Control and Management of Information Flow for Computer Integrated Manufacturing

G. Harhalakis, C.P. Lin
Systems Research Center
University of Maryland, USA

P.R. Muro-Medrano
Electrical Engineering and Computer Science Department
University of Zaragoza, Spain

Abstract

A mechanism to control and manage the information flow among all of the manufacturing application systems, in order to form an integrated manufacturing system, is proposed in this paper. The goal is to achieve a fully integrated manufacturing management system. The INformation System for Integrated Manufacturing (INSIM) reflects a design methodology to build a knowledge base to serve as the information control mechanism. The methodology includes the collection of rules (knowledge acquisition), their graphical modeling, systematic model validation and automated implementation to an operating production system. This design methodology features an enhanced graphic modeling tool - Updated Petri Nets (UPN) - which is capable of modeling database updates and retrievals, under specific constraints and conditions and uses a hierarchical modeling approach. For this purpose an UPN editor was developed which is used to create, explode, unfold, validate and correct the information flow model. Finally, a prototype knowledge based system written in Update Dependencies Language (UDL) - a special rule specification language - was implemented as a result of direct and automatic translation of UPN.

1 Introduction

Current research in the area of manufacturing systems software is quite intensive in dealing with product and process design, production planning, and job execution. However, the design of such systems has been traditionally made in a functional fashion that emphasized "local" solutions, using closed and self-contained architectures. This, together with the use of heterogeneous databases and incompatible computer operating systems, have led to "islands of automation" of various engineering application systems. Naturally, these systems suffer from data inconsistencies and lack of control of functional interactions between them.

Current and future trends for the use of computers in manufacturing include the control and the integration of information flow for production operations into a computer-controlled factory management system. Various research projects in the area of Computer Integrated Manufacturing (CIM) have been conducted by NIST [Jones 86], ESPIT [Meyer 87], CAM-i [Chryssolouris 87], and AT&T [Franks 87]. Most of research projects emphasize on individual aspects of CIM, such as RPI [Hsu 87] on developing a global database framework, TRW [Sepehri 87] on synchronizing the interface between application systems and distributed databases, and U. of Illinois [Lu 86] on developing a framework to perform common manufacturing tasks such as monitoring, diagnostics, control, simulation, and scheduling. These approaches are developing a generic CIM architecture, by creating a global database framework, or by interfacing shop floor activities. However, our research emphasis is the control and management of information flow of production operations to achieve a computer-controlled factory management system. We have developed such a control mechanism, in the form of a rule based system, for managing the information flow among the manufacturing application systems [Harhalakis 90] [Lin 91]. A smilar approach has been taken in [Dilts 91], to develop a framework for integrated CIM database by using knowledge based technology. The system architecture of its integrated CIM databases involves both the distributed database management systems and knowledge based systems for sharing information, and the communication between them is carried out through an integrated interface. Its knowledge base consists of several types of knowledge, including domain knowledge, conceptual data model, logical structure of the database, and data accessibility. Our approach, however, emphasizes the knowledge which reflects the company policy; more specifically, the functional relationships of procedures and operations of the engineering applications involved. This paper also proposes a powerful representation tool which can be analyzed in order to validate the underlying domain knowledge extracted from the company policy and can be implemented into rule production systems automatically.

The second section presents our INformation System for Integrated Manufacturing (INSIM), its architecture and the design methodology. The third section presents the formalism of the modeling tool which was developed to model the knowledge for integrated information systems, and the fourth section presents the modeling methodology. The fifth section describes knowledge verification of the UPN models. The last section presents our conclusions with recommendations for future work.

2 Information System for Integrated Manufacturing (INSIM)

We have concentrated on the control of information flow between each of the key manufacturing applications software at the factory level, including Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), Manufacturing Resource Planning (MRP II), and Shop Floor Control (SFC) systems. These applications form a coherent unit within a manufacturing environment, and the control and integration of them can be

