

# DESIGN OF INTEGRATED MANUFACTURING SYSTEMS

*M. Hamdy Elwany\* and Ibrahim M. Buseif\*\**

\* Department of Production Engineering, Alexandria University,  
Alexandria-2144, Egypt

\*\* Industrial Engineering Department, Garyounis University, Libya

## ABSTRACT

This paper addresses the interface between a high-level specification of manufacturing system functions and a model of system behavior. The interface is based on a number of rules transforming an IDEF0 specification into a corresponded Petri Net (PN). The high-level system design was developed using SADT method and Design/IDEFO software package. A generic SADT/IDEFO model of a prototype CIM solution and a software interface enabling a transformation of static models generated by SADT/IDEFO method into dynamic models (PN) through a set of transformation rules are presented. We developed also software, which is used in order to simulate the manufacturing system behavior in a real environment according to its functional specification.

**Keywords:** Manufacturing systems, Petri Net, Computer integrated manufacturing.

## INTRODUCTION

The design of manufacturing systems and the development of manufacturing control systems has become closely linked as the manufacturing environment has been more automated. This is especially true in the case of a FMS and the different forms of Computer Integrated Manufacturing (CIM) systems. Specifications of machining processes, materials handling functions, operational decision-and-control functions by system designers determine the software specifications for manufacturing control system designers who must integrate and automate machines and material handling equipment in accordance with the desired operational decision and control functions. Transition from manufacturing system design to manufacturing control system design is often an awkward process. Part of this problem is lack of standard tools for translating manufacturing system description into manufacturing management and control system specification. This paper focuses on IDEF0 tool to provide a generic

model of a prototype CIM solution. The IDEF0 is recognised as a complete methodology to provide the means of understanding complex manufacturing systems and as the aid of its implementation [1-4]. A software interface enabling a transformation of static models generated by IDEF0 (SADT) method into dynamic models (PN) will be presented. The software could be used to simulate the manufacturing system behaviour in a real environment according to its functional specification.

## SADT/IDEFO METHODOLOGY

SADT/IDEFO provides a structured analysis methodology capable of representing the complex functional relationships graphically and identifying the information and objects that can inter-relate functions. It is now gaining acceptance for CIM modelling applications by the industrial community.

For SADT/IDEFO to be successful, a number of steps must be taken early in the process:

*Bounding the subject* - to ensure that the analyst knows where the system ends and the environment begins. Generating the top-level diagram does this.

- *Determining the purpose of the study.* The models are created to allow the user to answer questions. The questions should be laid down before any modelling occurs, so that the objectives are known and the model can be developed to tackle them.
- *Determining the perspective required.* If the perspective of the model is not defined then the model will cease to be relevant to the purpose.

When the analyst has decomposed the system to a level, which answers the questions initially, set, then the model is sufficient for his purposes. There are some further essential points, which a user should be aware of before attempting to apply the methodology. The design process refers to the actual creation of function specification. When the analyst is satisfied with the model, a detailed specification can then be constructed. This is done by succinctly describing each activity, including non-functional aspects. Descriptions are ordered according to the numbering of the activities. When all activities have been described, a further description is written of how these functions relate to one another. The process is repeated for all the diagrams; the result is then ordered so as to give a complete specification of the system.

#### **A GENERIC FMS DESIGN METHOD USING SADT/IDEFO**

The Design/IDEF Software allows the user to generate IDEFO diagrams linked into set of structured decomposition models with correct diagrams linkage and syntax. This facility both speeds up the process of generating diagrams and models and automatically ensures that the diagrams have correct and logical linkage. The user can also create files of information relating to the system models, which can then be linked to the diagrams. It should be noted that not all ICOM (Inputs, Controls, Outputs

and Mechanisms) arrows that appear on parent boxes in the diagrams also appear on their respective child diagrams and vice versa. The model author must decide how much details is required from a particular diagram, remembering at what level in the decomposition it occurs, and edit out the superfluous arrows accordingly. Let us discuss the generic Computer Integrated Manufacturing System (CIM) model showing details of main activities (areas) and some of their decomposition's (in this paper we will show only first four pages from sixteen pages of CIM model generated using IDEFO).

#### **Activity A-0 (A minus 0): Design a CIM System and Manufacturing Parts (page 1).**

A-0 is the top diagram in the model so it has only one box with the main inputs, outputs, controls and mechanisms. Also on this top diagram the purpose and the viewpoint are defined. Compiling a model from a fixed viewpoint ensures that consistent system descriptions are used with appropriate terminology and emphasis. The purpose of compiling the model is also set out on the top, page since this determines the required level of detail that the model must show.

##### Inputs

- *Expertise* is an essential input designing a CIM system.
- *Existing CIM System Information* is another important input because there is no point in redesigning something that already exists. The system designer can study this existing systems information and apply it to his design, utilising its strength points and attempting to eliminate its weak points.
- *Available Resources* (equipment and personnel) are used in CIM system, if required.

##### Controls:

- *Approved Resources* will constrain a project since it will have a finite budget within which it must work.
- *Available Technology* naturally constrains the system design since it can not use technology that does not exist or has not yet been properly developed.

*User Requests.* This refers to the requirements of the organization that is going to use the CIM when built.

- *Commercial Software Constraints* is very important constraint because governing of all activities for design, analysis and selection of things.

