

# MIMCA: A Modelling and Simulation Approach in Support of the Design and Construction of Manufacturing Control Systems Using Modular Petri net

S. Ariffin, K. Hasnan, R.H. Weston

**Abstract**—A new generation of manufacturing machines so-called MIMCA (modular and integrated machine control architecture) capable of handling much increased complexity in manufacturing control-systems is presented. Requirement for more flexible and effective control systems for manufacturing machine systems is investigated and dimensioned-which highlights a need for improved means of coordinating and monitoring production machinery and equipment used to- transport material. The MIMCA supports simulation based on machine modeling, was conceived by the authors to address the issues. Essentially MIMCA comprises an organized unification of selected architectural frameworks and modeling methods, which include: NISTRCS, UMC and Colored Timed Petri nets (CTPN). The unification has been achieved; to support the design and construction of hierarchical and distributed machine control which realized the concurrent operation of reusable and distributed machine control components; ability to handle growing complexity; and support requirements for real- time control systems. Thus MIMCA enables mapping between 'what a machine should do' and 'how the machine does it' in a well-defined but flexible way designed to facilitate reconfiguration of machine systems.

**Keywords**—Machine control, architectures, Petri nets, modularity, modeling, simulation.

## I. INTRODUCTION

**I**N the present competitive for product in world market, high level of flexibility is required for automated manufacturing systems in order to be more responsive to changing needs of different product types from markets at a considerable fast production time. However the greater flexibility means a greater complexity in the processes of manufacturing machines. The configuration of manufacturing machines are normally distinguished by the use of computers and must be capable of handling multiple jobs and parts by performing a number of different operations, Hence a new generation of machine control system is required capable of handling much increased complexity, including means of handling primary issues of concurrency and reconfigurability.

In a well designed manufacturing workstation, individual machines and control elements are combined and require coordination to achieve a high production rate, high machine utilization minimum work-in-process inventory. Often such manufacturing machines are required to demonstrate flexibility and reconfigurability. However, current approaches to the design of manufacturing machine systems require extensive human resources in terms of time, cost and expertise, thus preventing the full achievement of potential automation benefits toward improving the manufacturing operations. For example, one of the key factors limiting the use of robotic systems as an integral part of a manufacturing workcell stems from constraints imposed by their control systems. Indeed many forms of manufacturing machine (including robots) are supplied with low-capability and simple controllers typically being restricted to position control and simple analogue and digital input/output capabilities. Their restrictive computational *architecture* does not allow the implementation, of flexible machine control Strategies nor does it facilitate their flexible and effective integration into a host, environment. To achieve such an integration in an effective way, it is necessary to place the operation or behaviour of a machine into the context of the system in which it operates.

A manufacturing system is usually organised as a collection of manufacturing pells which commonly are interconnected by material handling systems. A typical manufacturing cell comprises a small number of workstations which in turn comprise machines (including robots). Often such systems are organised in a *hierarchical* and *distributed* fashion, thereby naturally providing well defined structures, which essentially modularise solutions and make them easier to manage and control. Such an organisation can also structure the distribution of control functionality to locations which are appropriate to the operations performed by a particular machine. However, specific attributes of such structure need to accommodate different application requirements, different types of machine and their different classes of elemental building blocks.

Prof. Ir. Dr. S. Ariffin and Dr. K. Hasnan are both with the Universiti Tun Husein Onn Malaysia (corresponding author to provide phone: 605-4537704; fax: 607-4536080; e-mail: sapar@uthm.edu.my, khalid@uthm.edu.my).

R. H. Weston was with Loughborough University, United Kingdom (e-mail: r.h.weston@lboro).

## II. SIMULATION MODELLING IN MACHINE CONTROL SYSTEMS

From a survey of modelling methods (Ariffin 1996) the author compared their capabilities to meet the research objectives as depicted in Table I. This analysis led the author to conclude that *Petri nets (PN)* and *object-oriented (OO)* modelling are appropriate methods for modelling machine control systems. Petri nets are outstanding in terms of their support for concurrency, mathematical representation of physical machine behaviour (with respect to time), multiple events and interfacing among modules of real-time system; whereas the object-oriented methodology is particularly suited for control flow and message-passing among distinct physical machine components and also readily supports reusability and scalability. Furthermore, both methods are well-known, can be hierarchically structured, graphically represented, support data interoperability and are suitable for real-time control systems. This led the first author to develop simulation techniques based on Petri nets and object oriented approaches, which facilitate manufacturing systems design and evaluation.

On account of the complexity and flexibility with which components need to interact in machine control systems, it is difficult to define simple input/output relationships when designing them. A software tool is required which can be used successfully to support this kind of task based on simulation. *Simulation* promises in the near future to be an effective technique ([9], [49]) for handling complex system analysis. Simulation techniques can be used to shorten the time into market of products and to reduce their manufacturing costs. TWs can also help make decisions about improving manufacturing operations [22]

TABLE I  
COMPARISON OF MODELING METHOD FEATURES OF MACHINE CONTROL SYSTEMS

	Funct. Decomp -SADT, MAS- COT3 IDEFo	Data Flow SSADM YOUR- DON	Object- Oriented HOOD OPnets IOPN	DARTS, COSL, JSD	Formal Method Petri nets	Formal Method GRAF- CET	Formal Method Estelle	Formal Method ADA OC- CAM
Data Interoperability								
Mathematical based technique								
Hierarchical Structure								
Real-time control requirement								
Repr. dynamic behaviour								
Graphical representation								
World-wide/well-known method								

Simulation has also proved to be a reliable and powerful tool in the design, development evaluation, optimization and operation of manufacturing systems ([13], [23]). In this context simulation can be defined as the process of examining a production system (or production process) by using a model [45]. Potentially users who have limited expertise can use simulation software effectively.