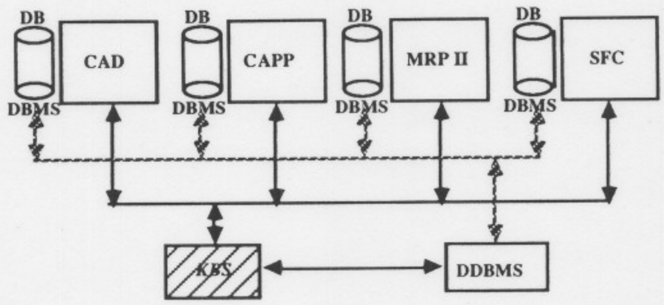


Figure 1: Overall CIM Information Flow Architecture at the Factory Level

seen as the major step for factory automation. The linkage among them is based on data commonalities and the dynamic control of functional relationships between these application systems. The common data entities, which form the basis of the integrated system, can be classified in two categories: Static and Dynamic. The former define the various entities of the distributed system such as parts, products, equipment and processes, while the latter deal with the functioning of the system as it operates to satisfy the market demand. Our goal is to demonstrate the viability of achieving the integration and the control of information flow, using generic operations on generic entities.

2.1 CIM System Architecture

Our CIM architecture concentrates on the integration of manufacturing applications at the Factory level as depicted in figure 1. CAD, CAPP, MRP II, and SFC can be integrated together through a general Distributed Database Management System (DDBMS). The Knowledge Based System, which is the subject of our research, drives the DDBMS to control the information flow, following procedural rules, constraints and other guidelines derived from the company policy. In order to build a prototype of the CAD/CAPP/MRP II/SFC integrated system, we have defined data structures of the common data entities involved in the various manufacturing applications of our integrated system and their relations, which are stored in the DDBMS. Therefore, it can be said that the management and control of information flow is performed by the KBS, while the integration aspect is addressed by the DDBMS.

2.2 Knowledge Base Design Methodology

The methodology for the design and maintenance of a Knowledge Based System(KBS) to control the functional relationships and information flow within the integrated system user-defined rule specifications , reflecting a specific company policy, which is then modeled using a special set of Colored Petri Nets - UPN(Updated Petri Nets) and a hierarchical modeling

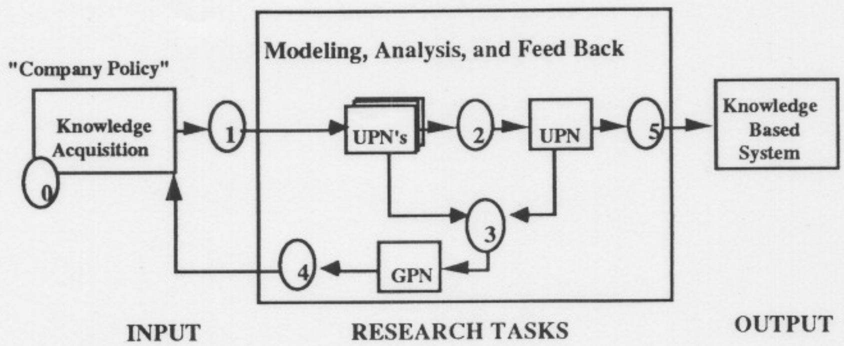


Figure 2: Knowledge Base Design Methodology

methodology. The next step is to convert UPN models into General Petri Nets (GPN) for verification purposes, and feed the results back to the user to resolve (i) conflicting company rules and (ii) errors introduced during the modeling phase. After the model has been validated, it is then translated into a rule specification language. The end result is a knowledge base that controls the data-flow and accessibility between several databases.

3 Structured Modeling of the Domain Knowledge - Updated Petri Nets

We have developed the Updated Petri Nets (UPN), which is a specialized type of the Colored Petri Nets (CPN) [Jensen 86]. In the following paragraphs we present the formal definition of UPN, which is based on both the CP-graph definition and CP-matrix definition given by [Jensen 86].

An UPN is a directed graph with three types of nodes: *places* which represent facts or predicates, *primitive* transitions which represent rules or implications, *compound* transitions which represent metarules (subnets). Enabling and causal conditions and information flow specifications are represented by arcs connecting places and transitions.

Formally, an UPN is represented as: $UPN =< P, T, C, I^-, I^+, M_0, I_o, MT >$, composed of four parts:

1. P, T, C, I^-, I^+, M_0 represent the classic Color Petri net definition. Only this part of UPN is used in the verification process. Its entities are defined as follows [Jensen 86]:

   - $P = \{p_1, ..., p_n\}$ denotes the set of places (represented graphically as circles).
   - $T = \{t_1, ..., t_m\}$ denotes the set of *primitive* transitions (represented graphically as black bars).
   - $P \bigcap T = \emptyset$ and $P \bigcup T \neq \emptyset$.

