Proceedings of GLOGIFT 09
November 12 – 14, 2009
National Institute of Industrial Engineering
Mumbai, India

Conceptual Data Driven Design & Simulation for Flexible & Stochastic Systems

Mohsen Karimi Haghighi
Research Scholar
Vishwakarma Institute of Technology, India
m621.karimi@gmail.com

Dr. S. P. Kallurkar
Professor
Industrial & Production Department,
Vishwakarma Institute of Technology, India

Saghar Izadpanah
Research Scholar
Computer Engineering Department, India
Vishwakarma Institute of Technology, India

Abstract

In this paper, we proposed an approach to simulation modelling and a modelling discipline with conceptual data-driven manufacturing design and simulation system to be use to aid the model development process. The intent of this system is to be able to rapidly create and modify system designs based on the changing manufacturing requirements. It also enables verification of those designs to meet new requirements through simulation. The design and simulation of these manufacturing systems will be based on real data drawn from existing manufacturing applications. There are three major parts of this system. The information management part contains manufacturing applications from which data can be extracted. The scenario creation part contains a scenario management application based on the concepts of function groups and Petri Nets (PN), this application can also generate the simulation model and support files necessary to run the simulation. The final part is the verification and analysis part. It includes a simulation engine that can execute a simulation using the model and data generated by the scenario manager.

Keywords--*Conceptual data-driven design and simulation system, petri net, reference model, Flexible manufacturing System.*

1. Introduction

Modern manufacturing is characterized by high levels of automation and integration, complex interactions among system elements, and high capital costs. While modelling and analysis are important to help ensure good system performance, the integration and complexity of systems often makes purely analytic tools difficult to use.

Hence, simulation remains one of the most widely used tools to fill this need. simulation in general, have experienced great improvements with recent advance in computational technologies. Adopting conceptual data-driven manufacturing design and simulation system implies the ability to reconfigure available manufacturing resources to execute the right production processes to produce any possible product. Because establishing a new production line requires a considerable investment, the existing line has to be able to produce a huge number of different variants, often with unequal capacity requirements. With increased frequency, the production line will need to be reconfigured to keep up with new product designs. The supporting manufacturing system is expected to be flexible enough to be able to adjust manufacturing capability by reconfiguring the system, including the developing and integrating of new functions needed to implement the reconfiguration. The challenge for manufacturers is to design and operate integrated manufacturing systems that can accommodate the accompanying increase in variety and uncertainty. In this paper, we propose a structure approach to simulation modelling, or a modelling discipline with conceptual data-driven manufacturing design and simulation system to be used to aid the model development process.

2. Why Petri Net?

Petri Net (PN) is a methodology that can be used to design discrete-event-system models graphically and mathematically where concurrency, synchronization, and cooperation exist among subsystems. It is also good at describing static and dynamic system characteristic, and system uncertainty.

A simple PN is defined as a bipartite graph consisting of places, transitions, and tokens. Places, or P elements, are defined as resources, which are classified by functions.

Transitions, or T elements, represent the consumption of resources, and the corresponding changes of tokens. Tokens represent factors that affect system state, including raw materials, labour, equipment, data, and information. In addition, a PN may have an associated set of enabling and firing rules to determine under which conditions (particular marking) a transition is enabled and may fire. Some basic definitions of Petri Net are given below [1]:

Definition 1: Petri Net (PN) is a 5-tuple, PN=(P, T, I, O, M_0), where:

- $P=\{ p_1, p_2,., p_n\}$ is a finite set of places represented by circles, $n \geq 0$
- $T=\{ t_1, t_2 , ., t_m\}$ is a finite set of transitions represented by bars or rectangles, in such a way that $P \cap T=\Phi$ and $P \cup T \neq \Phi$, $m \geq 0$
- $I: P \times T \to N$ is the input function that defines directive arcs from places to transitions ($\{N=0,1,2,.\}$)
- $O: T \times P \to N$ is the output function that defines directive arcs from transitions to places. Where ($\{N=0,1,2,.\}$)
- M_0 is the initial mark, $M_0(p)$ indicates the number of token at the initial state.