### Outputs:

- *CIM System Design* (goals and new system description), upon which the system planning is based, are outputs of the planning process.
- *CIM System Models* (static and dynamic) upon which the system implementation is based, is also an output of the design process.
- *System Performance* is the result of the design improving system user needs.
- *Complete Parts* (products) manufactured according to its specifications is basic system output.

### Mechanisms:

- *CPM/PERT, Gantt etc.* are examples of the basic tools of the project management which a person in charge of a project of this size would be well advised to use.
- *IDEFO, DFD* are static systems modelling techniques used in the early stages of system design to be define the system.
- *Petri Net (PN), Experts, etc.* are dynamic modelling techniques used in the later stages of the system.
- *Simulation Software* is used after the modelling or designing things. *People* (staff)
- Performing different jobs (e.g. parts and process design, system supervision and maintenance, loading/ unloading parts, etc.).

### **Activity A0: Design a CIM System and Manufacturing Parts (Page 2)**

Box A0 is the first decomposition of the IDEFO model and so it shows the main areas of the system. All of ICOM arrows that appear on the top diagram appear on this decomposition, plus a few others that were not required on the top diagram. All of the boxes on this diagram have been decomposed further in the model.

- *Activity A1- Plan CIM System:* In this activity all requirements and needs of the CIM to be designed must be determined.
- *Activity A2- Design CIM System:* Design and specify CIM system by using standard components as much as possible (Design CAD, Design CAM, and Design FMS).
- *Activity A3 - Integrate CIM System:* This means not just interface, but also integrate, such that the in-formation exchange, knowledge base and processes become part of a meaningful communication.
- *Activity A4-Design and Manufacture Parts:*
- This activity specify the design and manufacture process, process planning, cost evaluation, the flexibility of the system and extension of the system.

### **Activity A2: Design CIM System (page 3)**

This page breaks model down into three main functions of designing CIM systems:

- Activity A21: Design CAD System
- Activity A22: Design CAM System
- Activity A23: Design FMS System

### **Activity A23 Design FMS System (page 6)**

On this page designing FMS system is decomposed into four functions:

- *Activity A231: Study Requirements* according to the goals to be attained.
- *Activity A232: Analyse and Review Current System* to get a new FMS specifications.
- *specifications and updated system data,* the integrated FMS to get a new FMS design.
- *Activity A234: Test and Analyse New System* according to the needs and objectives.

### **Activity A4: Design And Manufacture Parts (page 10)**

This page breaks model down into four main areas of design and manufacturing parts.

- *Activity A41: Plan Manufacturing* of work-pieces or parts in the drawing form according to the needs or requirements of the work.

Activity A42: Manufacturing Process - specifies the sequence of manufacturing process steps and describes the state of parts at each workstation by using CAD/CAM software.

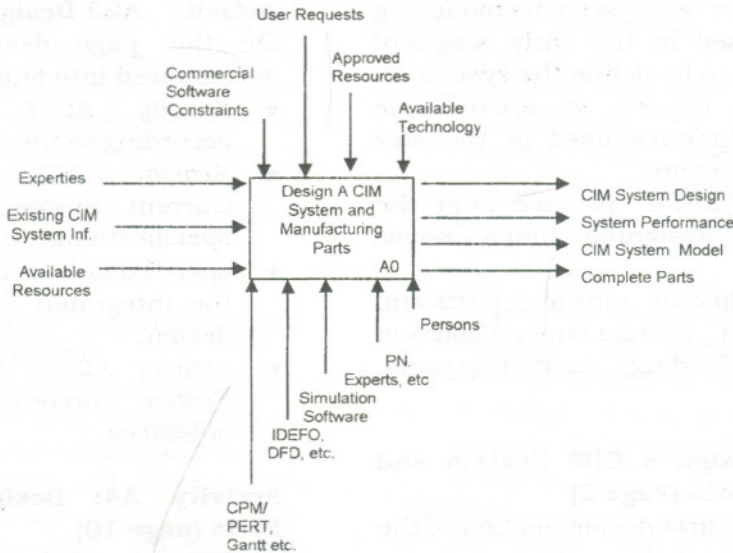
- Activity A43: Simulate Operations - use simulation software for parts to be manufactured at each workstation to check the validity or the performance of manufacturing process at each workstation.
- Activity A44: Manufacture Parts at subsequent workstation by moving them by selected material handling equipment.

It is possible to specify the system constraints and system resources (machines number, buffers capacities, number of operators, etc.) directly from the IDEF0 model. After subsequent decomposition of the model it is also possible to define

operational time for all system functions. The number of decomposition levels has a great impact not only on data gathered from the model but also on the future application of a model. In particular after decomposing a system into elementary functions which can be performed by specific system resources (system elements: machines, robots, material handling equipment, sensors, computers, data bases, control software, etc.) the dynamic model describing system behaviour can be directly obtained from SADT/IDEFO models.

Figure 1 shows the page structure of the generic CIM solution IDEF0 model is shown. The main activities of the model are discussed in the forthcoming pages (modeling software used: Design / IDEF package ver. 3.0 by Meta Software Corp.).