- C is the color function defined from $P \bigcup T$ into non-empty sets. It attaches to each place a set of possible token-data and to each transition a set of possible data occurrences.
- I^- and I^+ are negative and positive incidence functions defined on $P \times T$, such that $I^-(p,t), I^+(p,t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L \;\; \forall (p,t) \in P \times T$, where S_{MS} denotes the set of all finite multisets over the non-empty set S, $[C(t)_{MS} \rightarrow C(p)_{MS}]$ the multiset extension of $[C(t) \rightarrow C(p)_{MS}]$, and $[\ldots]_L$ denotes a set of linear functions.
- The net has no isolated places or transitions:
  $\forall c \in P, \;\; \exists t \in T : I^-(p,t) \neq 0 \vee I^+(p,t) \neq 0$ and
  $\forall t \in T, \;\; \exists p \in P : I^-(p,t) \neq 0 \vee I^+(p,t) \neq 0$
- M_0 the initial marking, a function defined on P, such that:
  $M_0(p) \in C(p), \forall p \in P$.

2. I_o is an inhibitor function defined on $P \times T$, such that:
   $I_o(p,t) \in [C(t)_{MS} \rightarrow C(p)_{MS}]_L, \;\; \forall (p,t) \in P \times T$.

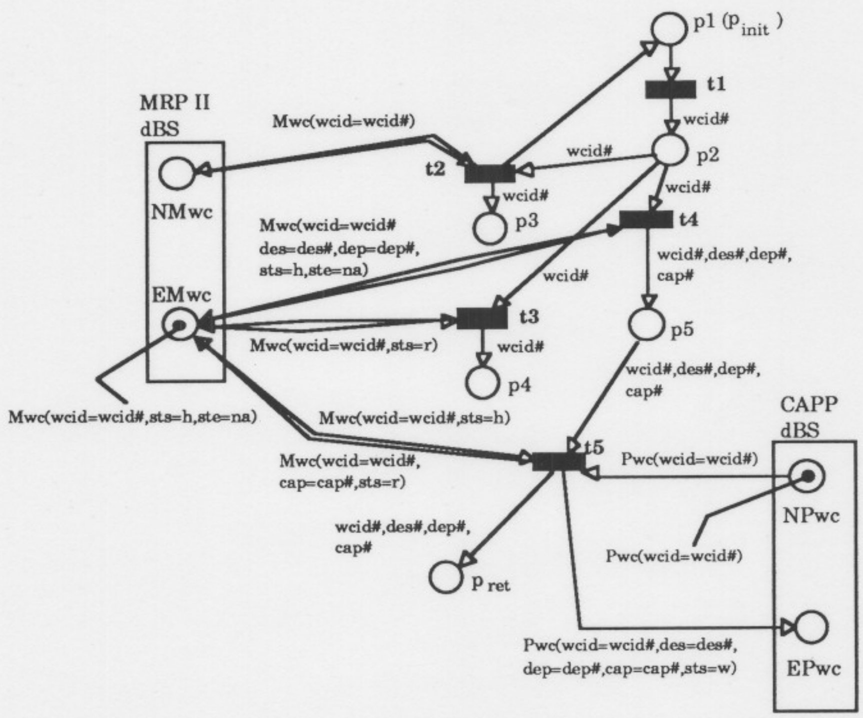
3. $MT = \{hm_1, ..., hm_l\}$ denotes the set of related transition sets. These are sets of transitions grouped into subnets.

Let us now consider the main entities of a model: *Data, Facts, Rules, Metarules. Data* and relations between different data are used in relational database management systems. *Facts* are used to declare a piece of information about some data, or data relations in the system. The control of information flow is achieved by *Rules.* Here, we are considering domains where the user specifies information control policies using "if then" rules. Rules are expressed in UPN by means of transitions and arcs. Metaknowledge, in the form of metarules, is represented by hierarchical net aggregation and net decomposition (compound transition), and will be detailed below.