When a model is constructed it should have an inherent ability to support the analysis of the behaviour of a system under a variety of operating conditions. Based on this model it should be possible to examine and optimize real processes and the behaviour of real systems. Furthermore, as jobs run concurrently there is need to share resources whilst considering possible deadlock and conflict conditions. As the complexity of a system increases, it becomes increasingly difficult to build usable and meaningful models. Heidelberger [24] argues that simulation, because of its versatility, often becomes the most viable analysis technique. Simulation runs can be made to estimate the performance of the modelled system, and changes in configuration may be suggested by the simulation output. Thus a simulation tool can enable a designer to test ideas before committing resources to build a new facility, revamp an existing facility or change a manufacturing operation. Any simulation method chosen could adequately represent the activity of machines and facilitate analysis of performance.

However, key problems here are how to represent concurrent behaviour and how sufficient flexibility can be provided by the simulation modelling method. Furthermore, there are outstanding research questions about how simulation can be used as a real-time decision and control tool for manufacturing machines. In particular effort needs to be placed on establishing generic links between simulation and physical machine environments, so that simulation logic can properly reflect the current system status. Harmorosky [19] highlighted a set of relevant implementation issues which need to be addressed, namely: *how to establish a more formal model structure to facilitate easy inter facing with physical machine systems; and how to derive a mechanism for retrieving system status data from p hysical systems for use in the simulation environment.*

## III. ANALYSIS AND SELECTION OF EXISTING ARCHITECTURES

Among current methodologies includes: **hierarchical control systems** ([2], [26] [27], [29], [36]). Generally hierarchical control architecture comprises some levels in a control hierarchy; facility, shop, cell, workstation and equipment in a well-defined and limited complexity at each level. A Real-Time Control System (RCS) architecture is designed and developed at NIST ([3], [5]) and has been used for the designed and implementation including: robotic crane [12], cold mine automation [25] and machining workstations [5]. A similar architecture described by Jeffrey [26] and Joshi [29], where at each level the same three classes of function is performed, namely: planning, scheduling and control. Whereas Pritschow [36] has described rules for constructing complex automation systems, in the form of organised factory, supported by a service hierarchy.

An **open system architecture** is another method which address and adopt use of modular, open control concepts to incorporate control functions of machines or to readily modify functions to suit. the specific needs [36]. Some emphasis

which was placed on developing CIM technology ([20], [44], [40], [18]) has allowed more configurable and open system to be developed during the 1990s to realise flexible integration and flexible automation. These includes; UMC (Universal Machine Control) and IMDC (Integrated Machine Design and Control) approach which has been developed by Weston [43], Harrison [20] and Wright [46]; CIMOSA (Open System Architecture for CIM) which was developed by an ESPRIT Consortium [17] and MOSAIC (Machine too Open System Architecture Intelligent Control) which was developed at New York University and recently MOSAIC-PM (Machine tool Open System Advanced Intelligent Controller for Precision Machining which is developed at University of California [47]. UMC is an open approach to machine control which defines methods and provides software tools which enable software reuse and interoperability between different classes of machine component and device ([14], [42]). The CIMOSA approach to open systems [28] was developed to handle issue related to the management of change, through facilitating flexible integration between enterprise operations. Another development so-called ARTICS (a reference model open system architecture for Real-time Intelligent Control Systems) based on NIST RCS [4] has a means of accelerating technological development in automation and robotics.

Another important software architecture uses *object-oriented (OO) approach* [11]. For example, the Robot Independent Programming Environment (RIPE) was developed at Sandia National Laboratories [34] is largely based on the OO approach. The development environment for RIPE has- the following primary layers: task level programming, supervisory control, real-time control and device drivers. The approach of NIST , RCS can be viewed as. an OO method [38], whereby the objects used to derive the design (in a bottom-up fashion) are sensor, actuator and controller modules. The sensor processing module (object) hierarchy needs to support the NIST RCS method and can benefit from the OO analysis approach. The world model has the responsibility to bridge the gap between the sensor processing (OO form) view the world and the task decomposition (control flow) view where actions are performed.

Based on attributes identified from some other architectures from literatures, the first author compared important features of machine control architectures, as depicted in Table II. These architectures include: MOSAIC (Modular Open System Architecture for Industrial Motion Control) developed in Germany [39]; ROBLINE for workcell design and control systems which was developed by CMIETRIX, USA [41]; NGC (Next Generation Controller) developed at the National Centre of Manufacturing Science (NCMS), USA [48]; and OSACA (Open System Architecture for Controls within Automation Systems) project described by Pritschow [37]. Whereas the various features include: the type of control hierarchy used; support for real-time control; the generality of solutions; types of modelling technique; the user friendliness of man-machine interfaces; the current implemented status of

hardware and software; and the extent to which the design method is known. Through this comparison NJST RCS, UMC and CIMOSA seem to be outstanding candidates which collectively promise to meet requirements for machine control.