USED AT:	AUTHOR IBRAHIM BUSEIF PROJECT CIM SOLUTION REV:	DATE: 1/15/9	WORKIN	READE	DATE	CONTEXT:
			DRAFT			TOP
			x RECOMMENDE			
			PUBLICATIO			
NOTES: 1 2 3 4 5 6 7 8 9						

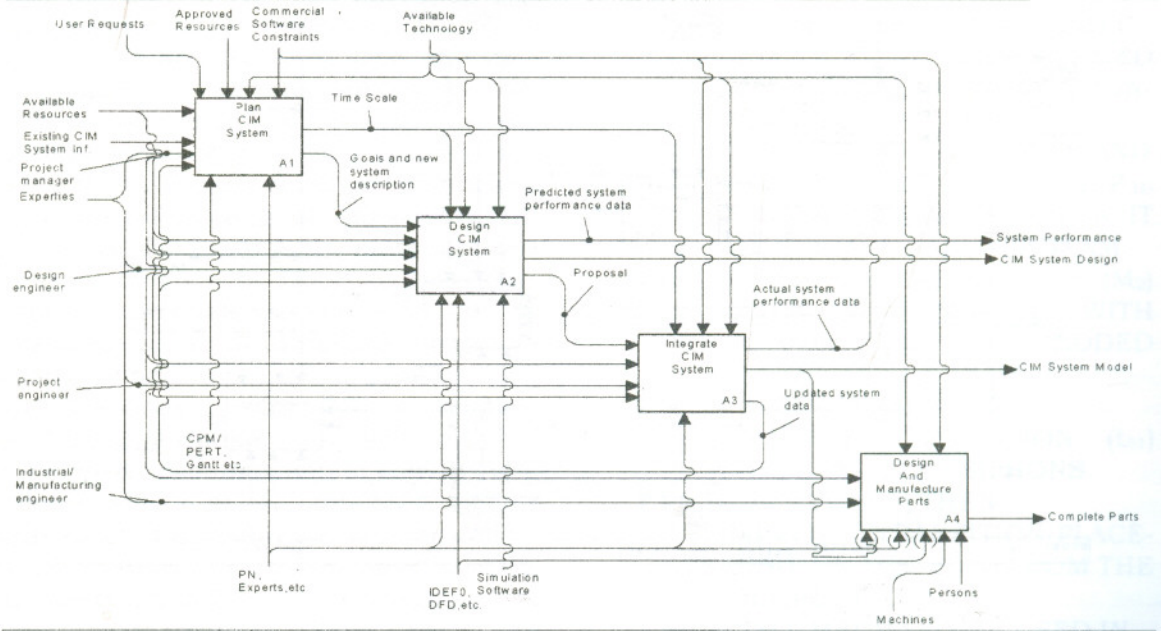


Purpose: Design and Simulate CIM and Manufacturing Parts  
Viewpoint: The System Designer and Prototype CIM Solutions

NODE: A-0	TITLE:	NUMBER: P.1
-----------	--------	-------------

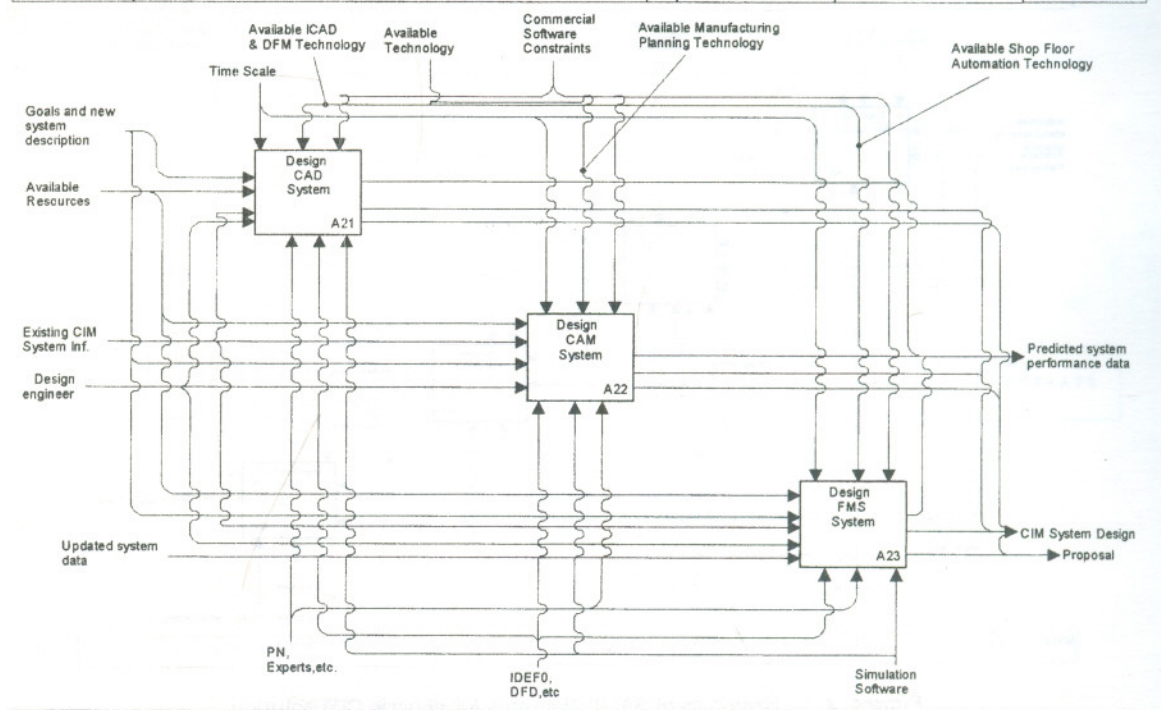
# Design of Integrated Manufacturing Systems