An example of modeling company policy using UPN is shown in figure 3. It represents the release of a work center in MRP II, and it is described in natural language as follows. Invoking the work center release transaction in MRP II triggers a set of consistency checks: the WC I.D. provided must exist in MRP II with a hold status; all the required data fields should have been filled, and any data fields left out by users are requested at this stage. If all these checks are satisfied, the system changes the work center status code from 'hold' to 'released', and a skeletal work center record is automatically created in the work center file in CAPP, with its status set to 'working'.

4 Modeling Methodology

Generally speaking, any "company policy" starts from the specification of general global rules which describe aggregate operations for a given entity within the system. These rules are then further refined into more detailed specifications on a step by step basis, until no



t1: request and read wcid
t2: write error message and restart
t3: write error message
t4: request other information
t5: update work center record in MRP II dBase
with sts=r, and additional data,
insert a work center record in CAPP dBase

p1(p_init): user starts the transaction
p2: wcid is provided
p3: work center ID does not exist in MRP II
p4: work center already has 'r' status in MRP II
p5: all the necessary data is provided
p_ret: return of the procedure call
EMwc: existence of work center in MRP II dBase
NMwc: non-existence of work center in MRP II dBase
EPwc: existence of work center in CAPP dBase
NPwc: non-existence of work center in CAPP dBase

Figure 3: Subnet of the work center creation scenario: "Release of a work center in MRP II" with initial marking.

aggregate operations are left. Following a similar concept, a hierarchical modeling method using UPN has been developed which allows the system designer to start from abstract global nets and continue with successive refinements until the desired degree of detail has been reached. In addition to the refinement of rules within each scenario, a technique is needed to synthesize all the scenarios to form a coherent net representing the company-wide policy for all entities in the system. The proposed hierarchical modeling methodology facilitates the modeling task. It incorporates a top-down stepwise refinement technique and a synthesis technique.

4.1 Top - down Stepwise Refinement Technique: Vertical Composition of Rules

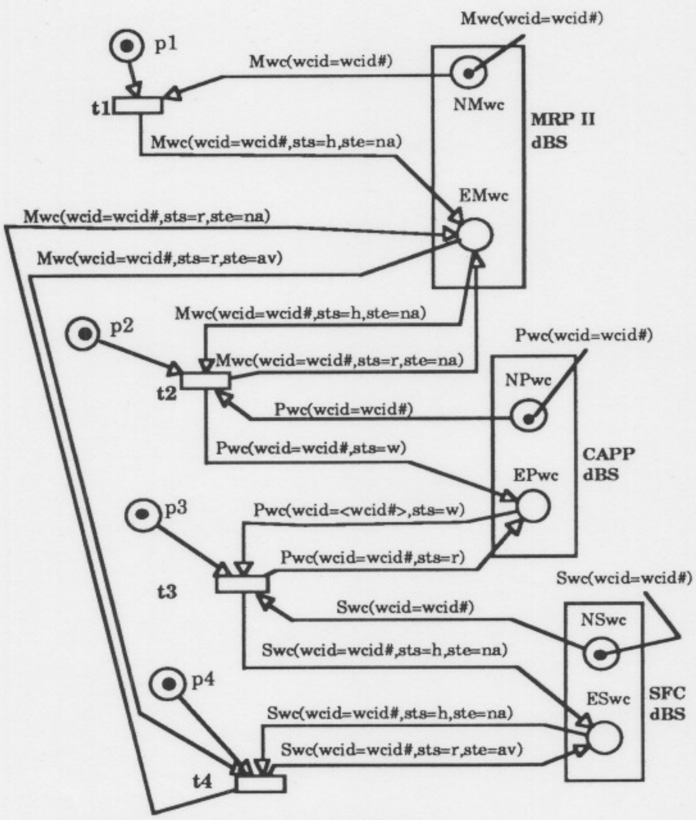
This methodology necessitates the development of new Petri net modeling entities which include two types of transitions, primitive and compound, as mentioned in the previous section; one to represent primitive rules, and the other to represent metarules which can be further refined into subnets. The connections are represented by calls from one compound transition of the net at the abstract level to the subnets at the more detailed level.