The first author's investigation of available control architectures led to the conclusion that no single available architectural model can support all of the essential characteristics listed above. However, the author did conclude that it is likely that a combination of available architectural frameworks and modelling methods could meet the defined need. In particular the author concluded that the combined use of NIST RCS, UMC, CIMOSA and various types, of Petri nets model could provide a significant advance on 'current' practice. Collectively this combination of architectures and methods should be capable of supporting complex machine operations and mapping the behaviour of machine systems, generated with a design environment, onto real machine systems. Hence modelling result was given as realising a federation of architectures and methods, choosing the best features from available architectures to achieve a more 'open' and more 'formal' approach to design and construction of machine control systems. However, a major difficulty arises conforming their ability to represent the behaviour of machining operations in a manner which promotes modularity, reconfigurability, integratability, reusability, scalability and so on as required by modern manufacturing environment.

TABLE II  
COMPARISON OF MACHINE CONTROL FEATURES

	NIST RCS ARTICS	UMC	MO- SAIC (EU- ROPE)	ROB- LINE	RIPE	NGC	MO- SAIC (U.S.)	OSACA / ESPRIT	CIM- OSA/ ESPRIT
Reference Architecture	generalised open hierar.	support various architecture	Motion control: closed	Robot specific: closed	Robot specific: closed	Mach. tools specific: closed	M/c tool specific: closed	M/c tool specific: closed	User independent
Real-time Control and Sensor-based system									User independent
Diversity of application areas	diverse machine	diverse machine	AGV, specific m.c.	Robot cells	Robots	Machine tools	Mach. tool specific	Mach. tool and	Diverse application
Modelling techniques	object-oriented funct. decomp.	object-oriented funct. decomp.	not known	not known	Object-oriented	not known	Object-oriented	function alities Object-oriented / data flow	User independent
Graphical Simulation Environment									User independent
Current implemented hardware and software	Motorola VME/ Vxwork C, C++/ SUN/ UNIX	Motorola VME/G 64-/ OS-9/C/ SUN/ PC/ UNIX, DOS	no current impl.	Intell/ VENV, C/ IBM WS/ VENIX	VME/ VxWork & C, C++ UNIX	information not avail.	Motorola, MVME/ SAGE, VxWork / SUN WS/ UNIX, C.	information not avail.	User independent
Well-known method and implementation									

The use of Petri nets for modelling the structural and the behavioural properties of complex manufacturing operations was also of interest to the first author. This is because the first

author is of the opinion that the creation of a Petri net model representing manufacturing operations is equivalent to planning a sequence of job operations, and hence repeatedly firing the transitions of this net is equivalent to scheduling the execution of a batch of jobs whenever the execution of operations becomes feasible. Monitoring can be performed by checking, prior to Petri nets execution, to see if it is feasible to execute these operations; a fault would be detected whenever the feasibility cannot be established.

#### IV. SYNCHRONIZATION OF A MULTI-MACHINE SYSTEM

The task performed by a machine system is the result of planned actions and the input of data. When a number of machines (such as a number of robots) is required to cooperate within a manufacturing cell, complex situations can arise. In some respects (as determined by characteristics of the application concerned) the task performed by each machine will be dependent upon those performed by the other machine(s). Such a multi-machine system can be viewed as a distributed mechanical mechanism, requiring synchronism between concurrently operating mechanical mechanisms, which in turn requires computational mechanisms to model and synchronize the distributed and concurrent activities involved. The real power of a concurrent modelling technique lies in its ability to detect deadlocks, conflict and other properties of interest in discrete-concurrent systems. Due to their inherent complexity, it is appropriate to model such systems for the purposes of verification, analysis and simulation. The introduction of time precedence in relation to the execution of these several processes (i.e. synchronization) is a crucial issue in concurrent logic programming [10]. It provides a means of restricting the freedom of scheduling and communication among active processes, and synchronizes the computation with external events. Furthermore, manufacturing machines normally incorporate a multitasking, high-speed manipulation capability. In addition they are often required to demonstrate flexibility (in term of their ability to demonstrate programmable operating characteristics) and reconfigurability (in respect of being capable of being restructured to meet different application requirements).

One way of meeting the synchronism requirements is to create modular machines which operate concurrently and can be readily programmed and reconfigured to suit changing needs. Often such systems may need to be physically distributed with local processors assigned to each module. However in such systems there may be complex communication problems among local machine control components. Thus, it is important that machines (including robots) operate as a distributed system whereby they can work co-operatively with each other within a cell or with other cells in a factory.

Hence future machine control systems should embody adequate real-time machine control with modularity and integration capabilities -which facilitate time-critical communication among machine control devices, where those

devices will often be supplied from a variety of sources. Furthermore, if both the mechanical and control system aspects of modular machine systems can be specified and constructed easily such systems will allow the creation of a new generation of machine and production systems, which enable agile manufacturing that is oriented toward consumer customization.

#### V. MAPPING BETWEEN MACHINE MODELS AND PHYSICAL MACHINES

In practice it proves to be very difficult and time consuming to optimise the configuration of a manufacturing cell once such a system has been installed and programmed (Angermuller and Niedemayr 1989). Whereas potentially use of an appropriate simulation technique should establish a configuration which is better designed and has less inherent faults (e.g. no deadlock and conflict in the real system). Thus it was understood by the author that the MIMCA simulation model-building capability should map as closely as possible onto any chosen physical layout within a manufacturing machine system, as illustrated in Fig. 1.