Definition 2: If the marking $M(p_i)= M_i$, then the number of tokens contained in place pi is M_i .

Definition 3: $I(p_i, t_j)$ indicates the directive arc connection from p_i to t_j. If $I (p_i, t_j)=K$, K is the priority value. The definition of $O (p_i, t_j)$ is similar to $I(p_i, t_j)$.

Definition 4: $PN=(P, T, I, O, M_0)$, If $p_i \in P$ and

$M(p_i) \geq \#(p_i, I(t_j))$, then transition t_j is enabled. Among them, $\#(p_i, I(t_j))$ stands for the priority factor of the transition from p_i to t_j.

Definition 5: When transition t_j is enabled, it is said to be .fired,. and a new mark $M'(P)$ is generated, $M'(P)= M_0(P)+O(p_j, t_j)-I(p_j, t_j)$.

In real-world systems, it is often found that even though many parts or operations are similar, they must be represented by disjoint and identical sub-nets in Petri Net. Coloured Petri Nets provide a more compact representation where individual sub-nets are replaced by one subnet with different kinds of tokens, each token having a colour and representing a different sub-net in the equivalent Petri Net. [2, 3] An example of Petri Net is presented below in Figure 1. The system is composed of computer-controlled machining centres (CNC), auto-guided vehicles (AGV), and buffers. The AGV takes a case from storage, delivers it to the CNC machine, takes the empty case back to storage, then returns to the starting place and waits for new commands.

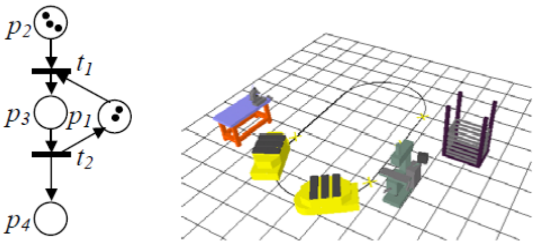


Figure 1: Example of Petri Net

p_1: AGV idle, waiting for task assignment

p_2: Part cases in warehouse waiting to be transport

p_3: AGV taking out parts and transporting them to CNC

p_4: machining process in CNC

t_1: indicating the state before process p_3

t_2: indicating the state after process p_3 and before process p_4

The Petri Net model is described as:

PN=(P, T, I, O, M_0) , where:

$P=\{p_1, p_2, p_3, p_4\}$

$T=\{t_1, t_2\}$

$\Sigma I (p_i, t_i) = \{ I(p_1, t_1), I(p_2, t_1), I(p_3, t_2)\}$

$\Sigma O(p_i, t_i) = \{ O(p_3, t_1), O(p_3, t_2), O(p_4, t_2)\}$

$M0=(2, 3, 0, 0)^T$.

M_0 indicates the initial marking. It means that at the initial state, there are two AGVs and three part cases available in the system. This initial marking state satisfies the fire rule for transition T_1. After transition T_1 is fired, place P_1 is enabled. Then the marking state changes to:

$$M1=(1, 2, 1, 0)^T$$

One issue with using Petri Net to model a manufacturing system is that the difficulty of building and analyzing a PN increases greatly with the complexity of the system being modelled. [4] If a system model is very complex, containing thousands of nodes and transitions, the analysis of this model will be very difficult and time consuming. An approach must be developed where correct PN models of a complex system can be developed and extended from simpler models that are easy to prove valid.

3. A Reference Model of Manufacturing

Our goal is to develop a set of domain-specific simulation modelling tools for manufacturing system. The first step in that process is to specify a reference model for manufacturing upon which the simulation tools are based. A reference model is a standardized representation for a problem domain such as manufacturing [5, 6]. In this section, we summarize a previously developed model used to develop object-oriented simulation modelling tools [7, 8].