An example for the top-down stepwise refinement technique can be seen in figure 4. In that example, a scenario to "create a work center via MRP II" is defined by aggregating the rules: "insert a work center in MRP II", "release a work center in MRP II", "release a work center in CAPP", and "release a work center in SFC". These rules are represented in figure 4 by transitions t_1, t_2, t_3 and t_4. These rules impose updating specifications on MRP II, CAPP and SFC databases which are represented by places $NMwc$ and $EMwc$ for MRP II, $NPwc$ and $EPwc$ for CAPP, and $NSwc$ and $ESwc$ for SFC. UPN net in figure 3 is the refinement of transition t_2 in figure 4.

Vertical compositions in UPN are used as a mechanism to establish relations between one rule at a given level of abstraction and other rules which define knowledge at a lower level of abstraction. Vertical composition allows the composition of rules to form an abstraction hierarchy.

The refinement of a *compound* transition of an abstract net produces a new UPN net which, in general, is the union of both nets minus the refined *compound* transition. A subnet being refined from a *compound* transition is formed with an attached calling protocol which establishes the link from the net at the abstract level with the subnet at the next lower level. The *compound* transition is replaced by the subnet and a calling net which contains one initiation transition (t_{init}), one waiting place (p_{wait}), and one returning transition (t_{ret}). In addition, p_{init} is the starting place of the subnet and p_{ret} is the ending place of it. One arc connecting t_{init} to p_{init} in the subnet and another arc connecting p_{ret} to t_{ret} in the subnet are added to link the calling protocol to the subnet. The calling protocol and the subnet are shown in figure 5.

An example of this vertical composition is the relation between the rule "release a work center in MRP II" (figure 3), and the metarule "create a work center via MRP II" (figure



t1: create a new work center record in MRP II
t2: release the work center in MRP II
t3: release the work center in CAPP
t4: release the work center in SFC

p1: MRP II user starts the create procedure
p2: MRP II user starts the release procedure
p3: CAPP user starts the releas procedure
p4:SFC user starts the release procedure

Database place notations
N'X'wc, E'X'wc
N = does not exist
E = exists
'X' = database: M -> MRP II,
P -> CAPP, S -> SFC
wc = work center

Figure 4: UPN graph of the scenario: "Creation of a work center via MRP II" at an abstract level with initial marking.

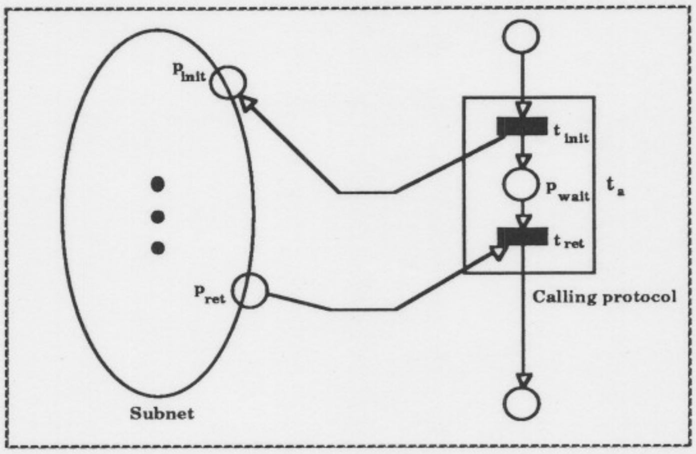


Figure 5: Example of a subnet and a calling protocol

4) which is shown in figure 6. Preconditions and postconditions represented by transition t_2 are preserved (arcs to/from $EMwc$, $NPwc$ and $EPwc$).

4.2 Synthesis Technique: Logic Connection Between Rules

It is necessary to synthesize related scenarios to build the company wide policy, represented by one single net. There have been some synthesis techniques presented in [Narahari 85], [Jeng 90] based on their application domain. In our work we take advantage of the features of UPN, such as global places are presented below. In addition, we take advantage of standard modification procedures embedded in the database management systems associated with each application system. The synthesis of UPN is achieved through the following mechanism:

Global places

Global places are places used to represent facts relevant to (accessible by) different scenarios or subnets. Global places must be referred by the same name in all of their occurrences. Typical examples of global places are facts about database state information, such as places $EMwc$ an $NMwc$ (in the MRP II database) and $EPwc$ and $NPwc$ (in the CAPP database) shown in figure 3. Our UPN models include the representation of database states and every scenario involves checkings, updates and retrievals in some of the databases of the system. Therefore, connections from and to global places, which represent database states, exist in every UPN. These global places provide the connectivities between all scenarios.