The aim here is to overcome difficulty in representing the complexity of the system especially when dealing with various, issues, such as machine layout, real-time control of operations and multiple part flows. The results generated from a modelling and layout design exercise were expected to be useful because such a layout should handle a variety of different parts involving various motion sequences and their co-ordination. It was also understood that various machine application areas could be investigated using such a simulation system. The result of this effort should be very useful either in terms of verifying the concept, seeking to optimise and develop alternative cell layouts or convincing management via visual demonstration.

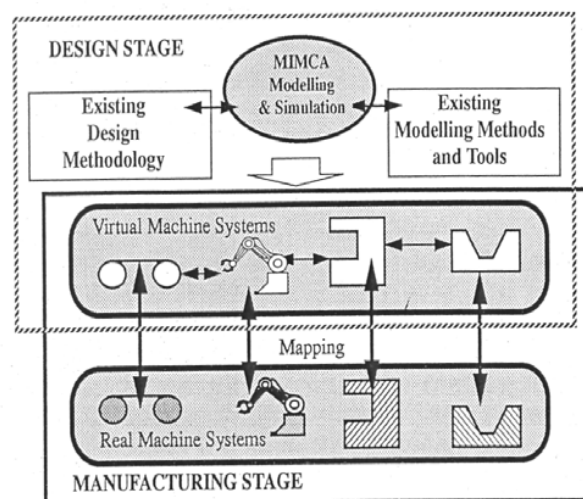


Fig. 1 Modeling Structure of the MIMCA Approach

Control modules which correspond to physical manufacturing machines, can have a corresponding task or control program in the virtual (or simulation) world, use of which can progress the design process. It is important that virtual manufacturing machines and processes created should have a close correspondence to actual manufacturing machines and processes. By simulating operational sequences and layout models on a computer system, the performance (including throughput time, machine utilization) can also be simulated.

Thus a new simulation modelling method would need to support the design of the physical machine layout and thereby facilitate the optimal location of materials handling systems. This phenomena is important to minimize the material handling cost and time of operation, and integrate manufacturing and storage activities, particularly in FMS ([21],[22])

The author believes that significant benefit could be realised by enabling mapping between modelled machines and real machines as this can lead to an environment for the rapid prototyping of machine systems. The resulting potential benefits are developed within this paper, as are means by which such a mapping can be established.

## VI. CONSTRUCTION OF MIMCA MODELLING AND SIMULATION ENVIRONMENT

### A. General Construction

Modelling and simulation offers possible means of optimising the way in which the coordination of tasks can be achieved. Theoretically it can play an important part in the process of designing both hierarchical and non-hierarchical models of real-time control systems ([15],[37]). Potentially this can allow experiments to be conducted with the purpose of either understanding the behaviour of machines or of evaluating various control topologies, control strategies and application logic required within a system.

Research into the modelling and simulation in modular machine systems presents real-time control problems and can lead to solutions of a centralised or decentralised nature, comprising a variety of processing modules. Potentially these solutions offer suitable means of optimising the design of flexible manufacturing facilities. The adoption of control software which is modular in construction can lead to easy modification, hardware flexibility, high-speed (through enabling parallelism), accuracy and low cost automation. A great deal of time in programming, testing and reprogramming can be eliminated by this methodology. Thus, research in this arena can seek ways of increasing the overall efficiency and productivity of automated manufacturing systems.

Based on these requirements, this research seeks to federate the use of a number of complementary architectures and methods to facilitate the development of models of complex machines which are deadlock and conflict free, and to readily support mapping between machine models and real machines in a scalable manner. Fig. 3 presented a modelling structure proposed by the author for the design and control environment

of the federated approach. This was conceived to realise the required mapping between modelled systems and real systems. This environment is referred in this paper as MIMCA, a "Modular and Integrated Machine Control Architecture".

Thus MIMCA was designed to federate 'best' features from existing control architectures and thereby provide a suitable 'workbench' for producing machine control solutions in a more 'open' and 'formalised' way. Use of MIMCA should result in the design and construction of well structured and scalable machines. Here, the MIMCA modelling and simulation environment would be expected to offer advantages of quick reconfiguration, modelling flexibility and reusability of modelling elements. A key paradigm of the solution will be functional decomposition into modules, whereby reuse and sharing of goals, commands and information will be achieved by model driven software modules. At a task level, modularisation of machine control functions, should allow individual motion control functions to be arranged optimally, and assigned appropriate application logic and operating conditions. Machine characteristics will be defined by coordinating module functions involving interfacing, actuation and sensing. At the device level, a distributed machine system will be developed in which mechanisms (which may be complex in nature, such as one or more robot arms) change their relative position with respect to products, tools, fixtures etc. which comprise their work environment.

Generally, the MIMCA environment was expected to emerge from an analysis and integration of previously established architectures and methods, namely: NIST RCS ([2],[5], Petri nets ([35], [49]), UMC ([20], [47] [14]) and CIMOSA [17]. Unification between the NIST RCS and UMC was expected to offer a generally applicable and usable hierarchical control system architecture, supporting functional decomposition and sensor processing in manufacturing systems, including machine systems. This unification has the potential to provide a software architecture for machine control which is flexible, reusable and easily modified as required. Hence MIMCA would need to support control systems and mechanisms, involving sensory input which is constantly monitored (based on the NIST RCS); thereby to determine events occurring in the environment so that they can be acted on. In this respect MIMCA aims to support the deployment of a hierarchical and distributed control structure for machine systems by utilizing open, machine configuration facilities of UMC machine systems (which were already available in the MSI Research Institute at Loughborough University). A Petri-net based software tool would be used to determine the basis of manufacturing operations design in order to provide design specifications, analysis and simulation. Such architectural framework can be suitably structured using the formalism provided - by CIMOSA. This construction would then result in the so-called the MIMCA simulation modelling tool which would be expected to provide a "workbench" which could have an application base in designing machine control systems.