USED AT:	AUTHOR: IBRAHIM M. BUSEIF	DATE: 1/16/96		WORKING	READER	DATE	CONTEXT:
	PROJECT CIM SOLUTION	REV:		DRAFT			☒
	NOTES: 1 2 3 4 5 6 7 8 9			x RECOMMENDE			
				PUBLICATIO			



NODE: A0	TITLE: Design A CIM System and Manufacturing Pa	NUMBER: p 2
----------	---	-------------

USED AT:	AUTHOR IBRAHIM M. BUSEIF	DATE: 2/04/9		WORKIN	READE	DATE	CONTEXT:
	PROJECT CIM SOLUTION	REV:		DRAFT			☐
	NOTES: 1 2 3 4 5 6 7 8 9			x RECOMMENDE			☐ ☐
				PUBLICATIO			



NODE: A2	TITLE: Design CIM System	NUMBER: P.3
----------	--------------------------	-------------

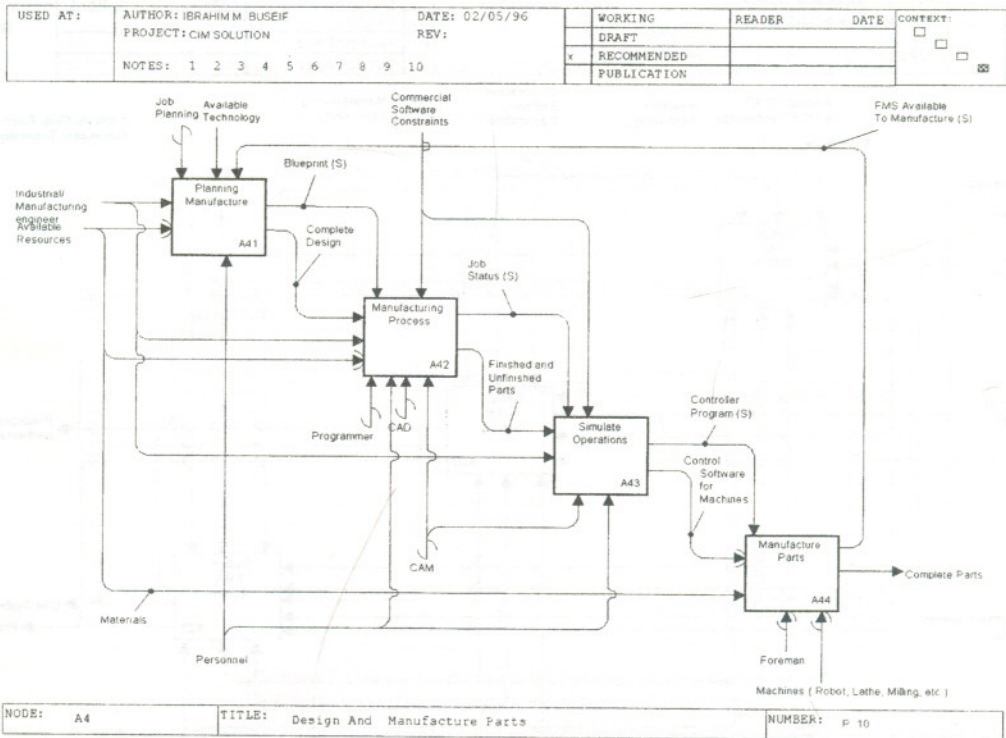
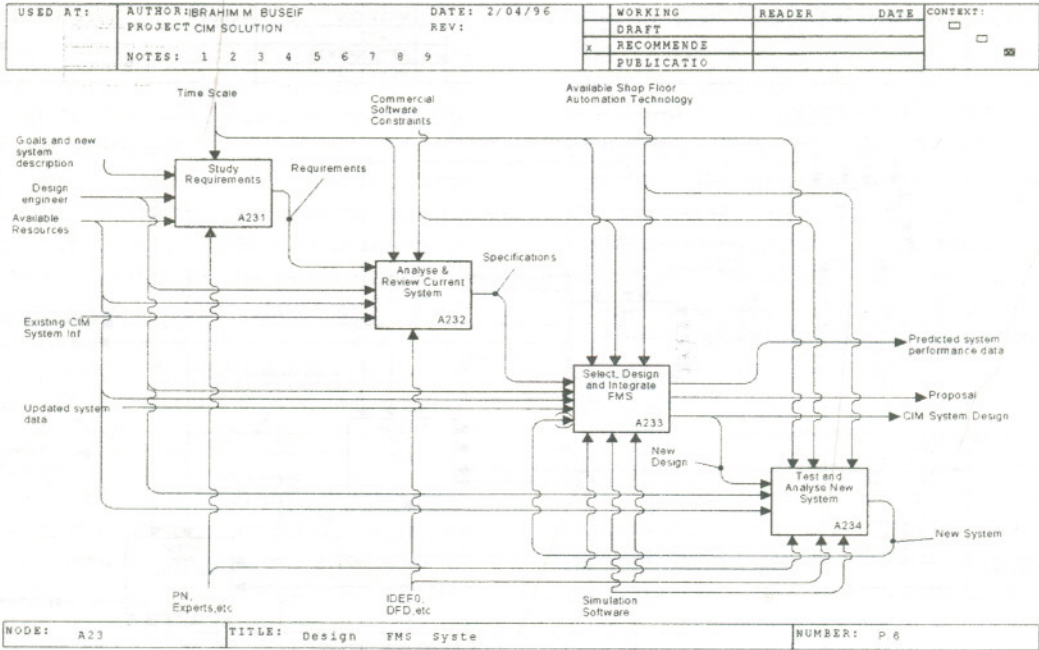