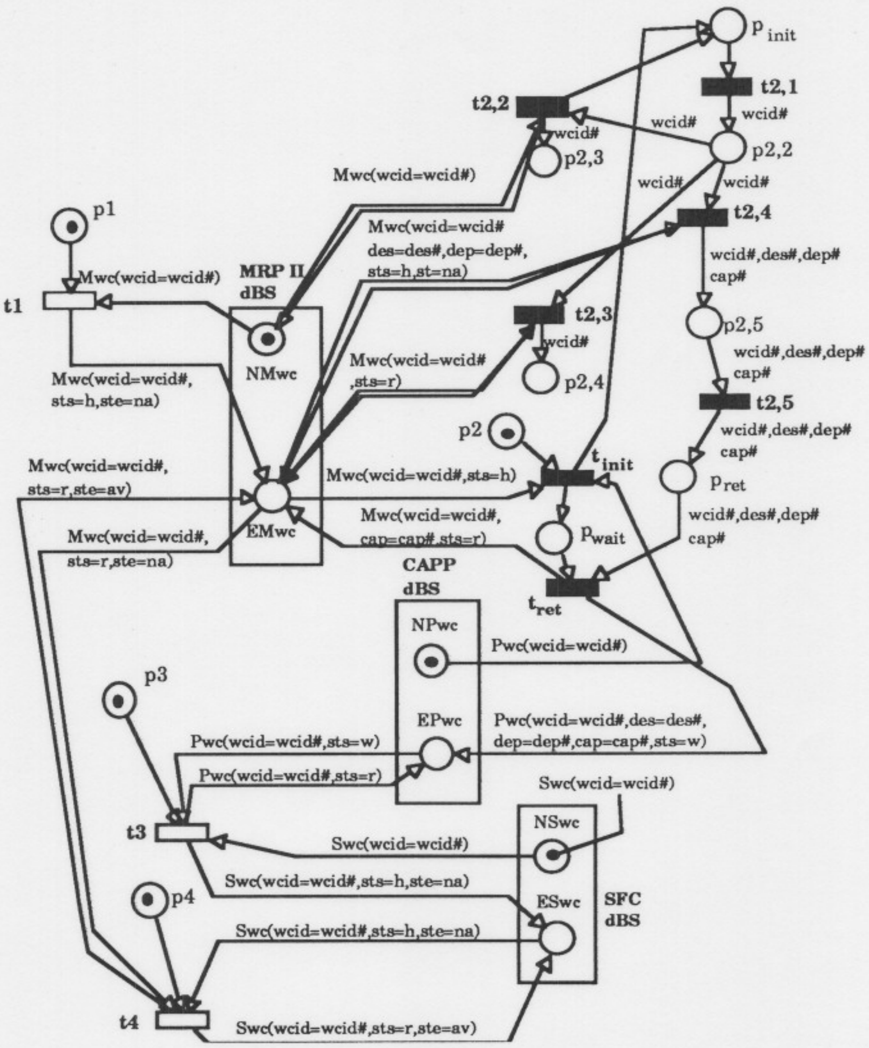


Figure 6: Partially refined UPN of the scenario "Creation of a work center via MRP II" with initial marking.

5 Knowledge Verification

The major objective of creating a KBS using Petri nets is the ability of validating the KBS mathematically and systematically. The *completeness* (dead-end rules, unfirable rules), *consistency* (redundant rules, subsumed rules, under-constrained rules), and *conflicts*, are the major issues in knowledge base verification [Nguyen 87], [Lopez 90]. The incidence matrices of Petri nets representing the rule base can be used to perform some of these verification checks, which can be complemented by the user with the aid of specific domain knowledge. Several other analysis techniques for Petri nets, including reachability trees and net invariants, are also used [Peterson 81] [Jensen 86]. The net invariants, which represent mutually exclusive conditions within the "company policy", can reveal logical conflicts in the specification of the original rules and possibly errors introduced during the modeling process. The reachability tree can be used to detect any deadlocks or inconsistencies in the model.