### B. A Task Level Architecture (MA) using Petri nets

The above unification forms a new multilevel control architecture which so-called Task Level Architecture (TLA). The TLA proposed organises the control modules so as to create the functional and distributed relationships and information flows over a hierarchical and modular computing structure as illustrated in Fig. 2, this essentially being based on the reference model proposed by Albus [1]. The architecture will permit modification of the machine control environment, which embodies a multiple sensing-recognition capability.

During operation, relationships between modules within and between level of hierarchy can be reconfigured for accomplishing different goal and task requirements. Each module will be assigned a responsibility to achieve particular objectives. Having abstracted the functional requirements of a module this leads to a definition of implementation requirements pertaining to that function (e.g. the real-time control) and of the interfaces between each module within the overall configuration of the control systems. An interface specification defines both the type of data and kind of action which needs to be taken by an instance of a module, as all communications with. The module instance must be through its interface. The type of inter-process communication mechanisms used between modules and the types of responsibilities assigned to different modules will have a significant impact on the way in which the system interacts with its environment and hence on its performance.

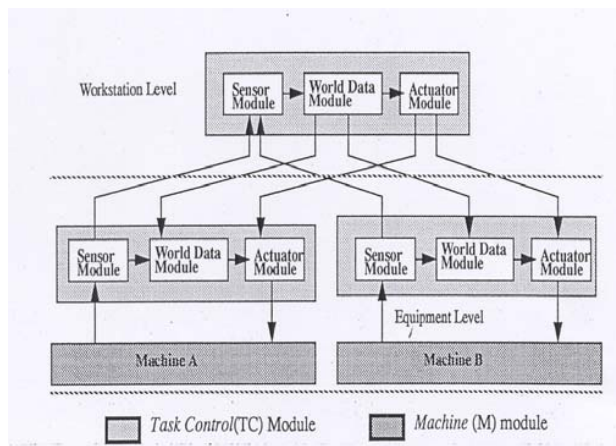


Fig. 2 Task Decomposition at the Workstation Level (based on the NIST RCS Reference Model)

This functional decomposition shown in Fig. 2 can be carried out at many levels of detail which are represented by a breakdown into *task control (TC)* modules. Practically, the lowest level of tasks must be executed by a single resource. The execution of such tasks is defined by a so-called *machine* (" module which is to be linked with hardware components. Each TC module is horizontally partitioned and comprises of a sensor module (SM), a world data module (WDM) and a actuator module (AM). The AM receives job commands from

the relevant workstation's TC module. The SM sensor data from a relevant machine systems and integrates the sensory systems of a machine in a generalized manner. The AM handles the problems of task monitoring, control of machines and interfaces with the operator. A five-stage block model presented by Levis [33] is incorporated into each TC module of the proposed TLA model (see Fig. 3). Thus each TC module, at different levels of the TLA, implements the modular structure of the PN model and in this way realizes communications among the modules in manner which in principle can be extended with relative ease and as such can provide scalable solutions.

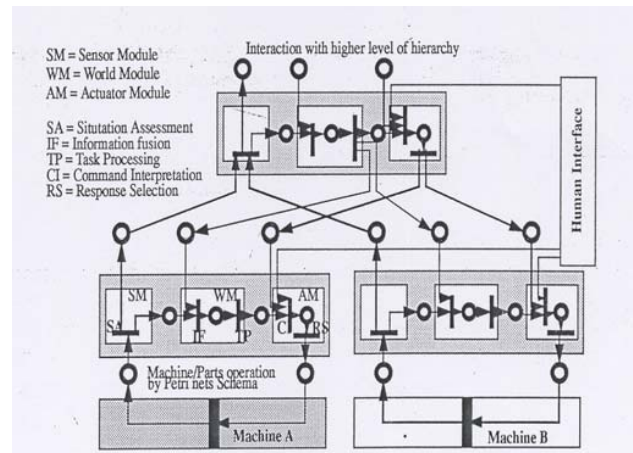


Fig. 3 The Modular Petri nets Interaction between Tasks at different Levels

### C. Modular Petri Nets (MPN) Modeling

Petri nets (PN) model provides the characteristics of graphical representation and mathematical analysis of deadlock, liveness, reachability and boundedness of control logic in machine systems ([16], [31],[32]). Based on a stochastic Petri net modeling approach it is possible to examine and optimize real processes and behaviour of real systems [50]. Using a PN approach, a model issued to describe time dependent processes and interactions within the real system. By analyzing timing relationships using a PN model, the sequential behaviour of test can be optimized by allocating appropriated physical components processes, jobs, operations and activities and at the same time avoiding deadlocks between processes. Thus, PN can be used to help understand and analyse job flows and involve complex interactions between a varieties of functions. In other words, it is an open system research question as to how is PN based modeling, simulation and analysis should appropriately be achieved as system grow in complexity.

Further, a control software tool can be developed to support PN based specification techniques. The output from such a tool can display and verify the correctness of connections between places and transitions, and the time-stamp of each place. The movement of tokens and the firing sequences can

be displayed to simulate and visualize conditions that will occur in a machine environment. Here control logic is required to drive the concurrent operation of PN, including the synchronization protocols required where PN are used to model distributed processes. Thus, the use of PN models to graphically observe interaction between concurrent and synchronous activities would require the development of customized software.