Figure 1 Examples of SADT diagrams for generic CIM solution

**TRANSFORMATION RULES**

We start with the IDEF0 (SADT) model for the specification of the cell operation. An example of the IDEF0 model is shown on Figure1 (P.10, activity A44). The activities (boxes) represent various operations that form the sequence of events in the machining of parts. For each activity, the mechanism is the machine or equipment used in the performing of this activity. For each activity, the input is the material upon which the activity acts. There are two kinds of outputs. The material flow is an output that becomes an input to the next operation or activity. In addition, there are information outputs that are controls for other activities. All of these information you will get after more detailed decomposition of activity A44. The developed software interface models through a set of transformation rules enable a transition from static (descriptive) models, which are generated by SADT diagrams into dynamic (behavioural) Petri net. The software was written in *Turbo Pascal* Figure 2.

The high-level design interface is established through SADT methodology. The design specification is then transferred into

a Petri net through a set of following transformation rules [2, 3, 5].

- T.R1: IF ACTIVITIES (A AND B AND C AND D) THEN TRANSFORMED INTO A PETRI NET TRANSITION-PLACE-TRANSITION.
- T.R2: IF ARROWS ( $O_1-S$  AND  $O_2-S$  AND  $O_3-S$ ) THEN PN PLACE (e.g.  $P_{O1-S}$ ) WITH RESIDE TOKEN IN IT ADDED TO THE PN DIAGRAM.
- T.R3: IF SHARED MECHANISM ( $M_2$ ) THEN PN PLACE ( $P_{M2}$ ) WITH RESIDE TOKEN IN IT ADDED TO THE PN DIAGRAM.
- T.R4: IF ACTIVITY (A) THEN ELIMINATE TRANSITION ( $t_{A1}$ ) AND TREATED AS SIPHONS.
- T.R5: IF ACTIVITY (E) THEN ELIMINATE TRANSITION-PLACE-TRANSITION ( $t_{E1-pE-tE}$ ) FROM THE PN DIAGRAM.
- T.R6: IF (ALL MECHANISMS USED IN THE PN DO NOT EXIST IN ANY SADT DIAGRAM NOT INCLUDED IN THAT PN) THEN SADT-TO-PN IS COMPLETE.

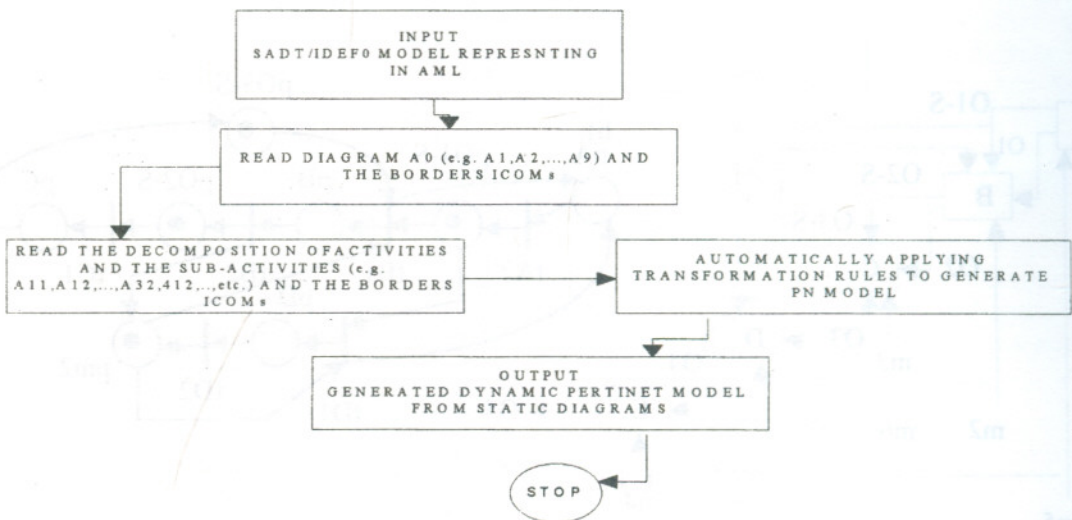


Figure 2 Structure of the developed software interface

The employment of interface software and transformation rules are illustrated in the example shown in Figure 3.

The tokens, which are generated from the transformation rules often, represent the objects or resources in the modelled system.

**SOFTWARE FOR A FUNCTIONAL ANALYSIS OF A PETRI NET MODEL**

The purpose of software development was to analyze the functional properties of the manufacturing system, which was designed by using SADT method and which behavior is described by Petri net as seen in Figure 4. The developed software enables analyzing a system for its various properties irrespective of time. Functional analysis has been used in the Petri net to refer to this type of analysis. The software analysis is based on some approaches used to determine the various properties of PN model: liveness, safeness, reachability and conservativeness [1,3].