According to the above discussion, the following two major aspects are considered:

Structural verification focuses on the correctness of the knowledge base structure, which mainly depends on the KBS representation formalism used. In our case, it is the structure of Petri nets. With a formal representation of the KBS, it is possible to verify the KBS structure mathematically. The structural verification does not depend on the domain which the KBS is applied to, or the rule specification language used for the implementation. Thus it is generic for all KBS using the same representation formalism. The following properties can be tested using structural methods. [Nguyen 87]

Completeness : The goal is to detect possible gaps in the rule base that have been overlooked in the modeling process.

Consistency : The goal is to perform a static analysis of the logical semantics of the rule structure.

Domain knowledge verification focuses on assuring a proper behavior with respect to the domain of the system model. It depends on the functionalities of the company policy and can be very different from one application to the other. Therefore, it can not be performed fully automatically; it needs user input to determine the correctness of the functionalities of the system, and the compliance with the company policy

Petri nets have provided traditionally the mathematical background to carry out this kind of verification. Generic analysis methods and properties of Petri nets are shown in the following section, which provide the basis of verifying the KBS.

The staged approach to our KBS verification system, shown in figure 7, consists of three major parts:

1. Generic analysis methods and properties of Petri nets

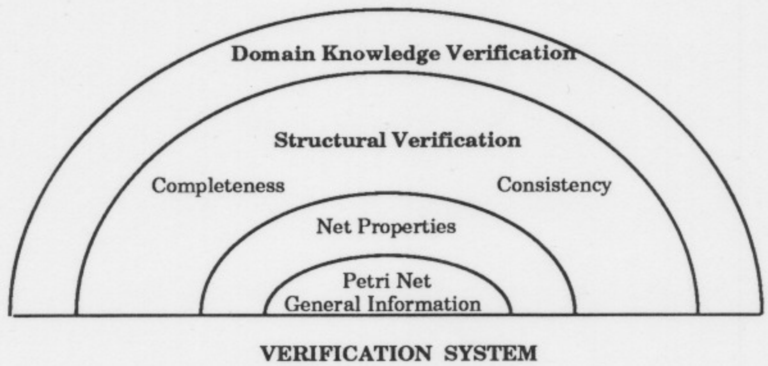


Figure 7: Staged model verification.

2. Structural Verification

3. Domain Knowledge Verification

However, these analysis techniques were initially developed for Generalized Petri Nets (GPN), and do not apply to Colored Petri Nets (CPN) which are characterized by a great diversity of linear functions that are associated to their arcs. Therefore, unlike analysis algorithms for GPN that use integer matrices, analysis algorithms for CPN need to manipulate matrices composed by linear functions. This fact introduces high complexities in the development and execution of these algorithms. Due to this fact, a procedure to convert the high level net of the UPN model into a low level net expressed in terms of Generalized Petri Nets (GPN), has been developed in [Lin 91].

6 Conclusions

The INformation Systems for Integrated Manufacturing (INSIM) design and maintenance methodology has been developed and implemented for generating knowledge based systems, to effectively manage and control the information flow among various engineering application systems. This knowledge base design methodology is fairly generic in that it can be applied to generate knowledge based systems for other applications as well. A formal structured representation schema for rule based systems has also been developed and demonstrated, as it applies to the modeling and verification of company policies. The representation schema, called UPN, is based on the graphical and formal capabilities of colored Petri nets to express and validate if-them rules. UPN are capable of representing user specification rules as well as database updates and retrievals. The implementation strategy aims at facilitating the translation between UPN and UDL and provides a powerful tool to reduce the life cycle of developing new or modifying existing knowledge bases. Changes in existing knowledge bases evolve dynamically as a result of changes in existing company policies.

Future work includes the improvement of the representation schema to consider timing and probablistic firing alternative rules. Also, a combination of UPN with other kinds of Petri net based knowledge representation schemas can be used to model factory layout and behaviour [Muro 89,Muro 91]. It can provide an integrated framework for factory modeling and verification of specifications.

References

[Chryssolouris 87] Chryssolouris, G., "MADEMA: An Approach to Intelligent Manufacturing Systems," *CIM Review*, 11-17(Spring 1987).

[Dilts 91] D.M. Dilts, and W. Wu, "Using Knowledge-Based Technology to Integrate CIM Databases," *IEEE Transaction on Data and Knowledge Engineering, vol.3, no. 2, pp. 237-245* ,1991.

