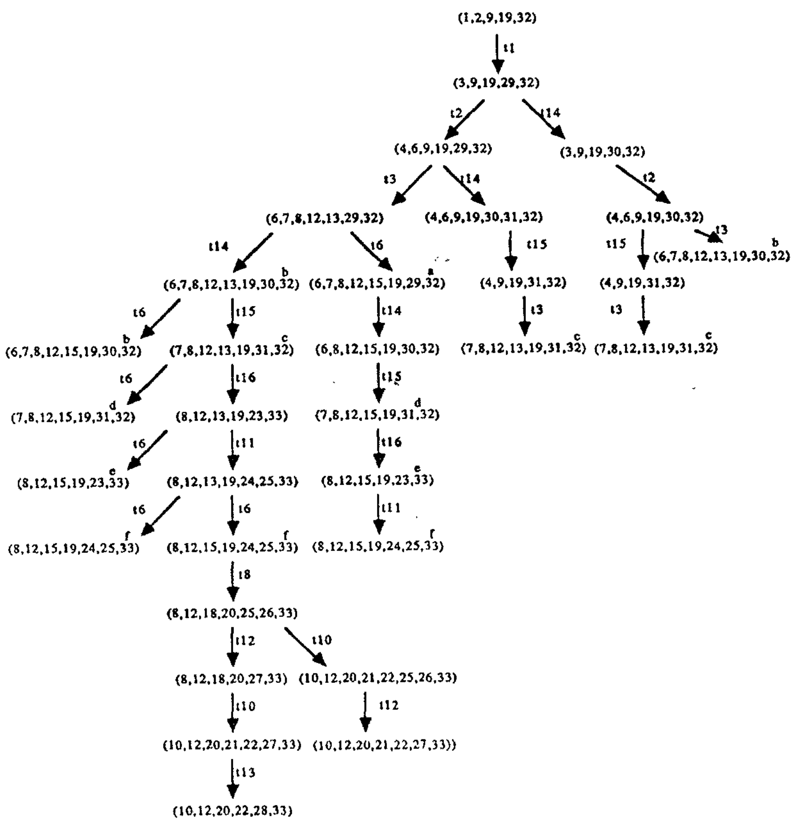

Fig. 5. Reachability tree.

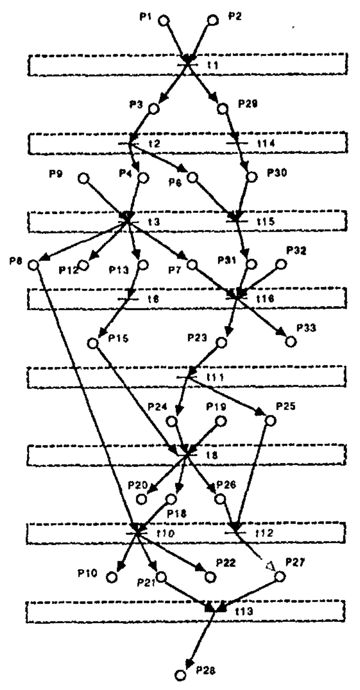

Fig. 6. Behavioral net.

4. Phase 4: Model Synthesis and Implementation

While Petri-Nets have been traditionally used in modeling and analyzing computer systems (hard/software), the application proposed here represents the first time they will be used in the study of CIM at the factory level. One of their major advantages is their hierarchial modeling capability. They can be used to describe the system at different levels of abstraction and detail. For modeling at a more abstract level, sub-nets can be replaced with a simple place or a simple transition, and for more detailed modeling, a place or transition can be replaced with a sub-net, [5]. This procedure is helpful in analyzing the model as it becomes more complex when other scenarios are considered.

One area which is being carefully investigated deals with the modification of the current Petri-Net model to include the handling of multiple parts. This enhanced Petri-Net model will then be able to represent the situation where a CAD user enters a new part into the system while there is another part already in the system.

Another goal of our research is the study and application of reduction methods for the model of the whole system. The goal is to transform the Petri-Net to a reduced net while retaining some desirable properties of the original net, [5]. Information about the highly complex CAD/CAPP/MRP II integrated system can still be obtained by analyzing the reduced net. There have been some reduction methods proposed for different applications, [6], [7], [8] and all of them will be carefully reviewed to develop an appropriate reduction method for the CIM system, in an attempt to reduce the complexity of a fully automated manufacturing environment.