**The Reachability Tree**

The reachability tree represents the reachability set of events in a Petri net. Every marking in the reachability set will be produced, and so for any Petri net with infinite reachability set, the corresponding

tree would also be infinite. Even a Petri net with a finite reachability set can have an infinite tree. The tree represents all the possible sequences of transition firings. Every path in the tree, starting at the root, corresponds to a legal transition sequence [6]. The main idea is to use the reachability tree to generate all the possible markings that can be reached as seen in Figure 5.

Each node  $i$  ( $i=1,2,\dots$ ) in the tree is associated with an extended marking  $\mu(i)$ . The marking is extended to allow the number of tokens in a place to be nonnegative integer. Each node is also classified as [6]:

- Frontier nodes - are nodes which have not yet been processed by the algorithm (the algorithm begins by defining the initial marking to be the root of the tree and, initially, a frontier node). They are converted by the algorithm to terminal or duplicated nodes.
- Duplicate nodes represent another class of markings, which have previously appeared in the tree.
- Terminal (deadlocks) nodes with markings in which no transition is enabled.

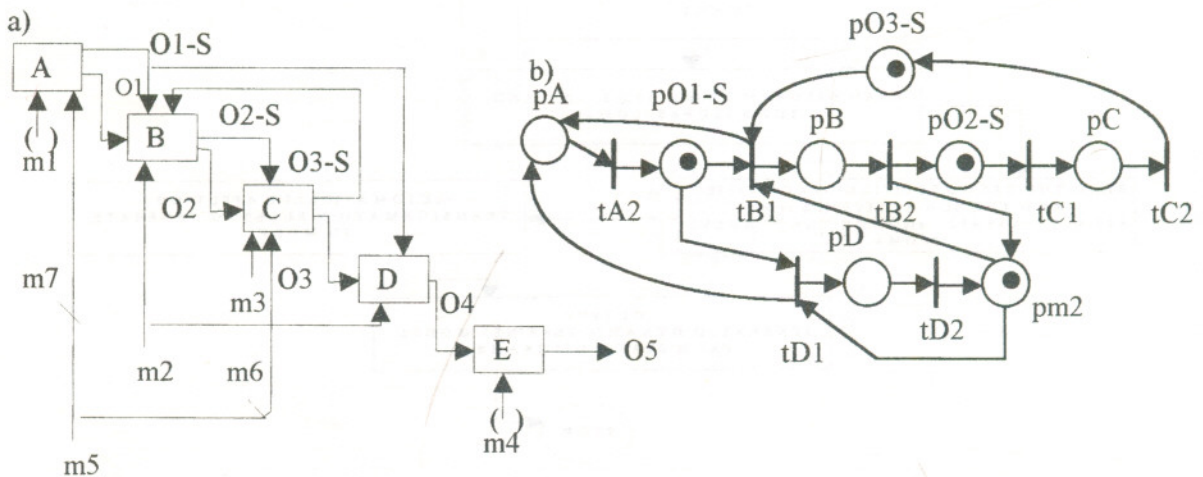


Figure 3 An example of the transformation rules



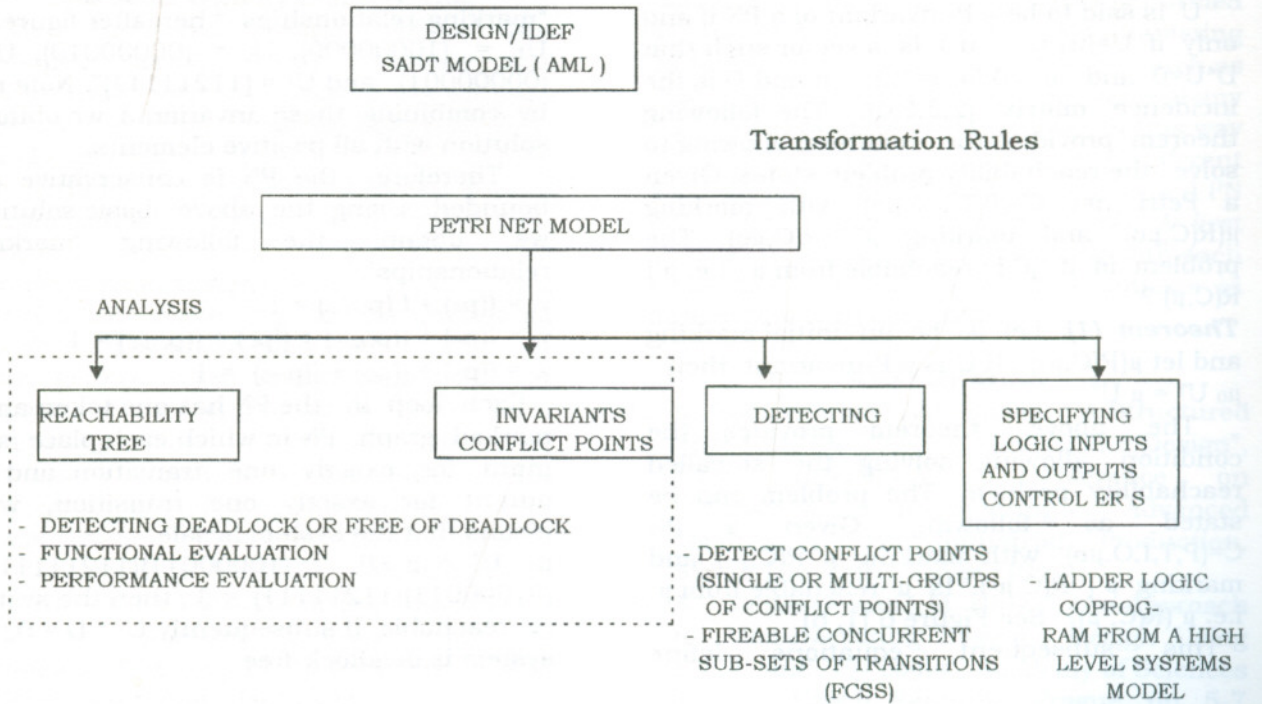


Figure 4 Structure of the developed software interface

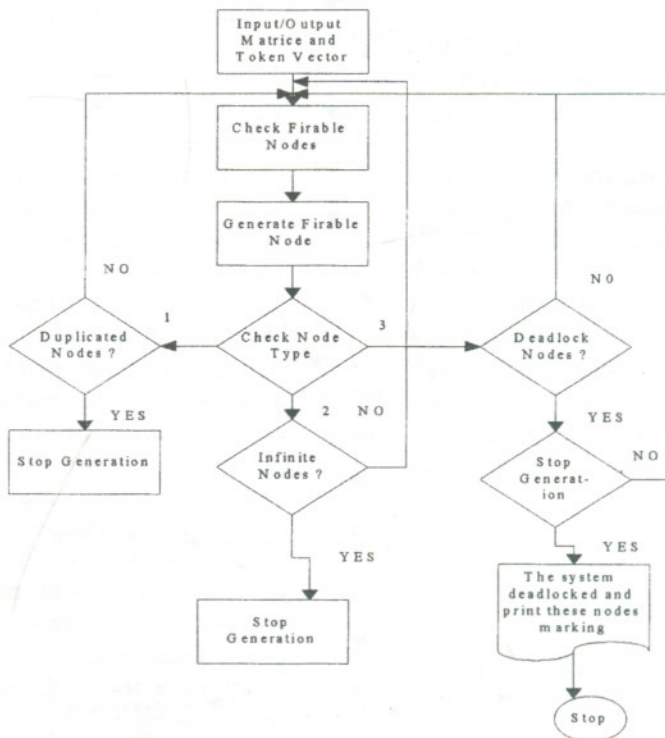


Figure 5 The deadlock detection processing logic