[Doumeingts 87] G. Doumeingts, B. Ballespir, D. Darricau, and M. Roboam, "Design Methodology for Advanced Manufacturing Systems," *Computers in Industry, vol. 9, pp. 271-296*, 1987.

[Franks 87] Franks, R.L., Holtman, J.P., Hsu, L.C., Raymer, L.G. and Snyder, B.E., "Productivity Improvement Systems for Manufacturing," *AT&T Technical Report, vol 66, issue 5*, 1987.

[Harhalakis 90] Harhalakis, G., Lin, C., Hillion, H., and Moy, K., "Development of a Factory Level CIM Model", *Journal of Manufacturing Systems, vol. 9, no. 2, pp. 116-128*,1990.

[Hsu 87] Hsu, C., Angulo, C., Perry, A., and Rattner, L., "A Design Method for Manufacturing Information Management", *Proceedings of Conference on Data and Knowledge Systems for Manufacturing and Engineering, Hartford, Connecticut, pp. 93-102*,1987.

[Jeng 90] Jeng, M. D. and DiCesare F., "A Review of Synthesis Techniques for Petri Nets", *Proceedings of IEEE Computer Integrated Manufacturing Systems Conference, RPI*, May 1990.

[Jensen 86] Jensen, K., "Colored Petri Nets", *Petri Nets: Central Models and Their Properties. Advances in Petri Nets 1986, Part I. Proceedings of an Advanced Course, Bad Honnef, 8-19. September 1986, pp. 248-299*, Edited by G. Goos and J. Hartmanis. Springer-Verlag Berlin Heidelberg 1986.

[Jones 86] Jones, A.T. and Mclean, C., "A Proposed Hierarchical Control Model for Aotomated Manufacturing Systems," *Journal of Manufacturing Systems, vol. 5, no. 1*, 15-25(1986).

[Lin 91] Lin, C.P., "Design, Verification and Implementation of Rule Based INformation Systems for Integrated Manufacturing (INSIM)", *PhD Thesis, Department of Mechanical Engineering*, University of Maryland, College Park, 1991.

[Lopez 90] Lopez B., Meseguer P. and Plaza E.: "Knowledge Based Systems Validation: A State of the Art". *AI Communications, Vol.3, No. 2, pp. 58-72*. June, 1990.

[Lu 86] Lu, S.C.Y., "Knowledge-Based Expert System: A New Horizon of Manufacturing Automation", *Proceedings of Knowledge-Based Expert Systems for Manufacturing in the Winter Annual Meeting of ASME, Anaheim, California, pp. 11-23*,1986.

[Meyer 87] Meyer, W., "Knowledge-Based Realtime Supervision in CIM - The Workcell Controller," *ESPRIT'86: Results and Achievements*, 33-52(1987).

[Muro 89] Muro, P.R., Villarroel, J.L., Martínez, J., and Silva, M., "A Knowledge Representation Tool for Manufacturing Control Systems Design and Prototyping", *INCOM'89, 6th IFAC/IFIC/IFORS/IMACS Symposium on Information Control Problems in Manufacturing Technology*, Madrid, Spain, September 1989.

[Muro 91] Muro, P.R., Ezpeleta, J. and Villarroel, J.L., "Knowledge Based Manufacturing Modeling and Analysis by Integrating Petri Nets", *Proc. of the IMACS Symposium on Modelling and Control of Technological Systems*, Lille, France, 7-10 May, 1991.

[Narahari 85] Narahari, Y. and Viswanadham, N., "A Petri Net Approach to the Modeling and Analysis of Flexible Manufacturing Systems", *Annala of Operations Research, vol. 3, pp. 381-391*, 1985.

[Nguyen 87] Nguyen T.A., Perkins W.A., Laffey T.J. and Pecora D.: "Knowledge Base Validation". *AI Magazine, summer, pp. 67-75*. 1987.

[Peterson 81] Peterson, J.L., "Petri Net Theory and the Modeling of Systems", Prentice Hall, Englewood Cliffs, New Jersey, 1981.

[Sepehri 87] Sepehri, M., "Integrated Data Base for Computer Integrated Manufacturing", *IEEE Circuits and Devices Magazine, pp. 48-54*, March 1987.

[Staff 83] Staff Report, "ICAM Program Overview", Materials Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, Ohio, 1983.