OPERATIONAL SPECIFICATION LANGUAGE. The Petri-Net model of the CAD/CAPP/MRP II integrated system is specified in a high level operational specification language. The result is an operational production system that can be tested, verified and executed. We illustrate the capabilities of the language by continuing the example.

Relation definition:

A relation is defined the following way:

```
define_rel R(A1:D1, ... ,An:Dn).
```

where R is the name of the relation, Ai is an attribute of the relation, and Di is the type of the attribute.

Example 1

The following defines the relation CADREV with attributes Pnum, Rev, Estart, Eend, Cstat, and Dfname.

```
define_rel CADREV(Pnum:INT, Rev:INT, Estart:TIME, Eend:TIME,
   Cstat:CHAR, Dfname:STRING).

define_rel CADCOMPONENT(Pnum:INT, Rev:INT, Itemnum:INT,
                              Compnum:INT, Compqty:QTY).
```

Rule Definition:

A deductive rule is defined the following way:

```
define_rule RULE_NAME(A1=B1, ... ,An=Bn)
->   condition1(B1, ... ,Bn).
->   ...
->   conditionm(B1, ... ,Bn).
```

where conjunction is expressed within conditions and disjunction is expressed between conditions.

Example 2

The following defines a rule CAD_ON_HOLD(Pnum) to be true if the part with part number, Pnum, is in the CADREV relation and its status code, Cstat, is "H" for "Hold".

```
define_rule CAD_ON_HOLD(Pnum=P)
->   CADREV(Pnum=P,Cstst="H").
```

We can define a rule CONTAINS that describes the transitive closure of the CADCOMPONENT relation, i.e. a rule that evaluates to true if a part is a subpart of another part at any level in the design of the part. This kind of rule involves recursion and cannot be expressed in any commercially available database system.

```
define_rule CAD_CONTAINS(Pnum=P, Compnum=SP)
->   CADCOMPONENT(Pnum=P,Compnum=X) and
     CAD_CONTAINS(Pnum=X,Compnum=SP).
->   CADCOMPONENT(Pnum=P,Compnum=SP).
```

Operation Definition:

An operation is defined the following way:

```
<op>
->   <c<1>>,
     <op<1,1>>,
     ..
     <op<1,n1>>.
->   <c<2>>,
     <op<2,1>>,
     ..
     <op<2,n2>>.
->   ..
```

where

<op> is the compound operation being defined,
<c<i>> is a condition on the database state, and
<op<i,j>> is an implied compound or primitive operation.

Example 3

```
insert(CADCOMPONENT(Pnum=P,Rev=R,Itemnum=I,
      Compnum=SP,Compqty=Q))
->  CAD_CONTAINS(Pnum=SP,Compnum=P),
    write("cycle in part design").
->  not CAD_CONTAINS(Pnum=SP,Compnum=P)
    and not CADREV(Pnum=P),
    insert(CADREV(Pnum=P,Rev=R,Estart="unknown",
          Eend="unknown",Cstat="W",Dfname="unknown")),
    add(CADCOMPONENT(Pnum=P,Rev=R,Itemnum=I,
          Compnum=SP,Compqty=Q)).
```

The (very) simplified operation above illustrate a relevant examples of an operation in the integrated system. The insert operation on the CADCOMPONENT relation use the previously defined CAD_CONTAINS rule to decide whether the insertion will cause a cycle in the part design hierarchy. If not, then a skeleton tuple is first created for the designed part in the CADREV relation and the tuple describing the design is added to the CADCOMPONENT relation.

It is important to notice, that the operation specifications above use the high level of knowledge provided by the deductive rules in providing a high level of activity in the system. It is also important to notice, that the condition part of an operation specification is used to enforce consistency rules on and between relations in the system.

Semantics

The semantics of the database specification language is defined as follows.

A compound update operation succeeds if, for at least one of the alternatives in the its update dependency, the condition evaluates to true and all the implied operations succeed. It fails otherwise.

When a compound update operation is invoked its formal parameters are bound to the actual parameters.

Existentially quantified variables are bound by the database system or by the user on request from the database system.

The scope of a variable is one update dependency.

Evaluation of conditions, replacement of implied compound update operations, and execution of implied primitive operations is left-to-right and depth-first for each invoked update dependency.