3.1. Domain Analysis

In developing the reference model, we have used the process of domain analysis, which is used in software engineering for software modelling of complex systems [9]. In domain analysis, the goal is to organize knowledge about a class of systems or problems (i.e., the domain). Our domain is the set of discrete-parts manufacturing systems, with a primary focus on material flow modelling and control in these systems. Domain analysis is used to classify important manufacturing system elements, their structure, behaviour and inter relationships.

3.2. Reference Model

The resulting reference model classifies manufacturing system entities along two main axes: plant vs. control, and processing vs. transportation. Elements in the plant classification comprise the physical factory, such as machines, material and transporters. Elements in the control classification comprise the logical factory, including decision-makers, performance evaluators and information about the physical factory. Elements in the processing classification focus on the intermediate transformation steps that turn raw materials into finished goods, while elements in the transportation classification address the logistics of moving material through the various process stages. Table 1 shows this classification, and explicitly addresses the importance of material and information transfers/flows.

Important elements in the reference model include the following.

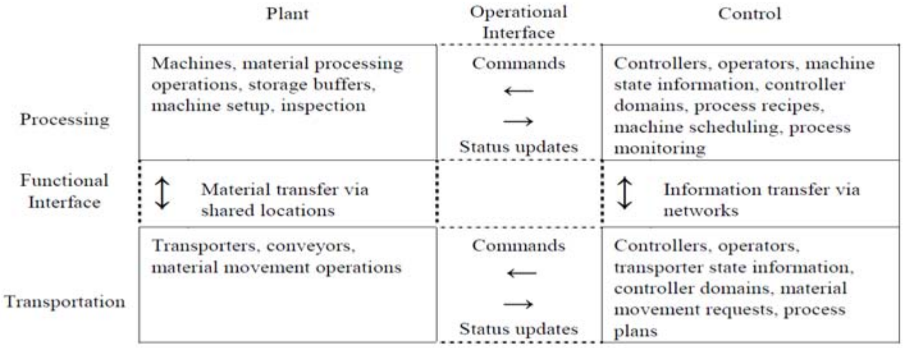
- *Material* is processed as a set of jobs, each of which has a type, a space requirement and a process plan (set/sequence of operations that must be performed to transform a job so that it is finished).

- A *location* is a well-defined place that can hold, transport or process material. It has capacity and behaviour. For example, a machine processing location can be "busy," "idle," or "failed." Its behaviour also includes the set of operations that it can perform.

- The *control system* is increasingly important in today's systems. It is

- The *production system* consists of the set of devices and controllers that transform material. An "operation-location map" specifies which operations can be performed at which processing locations.

- The *transportation system* consists of the set of devices and controllers that move material. A controller's



	Plant	Operational Interface	Control
Processing	Machines, material processing operations, storage buffers, machine setup, inspection	Commands ← → Status updates	Controllers, operators, machine state information, controller domains, process recipes, machine scheduling, process monitoring
Functional Interface	↕ Material transfer via shared locations		↕ Information transfer via networks
Transportation	Transporters, conveyors, material movement operations	Commands ← → Status updates	Controllers, operators, transporter state information, controller domains, material movement requests, process plans

Table 1. Reference model classification

4. Example of a Multi-Robot Flexible Assembly Line and Its Associated Valid P.N. Model

A multi-robot flexible assembly line and its Petri Net model are presented in figures 2, 3 and 4. It is composed of 3 CNC machines (M1, M2, M3), 2 MOTOMAN type robots (R1, R2), composed of a set of controllers that make operational decisions and communicate with one another. Controllers may or may not be organized into a hierarchy. An individual controller has a domain of responsibility (controller domain), which may for example be the set of machines in a cell controller's cell, or may be a device for a device controller. Material is transferred between domains through shared locations (visible to controllers of more than one domain). A controller maintains a representation of each entity with which it communicates (clients)."Domain map" specifies which transporters or transport systems can move material from one location to other parts from Buffer A and Buffer B, and assemble them together with a new part C. If one of these robots is broken, an emergency step is adopted to use the other robot to perform the broken ones work.

Parts to be processed are Part A, Part B and Part C. The system is divided into 3 cells. one PUMA type robot (R3), two SCAR type robots (R4, R5) and two buffers (B1, B2), 3 conveyors and an assembly station(AS).