Once a model of machine system has been created it may be run within a simulation environment. Task behaviour may be emulated by the generation of events for each node of the model. For example when modeling a flexible manufacturing system, a node may represent an assembly or buffer station. This node is characterized by a list of properties, including a time stamp. To more closely model to the way that process dynamics occur in real systems, the simulation environment should allow several events to occur in parallel. In other words, several transitions should be allowed to fire simultaneously if they do not share resources. Furthermore, an event-generated task should normally be repeated until some predefined stop criteria is satisfied. Under such conditions a Petri net simulation should faithfully follow the behaviour of modeled systems, this being characterized by transitions' firing and the location and movement of tokens between places. When all input places with token are searched and found, relevant transitions can be enabled and instantiated and the firing of transitions can begin. Thus the current flow of events occurring in the actual system should be similar to the computation processes which occur to execute transitions in the modeled system. All event (transitions) is enabled (fired) when all the preconditions (tokens in input places) are fulfilled.

In this respect the author concluded that it might prove highly beneficial to extend the capability of PN by investigating means of facilitating their use in a modular manner. It was envisaged that the use of modular PN modeling approach, within a control hierarchy for machine systems could be considered necessary and useful to enable them to be used to help support the life-cycle engineering of machine systems. Potentially the modularization of a PN model can be achieved by decomposing a PN based model into smaller model which then communicated with one another. Subsequently to use a modular PN (MPN) to support the life-cycle engineering of machine systems, a module design method has been established. The first author proposed that this could be guided by the principle of description of the machine (M) module; coordination between machine (M) modules; simulation of machine (M) modules; and implementation of real machine systems.

To facilitate the procedure for generating MPN for manufacturing machines, C code was developed by the first author. The program for the MPN modeling approach is generated by inputting the number of tasks (modules) which represent each machine (including their places, transitions and processing sequence), the number of parts and their operation sequence, and machine operation times and parts travelling

time. This code can be executed and provide a tabulation which represents the performance of a machine system.

Thus an approach to creating and using MPN modeling was proposed and analysed by the author ([7], [8]). The approach exhibits properties of a generalized Petri net within the confined task module. The establishment of each task module is based on the physical layout and operating sequence of the system. Another important inherent property of MPN is that they facilitate the development of manageable and easily extendible PN systems, which function as a virtual hardware analogue of the real system. MPN's flexibility when modeling the system can be achieved by defining a new Petri net module to represent additional hardware components. A MPN based model enables the interchangeability of component modules to be studied, without changing the whole machine layout, and also offers the possibility to add, delete and modify one or more machine units without affecting the others. However the operating sequence of a workstation could be changed following changeover of machines and part types within the workstation. Potentially this is particularly useful since a simulation tool could be designed to simulate breakdowns or work-in-progress interruptions in some of the machine units within the workstation. Thus the study of machine modularity and how to handle behavioural complexity in machines, especially as the number of machines grows, are important issues in this research.

A shared-transition (ST) is used to represent a communication path between at least two task modules (hardware components) and encodes state changes associated with that link, such when a job or part is transferred from one machine to another machine.

#### D. Basic Modular Petri nets (MPN) Definitions and Graph

A basic marked MPN conceived by the author is a seven-tuple,

$$\text{MPN} = (\text{P}, \text{T}, \text{I}, \text{O}, \text{ST}, \text{SP}, \text{m})$$

For which,

$\text{P} = \{\text{P}_{ij}\}$  is a set of places  $j$  in each task  $i$  where  $i = 1, 2, \dots, n$  (task number) and  $j = 1, 2, \dots, n$  (transition number)

$\text{T} = \{\text{T}_{ij}\}$  is a set of transition  $j$  in each task  $i$ , where  $i$  is the task number and  $j$  is the transition number

$\text{I} = \{\text{P}_{ij} \text{T}_{ij}\}$  is a set of place inputs into a transition (i.e. input function)

$\text{O} = \{\text{T}_{ij} \text{P}_{ij}\}$  is a set of place outputs from a transition (i.e. output function);

$\text{ST} = \{\text{ST}_x\}$  is a set of two or more single transition (T), where  $x$  is the ST number (i.e.  $\text{ST}_x = \{\text{T}_{ij}\}$  and  $x = 1, 2, \dots, n$ );

$SP = \{SP_{ij}\}$  is a set of shared-resource place in each task  $i$  and place  $j$ , and each comprises  $SP_{ij}$  comprises of corresponding shared-place transition, i.e.  $SP_{ij} = \{T_{ij}\}$

$m = \{m_1, m_2, \dots, m_n\}$  is a set of marked ( $m$ ) places in each task  $i$ , and token number  $n$ , defining the state of the system and token at particular time stamps.

A Petri net graph is a pictorial representation of a Petri net which consist of circle nodes, bar nodes that are linked by directed arcs. The circle nodes and bar nodes represent places and transitions respectively.

A shared-transition  $ST_x$  is comprised of single transitions when coordination between events occurring at two or more real machines (which are represented by modular nets) needs to be modeled and controlled.

## VII. ANALYSIS OF MODELING AND SIMULATION METHODS

### A. Modeling Deployment

Programming procedures corresponds to the above modeling mechanisms has been adopted within the deployed and developed MPN modeling and simulation environment which provide underlying formalisms and support the design and construction of machine control systems. The approach supports change by allowing alternative models and solutions to be evaluated i.e. the approach enables alternatives designs to be considered and compared. The corresponding simulation results can then be executed to evaluate machine performance, allowing decisions to be made prior to actual implementation.