The execution of 'add' and 'remove' operations done by the system in an attempt to make a compound update operation succeed, will be undone in reverse order during backtracking. This implies, that a (user invoked) compound update operation that fails will leave the database unchanged.

Because logic programming is found difficult by a large class of programmers, we have extended the formalism to include familiar control structures from ordinary programming languages such as while, case, and repeat statements [9].

Implementation

An interpreter for the operational specification language has been implemented in PROLOG in the Computer Science Department at the University of Maryland [10]. The interpreter has been successfully used to implement the quite large rule and operation base for the integrated CAD/CAPP/MRP II system.

The interpreter is currently being re-implemented in the programming language C and will have a considerably better performance than the current version.

The new version will eventually have an integrated Remote Procedure Call (RPC) facility and will allow different copies of the interpreter to communicate both in the same processor and over a network of processors.

A main requirement in the planning of the implementation project has been to keep the already developed parts of the rule base unchanged.

RESULTS. A prototype CAD/CAPP/MRP II integrated system, which includes all the interoperability functions listed in Figure 7, is currently running on Sun workstation. All of these basic functions have been tested and proved to be appropriate according to the previous model specifications and the expert rulebase. When applying to real CAD, CAPP, and MRP II application systems, the overlapping functions in both the interoperability system and the application systems will be moved to the application systems.

CAD	CAPP	MRP II
1 . insert(cadpart)	1 . insert(capppart)	1 . insert(mrppmr)
2 . insert(cadrev)	2 . insert(capprev)	2 . insert(mrprev)
3 . insert(cadcomponent)	3 . insert(cappwc)	3 . insert(mrpcomponent)
4 . modify(cadpart)	4 . insert(capprout)	4 . insert(mrpwc)
5 . modify(cadrev)	5 . modify(capppart)	5 . insert(mrprout)
6 . modify(cadcomponent)	6 . modify(capprev)	6 . modify(mrppmr)
7 . releasework(cadrev)	7 . modity(cappwc)	7 . modify(mrprev)
8 . hold(cadrev)	8 . modify(capprout)	8 . modify(mrpcomponent)
9 . delete(cadpart)	9 . releasework(capprev)	9 . modify(mrpwc)
10. delete(cadrev)	10. releasework(cappwc)	10. modify(mrprout)
11. delete(cadcomponent)	11. releasework(capprout)	11. releasehold(mrprev)
	12. hold(capprev)	12. releasehold(mrpwc)
	13. hold(cappwc)	13. releasehold(mrprout)
	14. hold(capprout)	14. delete(mrppmr)
	15. delete(capppart)	15. delete(mrprev)
	16. delete(capprev)	16. delete(mrpwc)
	17. delete(cappwc)	17. delete(mrprout)
	18. delete(capprout)	

Fig. 7. Current interoperability functions in CAD/CAPP/ MRP II integrated system.

Several of the basic interoperability functions are involved to accomplish the design of a new part as shown in Figure 8, starting from a new part design drawing in CAD, to the release of the part and all manufacturing data associated with it in CAD, CAPP, and MRP II.

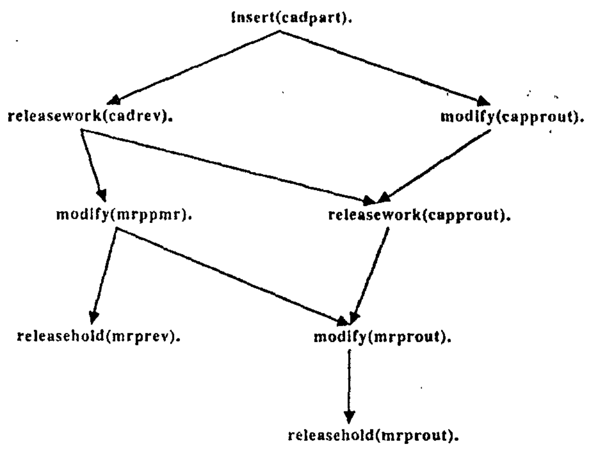

Fig. 8. Basic interoperability functions for the design of a new part

An interactive session demonstrating the interoperability functions of insert part in CAD and release part via CAD is shown in figure 9. Initially, CAD, CAPP, and MRP II databases are empty. A new part is inserted into the CAD database, and 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 of "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. After an internal verification procedure, the first revision of the newly created part is released, and the listing of records after the part was released shows the updating that 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 status "hold" for the first revision of the part. Any data not included in the CAD record are given the value "unknown".