- Processing cell A: R1, M1, R2 and M2 process Part A. When parts are finished processing, they are stored in Buffer 1

- Processing cell B: R3 and M3 process Part B. When Parts are finished the machining process, they will be stored in Buffer 2

- Assembly cell C: Robot 4 and Robot 5 work cooperatively. They pick up

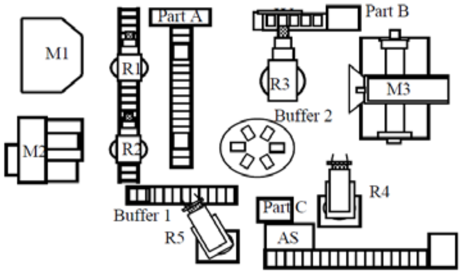


Figure 2: System Layout

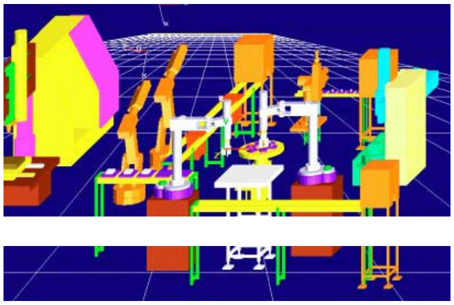


Figure 3: Multi-Robot Flexible Assembly

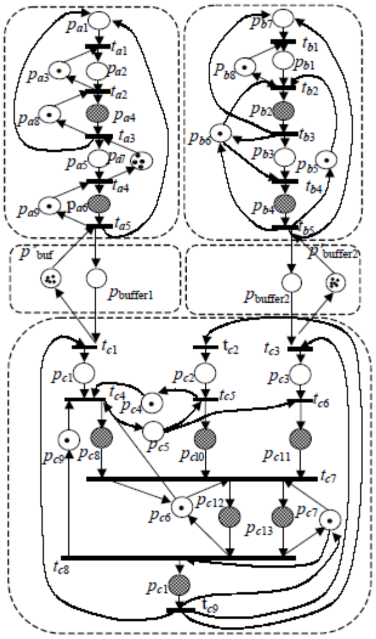


Figure 4: System Valid Petri Net Model

5. Proposed Manufacturing System Design with Data Driven Approaches

Figure 5, a conceptual data-driven manufacturing design and simulation system is presented. The intent of this system is to be able to rapidly create and modify system designs based on the changing manufacturing requirements. It also enables verification of those designs to meet new requirements through simulation. The design and simulation of these manufacturing systems will be based on real data drawn from existing manufacturing applications.

There are three major parts of this system. The information management part contains manufacturing applications from which data can be extracted, in accordance with the Shop Data Information Model, to describe the manufacturing problem to be solved. The scenario creation part contains a scenario management application that can input a Shop Data File and allow a user to define and regroup manufacturing capabilities based on the concepts of function groups and Petri Nets (PN). This application can also generate the simulation model and support files necessary to run the simulation. The final part is the verification and analysis part. It includes a simulation engine that can execute a simulation using the model and data generated by the Scenario Manager.

5.1. Shop Data Information Management

At the bottom of figure 5, several manufacturing applications that have been enhanced with the ability to export and/or import information based on the Shop Data Information Model are depicted. By enhancing these applications in this way, many problems associated with exchanging manufacturing information can be mitigated. This will reduce the time and complexity included in developing integrated applications suites to manage manufacturing operations and to design new manufacturing capabilities.