The ability to support change can be characterized in further detail as follows:

#### *Addition, Deletion and Modification of the Physical Layout of Machine Systems*

Fig. 4(a) and (b) illustrated how the MPN modeling approach can readily support the addition, deletion and modification of the physical layout of a machine system. The example machine cell Fig. 4(a) is operated without (or with deletion of) machine M2 (see Fig. 4(b)). This machine (M) module deletion results to omission of the software input data for machine (M) module No. 5 and some modification for machine (M) modules No. 1, 2 and 3.

#### *Changes to Machine Activity*

Changes in the activities performed by a machine typically occur when jobs or parts change. For example shown in Fig. 4(b), indicates changes of robot activity due to the deletion of machine M2.

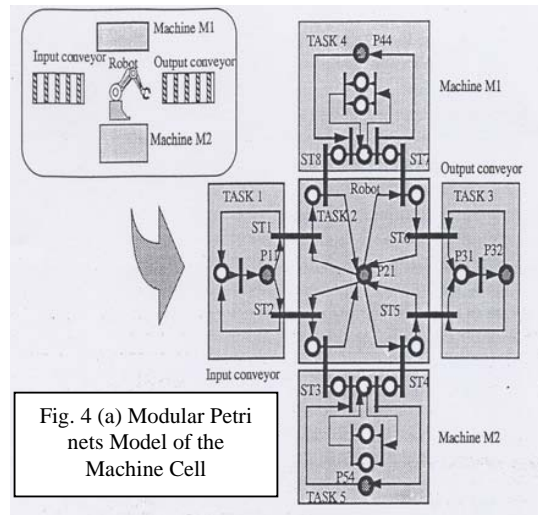


Fig. 4 (a) Modular Petri nets Model of the Machine Cell

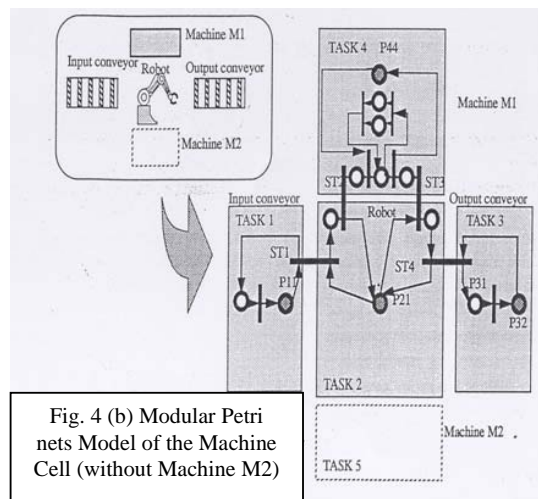


Fig. 4 (b) Modular Petri nets Model of the Machine Cell (without Machine M2)

#### *Changes in Operation Sequences*

Jobs or parts are normally sequenced through a series of workstations on a production line. Changes of jobs or parts will typically result in different sequences of operations. Making changes in operation sequences is similar to that of changing the strategy determining real-time control flow in machine systems.

Thus, it is a common concern for the modeling of different operation sequences to enable the processing of different part types, using same physical machine layout. Fig. 5(a) and (b) illustrate an example to the requirement. This is carried out by having two different operation sequences for two different part types.



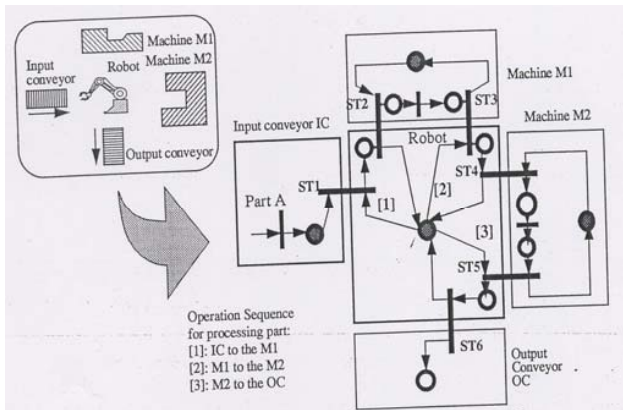


Fig. 5(a) Multiple parts production with different routing sequence: part A flow

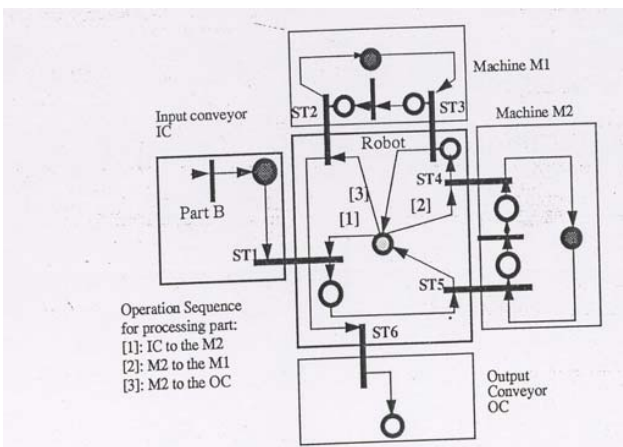


Fig. 5(b) Multiple part production with different routing sequence: part B flow

**B. MPN Simulation Environment**

Fig. 6 illustrates how the MPN modeling and simulation environment supports the production of multiple parts with different sequences of operation. Input data for each part type is prepared in a modular form, and establishes parameter associated with the task modules and the control flow of operation sequences. The files can be read by a task program, which can then be compiled and executed. The files and subsequent task programs can also be compiled and executed in any suitable environment.