**P-Invariant**

U is said to be a P-invariant of a PN if and only if  $U=(u_1, u_2, \dots, u_n)$  is a vector such that  $D*U=0$  and  $u_i \geq 0$  for  $i=1, 2, \dots, n$  and D is the incidence matrix [2,3,5,6]. The following theorem provides the condition allowing to solve the reachability problem states. Given a Petri net  $C=(P, T, I, O, \mu_0)$  with marking  $\mu(R(C, \mu_0))$  and marking  $\mu'(R(C, \mu_0))$ . The problem is if  $\mu'$  is reachable from  $\mu$ , i.e.  $\mu' \in R(C, \mu)$ ?

**Theorem (1):** Let  $\mu_0$  be an initial marking and let  $\mu(R(C, \mu_0))$ . If U is a P-invariant, then  $\mu_0 U^T = \mu U^T$

The above theorem provides the condition allowing solving the so-called reachability problem. The problem can be stated as following. Given a PN  $C=(P, T, I, O, \mu_0)$  with marking  $\mu(R(C, \mu_0))$  and marking  $\mu'(R(C, \mu_0))$ . Is  $\mu'$  reachable from  $\mu$ , i.e.  $\mu' \in R(C, \mu)$ ? See Figure 6 [1, 6]

This subsequent equations define

relationships between markings, called "marking-relationships" hereafter figure 3 b:  $U_1 = (10000000)$ ,  $U_2 = (00000010)$ ,  $U_3 = (00000001)$  and  $U^T = (11211111)^T$ . Note that by combining these invariants, we obtain a solution with all positive elements.

Therefore, the PN is conservative and bounded. Using the above basic solutions we obtain the following 'marking-relationships':

$$\gamma_1 = ((p_A) + ((p_{O1-S}) = 1$$

$$\gamma_2 = ((p_B) + ((p_{O2-S}) + ((p_C) + ((p_{O3-S}) = 1$$

$$\gamma_3 = ((p_B) + ((p_D) + ((p_{Tn2}) = 1$$

Each loop in the PN has one token and a marked graph PN in which each place is an input for exactly one transition and an output for exactly one transition, what means that the system is safe.

$\mu_0 U^T = \mu U^T = (10000011)(11211111) = (01000011)(11211111) = 3$ , then the system is reachable. If subsequently  $U^T * D = 0$ , the system is deadlock-free.