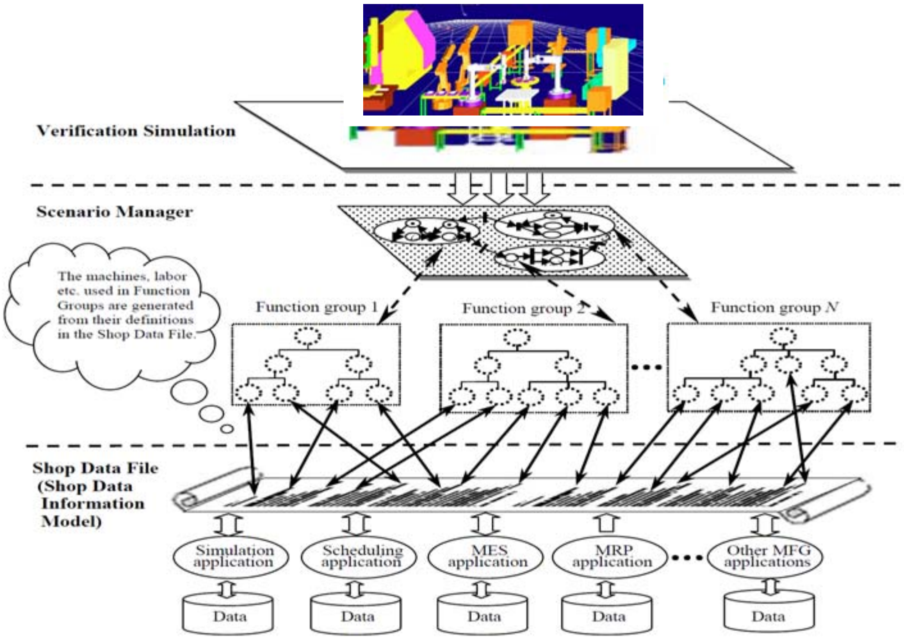


Figure 5: Conceptual Data Driven Design And Simulation System

To support the Scenario Manager, data describing potential manufacturing system layout designs will be extracted from the existing applications (such as simulation application, scheduling application, manufacturing execution systems (MES) application, and manufacturing resource planning (MRP) application, etc.) and encoded in the Shop Data File. By enabling data-driven simulation in this way, many more layout design scenarios can be examined than would otherwise be possible.

5.2. Scenario Manager

The Scenario Manager manages function groups. Each function group represents a typical manufacturing capability, e.g., fabricating a specific component, a unique welding method, or a testing operation [11]. The machines, labour, tools, etc. used in a function group are generated from their definitions in the Shop Data File, which is an input to the Scenario Manager. The more Diversity among the function groups associated with a system design, the more capability the system possesses to satisfy different manufacturing requirements. Function groups can be combined or reconfigured by the Scenario Manager to form new manufacturing capabilities according to the requirements.

The establishment of a dynamic network among the function groups of a system is critical to support the ability to rapidly reconfigure the system. The goal is to quickly identify required changes and use the changed data to support model reconfiguration. In this system, the valid Petri Net methodology is employed to reach this goal.

One issue with using a Petri Net methodology is that it is hard to verify the validity of the PN model, especially when the model is very complex [12]. The validity of a PN model is defined by three properties:

- *Bounded* indicates the absence of overflow in the system model. This characteristic allows for the specification of a limit on the number of tokens that may be in a place at any time.
- *Live* implies that there is no possibility of deadlock.
- *Reversible* indicates that the system can return to its initial state from any current state. This characteristic is very important for error recovery.

To be valid means the model developed is reliable, without overflow, deadlock or conflicts. The approach used here is to develop valid PN models of a complex system by extending simpler valid models whose validity is easy to prove [13]. This approach uses the PN valid extension theorem. This theorem and its proof can be seen in [14,15]. It states that, if valid PN models are combined in accordance with the connection rules that are defined in the theorem, then the entire combined PN model is valid.

The model developed is flexible enough to handle problems of dynamically inserted schedules, system changes, and other unpredictable situations. It offers not only a means to model discrete-event systems graphically and mathematically where concurrency, synchronization and co-operation exist among subsystems, but also can be easily converted into computer control code for manufacturing processes control. The information from the Shop Data File and