EXTENSION WORK AND CONCLUSIONS. Being differentiated from the other CAD/CAM, CIM, or FMS approaches, a CAD/CAPP/MRP II integrated manufacturing system, using a PROLOG-based AI technique, has been developed to accommodate the need of a factory-level integration. The flexibility of the multi-database interoperability approach makes the system suitable to be used, and able to be extended, with various computer application systems without changing the initial database structures.

The extension work includes the developing of model synthesis and reduction techniques with Petri Nets, as well as appropriate application Petri Net algorithms, useful for the analysis and simulation of general CIM systems. Also, another and maybe the most important module in manufacturing, Shop Floor Control (SFC), will be included in the next version of our system. Several important issues, such as Flexible Manufacturing Cells, Shop Floor Scheduling, Timed Petri Net, etc., will be involved in the future. Some parallel research on resource selection, based on one or more criteria (eg. cost), plant layout based on group technology approach, and feature extraction for process planning based on AI technology, is currently underway at the CIM laboratory of the Systems Research Center, at the University of Maryland.

```
ud> insert(cadpart).
Part Number? b1234.
Description? ball_bearing_assembly.
Unit of Measure? each.
New Revision Level? 1.
Drawing File Name? bearing_folder.
Revision has been added to CAD
Part has been added to CAD

ud> listing(cadpart).
cadpart(1234,unknown,unknown,ball_bearing_assembly,each,unknown,unknown).

ud> listing(cadrev).
cadrev(1234,1,unknown,unknown,w,bearing_folder).

ud> listing(mrppmr).
listing(mrppmr)

ud> listing(mrprev).
listing(mrprev)



ud> releasework(cadrev).
Part Number? 1234.
Revision Level? 1.
Part has been added to MRP II
Revision has been added to MRP II
Revision has been released in CAD

ud> listing(cadrev).
cadrev(1234,1,unknown,unknown,r,bearing_folder).

ud> listing(mrppmr).
mrppmr(1234,unknown,unknown,ball_bearing_assembly,each,unknown,unknown,unknown,unknown,unknown,unknown,unknown).

ud> listing(mrprev).
mrprev(1234,1,unknown,unknown,h).
```

Fig. 9. Demonstration of insert and release part in CAD.

Acknowledgements The authors would like to acknowledge the Engineering Research Center's Program NSFD CDR 88003012 of the National Science Foundation for funding part of this work.

BIBLIOGRAPHY

1.Anonymous, "Design and Analysis of Integrated Manufacturing Systems", Supplement to THE BRIDGE, vol.17, no.2, Summer 1987, National Academy of Engineering.

2.Petri, C. A., "Kommunikation mit Automaten", Ph.D. dissertation, University of Bonn, Bonn, West Germany, 1962.

3.Martinez, J. and Silva, M., "A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri-Net", Lecture Notes in Computer Science, Springer Verlag, vol.52, 1982.

4.Memmi, G. and Roucairol, G., "Linear Algebra in Net Theory", Lecture Notes in Computer Science, Springer Verlag, vol.84, 1980.

5.Lee, K.H. and Favrel, J., " Modeling, Analyzing, Scheduling and Control of Flexible Manufacturing Systems by Petri-Nets", Modeling Production Management Systems, IFIP, 1985.

6.Kwong, Y.S., "On Reduction of Asynchronous Systems", Theoretical Computer Science, vol. 5, 1977, pp. 25-50.

7.Grislain, J. and Pun, L., "Graphical Methods for Production Control", 5th International Conference on Production Research, Amsterdam, August 1979.

8.Lee, K.H. and Favrel, J., "Hierarchical Reduction Method for Analysis and Decomposition of Petri-Nets", IEEE Trans. on System, Man and Cybernetics, vol. SMC-15, no. 2, March/April 1985.

9.Mark, L. and Roussopoulos, N., "Operational Specification of Update Dependencies", Technical Report, Department of Computer science, University of Maryland, College Park, 1987.

10.Mark, L. and Cochrane, R., "Update Dependency Interpreter - Users Manual", Systems Research Center, University of Maryland, College Park, 1987.