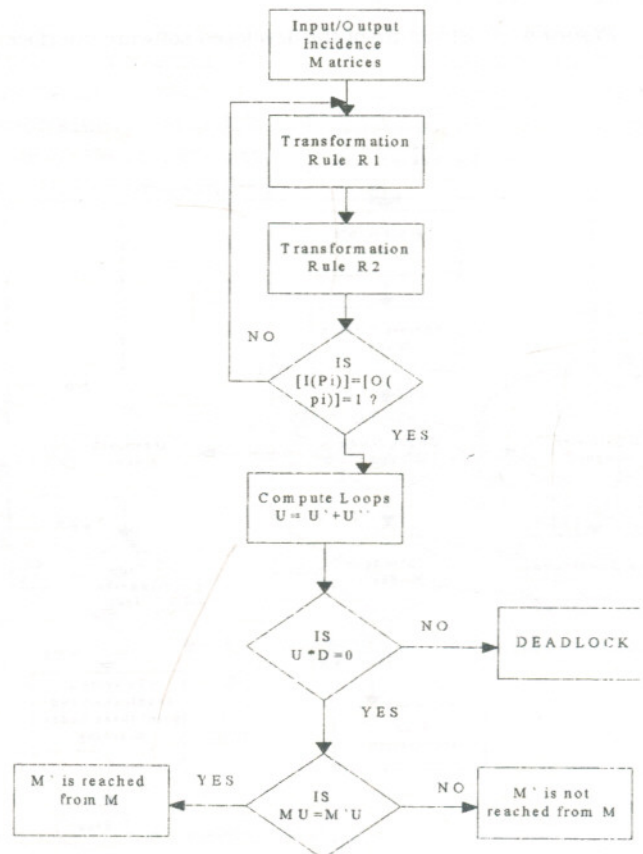


Figure 6 Flow Chart of Computing P-invariant

### CONCLUSIONS AND REMARKS

The high-level system design was developed using SADT method and Design/IDEF software package. A software interface enabling a transformation of static models generated by SADT/ IDEF0 method into dynamic models (PN) through a set of transformation rules was developed. Software used in simulation the manufacturing system behaviour in a real environment according to its functional specification was also presented.

The generic CIM model serves as an example of the capability of IDEF0 as a general purpose functional modelling technique to provide a clear picture of a complex aspect of a manufacturing system or organisation. Ultimately the model could be developed to interface with models of other areas of a manufacturing organisation such as product manufacturing or management to provide a complete generic model of batch manufacturing.

IDEF0 was found to be a powerful descriptive tool that offers a number of features which make it easy to apply and what is more important easy to understand. For example it allows a top down step refinement. Using a graphical representation with few constructs and simple rules, it is based on the widely accepted concept of a function or activity transforming inputs into outputs under external constraints and enables activities to present a diverse group of activities in ordered graphic form. The use of a graphical presentation minimises the necessity for elaborate text and provides clarity in a standard form for the diverse aspects of a manufacturing organisation. Also IDEF0 able to modify the existing system by adding a new constraints (machine, buffer, robot, etc.) without essential changing in the existing model.

Standard PN do not include any time concept. Therefore with standard PN it is possible to describe only the logical structure of systems, and not their time evolution. The introduction of time into standard PN models allows the description of the dynamic behaviour of systems, taking into account both the state of the evolution

and the duration of each action performed by the system. Many of the most interesting parameters of the system design, such as delay and throughput, and in general any time-dependent behaviour, can in this way be modelled. There are, however, different way we can introduce time into standard PN is called Timed Petri Net (TPN). A first possibility consists in associating with each transition. Second possibility consists in associating with each place.

### REFERENCES

1. K. Santarek and I.M. Buseif, "A Structured Approach to FMS Modelling and Design", First International Conference on Mechanical Engineering Advanced Technology for Industrial Production, Assuit University Egypt (1994).
2. K. Santarek and I.M. Buseif, "An Approach to Manufacturing Systems Design Using SADT and PN Tools", Academy of Sciences of the Czech Republic, Prague, pp. 5-7 (1995).
3. K. Santarek and I.M. Buseif, "A Methodology of Advanced manufacturing systems Design and Control by Using SADT and PN Tools, Al-Azhar Engineering Fourth International Conference Cairo, Egypt Pleare, re-organize the references (1995).
4. G.J. Colquhoun R.W. Baines and R. Crossely "A State of the Art Review of IDEF0", International of CIM, Vol. 6, No. 4, (1993).
5. Z. Banaszak K. Jedrzejak, "Rule-Based Knowledge Verification Using Petri Nets", The Third Turkish Symposium on AI and Networks, Ankara, Turkey. (1994).
6. T. Peterson, "Petri Net Theory and the Modelling of Systems", Prentice Hall, (1981).

Received March 29, 1998  
Accepted June 6, 1998

## تصميم منظومات الإنتاج المتكاملة

\*محمد حمدى علوانى و \*\* إبراهيم بوسيف

\*قسم هندسة الانتاج - جامعة الاسكندرية

\*\*قسم الهندسة الصناعية كلية الهندسة - جامعة قار يونس

### ملخص البحث

تقدم هذه الورقة واجهة ربط بين مواصفات المستوى الأعلى لوظائف منظومة إنتاج وسلوك نموذج المنظومة، تعتمد واجهة الربط على عدد من دوال التحويل من مواصفات IDEFO إلى ما يقابلها من Perti Net ، وقد تم تصميم المنظومة في المستوى الأعلى باستخدام طريقة SADT وحزمة برامج تصميم IDEFO ويسمح النموذج المركب من SADT/IDEFO بمعاونة حزم البرامج المصاحبة بتحويل النماذج الإستاتيكية المخلفة بواسطة SADT/IDEFO إلى نموذج ديناميكي Perti Net ويتيح ذلك متابعة أداء المنظومة الإنتاجية بصورة ديناميكية في حالة لتشغيل الفعلى العادى وفي حدود مواصفات أدائها.