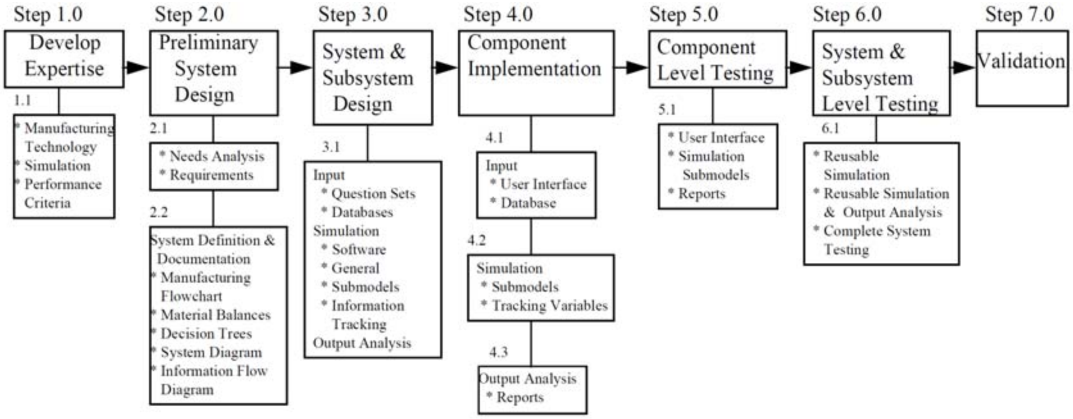


Figure 6: Flow Diagram of Methodology for Designing and Developing a Discrete Event Simulation

model and support files necessary to run the simulation Based on an analysis of the output of the simulation, the Scenario Manager can be used to modify the simulation scenario and generate a new simulation model and support files. These new files can then be used for a new simulation execution. In this way, simulation studies can be done using a data-driven approach without making repeated, tedious modifications to the simulation for each simulation replication. As requirements change or if different analysis results are desired, the simulation model can be quickly modified to perform analysis according to the new requirements.

6. Methodology

The simulation system being developed can be characterized as being narrower (and more detailed) in application, but broader in scope. First, this simulation system is designed to capture knowledge of a specific manufacturing technology and utilize this knowledge to help design engineers model and compare individual manufacturing options. Therefore, the knowledge on which to base the simulation is deeper than for general-purpose simulations. The knowledge required for simulation includes an understanding of all the ways that the manufacturing process could be configured, which possibilities are most often utilized, which are unlikely, and what variables effect the layout. Although an understanding of the data requirements is important, very little data is required to design a simulation.

The second difference between this simulation system and traditional simulation model development is the integration of environmental, quality, and cost criteria into the model. Because of this, a thorough understanding of possible material flows, both in and out, is required for each process step and the overall process. This is one of the most difficult aspects of designing a simulation of this type. Depending on the way the manufacturing process is laid out, scrap and waste products as well as good product will be composed of varying amounts of materials. A thorough understanding of what parameters effect the composition of the input, output, and recycle streams is required so that during the design of the simulation, appropriate variables will be put in place to keep track of each material. A flowchart of the development methodology is shown in Figure 6. number of steps that the analyst must address to build a simulation structured to a specific area of application. In step 2, a needs analysis is conducted to determine the systems requirements in terms of the manufacturing process flowchart and the ways in which they can be assembled together, material balances around each process and throughout the overall process, and the decision steps needed to decide what elements and process structure is necessary based upon user inputs. In step 3, the items developed in step 2 are used in the system design. Among the necessary items are sets of questions that the user will need to address for parameter input and to guide the appropriate configuration of the simulation, together with an appropriate database repository for knowledge and user input data. From these software to elicit input, questions that the user will need to address for parameter input and to guide the appropriate designed. These, finally, can provide output statistics generated during the simulation itself and presented to the user in output reports covering material flows, costs, and production throughputs. In steps 4 through 7 these designs are implemented, tested, and validated.

The process depicted in Figure 6 can be considered the preliminary stages for a class of simulation models, and thus it is not surprising that the process is parallel to that required in any simulation study. That is, the stages are a subset of those given in [10] for the design of simulation studies. The design of this simulation is to build once and can be reused in specific applications. It also means that high fidelity simulations can be produced by users without having to program (or master) all details of the simulation.

7. Conclusion

In this paper, based on the Petri Net theory, an efficient valid Petri Net manufacturing system modelling methodology in Conceptual data-driven manufacturing design and simulation system is developed. The model developed is flexible enough to handle problems of dynamically inserted schedules, system changes, multi-robot co-operation control and the handling of unpredictable cases . It offers not only a means to model discrete-event systems graphically and mathematically where concurrency, synchronization and co-operation exist among subsystems, but also can be easily converted into computer control code and interfaced to practical manufacturing processes . it has enough flexibility when requirements change, the simulation model can be quickly modified to perform analysis according to the new scenario. Potential manufacturing process and factor layouts can then be planned and optimized.

References

[1]Yuan Chongyi. 1998. Petri nets theory. Machinery Press.

[2]Kasturia E., Dicesare F., Desrochers A. A. 1988. Real-time control of multi-level of manufacturing systems using coloured petri nets. IEEE Int. Conf. Robotics and Automation, Raleigh,

[3]Elkoutbi Mohammed, Keller Rudolf K.. 1998. Modelling interactive systems with hierarchical coloured Petri nets. Available online via http://citeseer.nj.nec.com/elkoutbi98modeling.html [accessed Feb. 16, 2002].

[4]Lee K. H., Favrel J., Baptiste P.. 1987. Generalized Petri net reduction method. IEEE Trans. Syst. Man Cybern 17(2)

[5]Biemens, F. P., and Vissers, C. A., 1989, "Reference Model for Manufacturing Planning and Control," Journal of Manufacturing Systems, 8(1), 35-46.

[6]Van Haren, C. R., and Williams, T. J., 1990, "A Reference Model for Computer Integrated Manufacturing from the Viewpoint of Industrial Automation," in Proceedings of CIMCON '90, Gaithersberg, MD, 42-62.

[7]Bodner, D. A., 1996, Real-Time Control Approaches to Deadlock Management in Automated Manufacturing Systems, Ph.D. dissertation, School of Industrial & Systems Eng., Georgia Institute of Technology, Atlanta, GA.

[8]Narayanan, S., Bodner, D. A., Sreekanth, U., Govindaraj, T., McGinnis, L. F., and Mitchell, C. M., 1994, "Modeling Control Decisions in Manufacturing Systems Simulation Using Objects," in Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, 1392-1397.

[9]Arango, G., and Prieto-Diaz, R., 1991, "Domain Analysis Concepts and Research Directions," in Domain Analysis and Software Systems Modelling, Prieto-Diaz, R., and Arango, G. (eds.), IEEE Computer Society, Los Alamitos, CA, 9-33.

[10]Law, A.M., and W.D. Kelton, 1991 Simulation Modelling and Analysis, 2ed., McGraw-Hill, New York, NY.

[11]Gilmore, J. H., and B. J. Pine. 2000. Markets of One Creating Customer-unique Value Through Mass Customization. A Harvard Business Review Book. Harvard Business School Press.

[12]Jiang, Z., R. Y. K Fung, and M. J. Zou. 2003. Automatic Modelling of One-of-a-Kind Production Systems by Temporized Object-Oriented Petri Nets with Changeable Structure (TOPNs-CS). International Journal of Advanced Manufacturing Technology, Vol. 21, 45-65.

[13]Qiao, G., C. McLean, and F. Riddick. 2002. Simulation System Modelling for Mass Customization Manufacturing. In Proceedings of the 2002 Winter Simulation Conference, ed. E. Yucesan, C. Chen, J. L. Snowdon, and J. M. Charnes, 1108-1116. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

[14]Zhou, M. C., and F. DiCesare. 1991. Parallel And Sequential Mutual Exclusions For Petri Net Modelling Of Manufacturing Systems With Shared Resources. IEEE Trans. on Robotics and Automation. Vol. 7, No. 4, 515-527.

[15]Wu, N. and M. C. Zhou. 2001. Avoiding Deadlock and Reducing Starvation and Blocking in Automated Manufacturing Systems. IEEE Trans. on Robotics and Automation. Vol. 17, No. 5, 65-72.