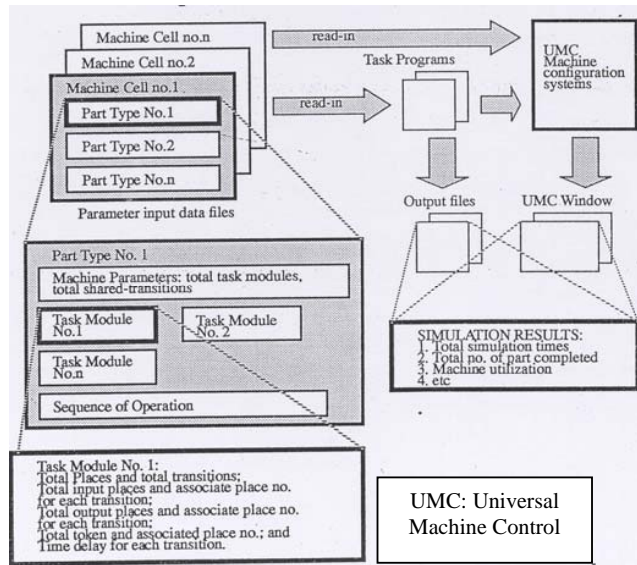


Fig. 6 The Elements of the Simulation Environment for Machine Control involving the Production of Multiple Parts

**C. Simulation Analysis and Results**

The research utilized a Petri net simulation software program based on Visual ObjectNet ++. The Visual Object Net ++ is an innovative Petri Net CAD/CAE Tool for PC that mixed continuous and discrete/event of Petri nets models. The building of nets, the simulation and visualization and documentations are integrated into the Modular Petri Net (MPN), as depicted in Fig. 7.

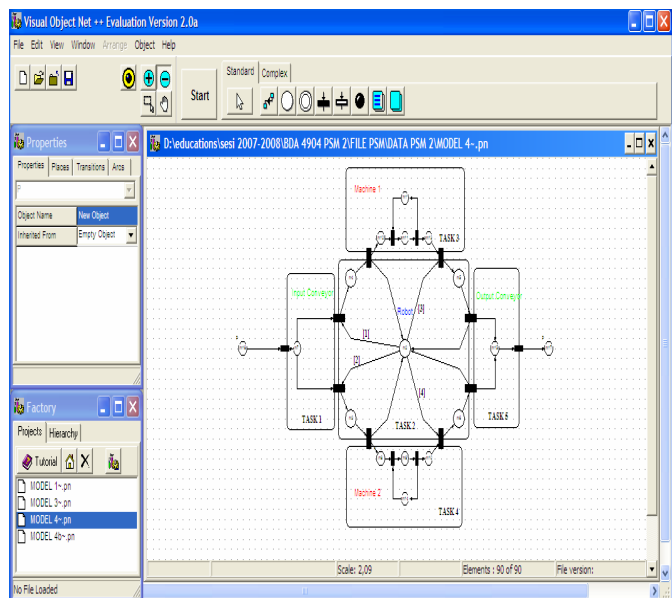
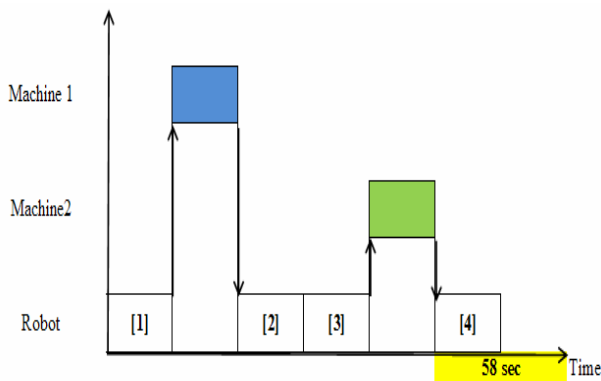
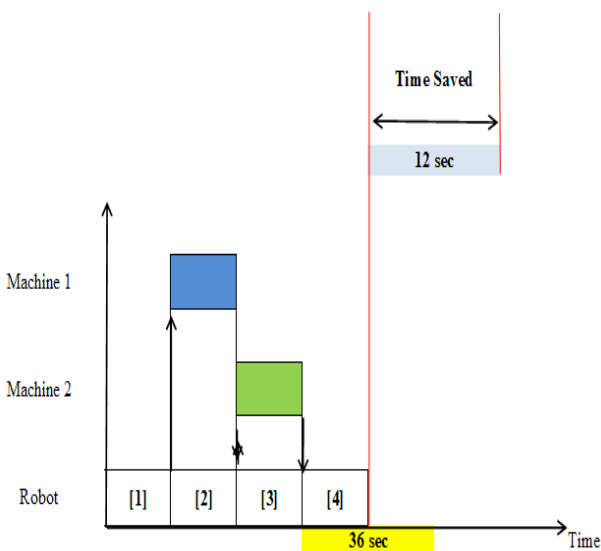


Fig. 7 A example Graphical MPN Layout using Visual Object ++ Software Tool



(a) Schedule 1: The robot and machines operate in sequence



(b) Schedule 2: The robot operates concurrent with machining process

Fig. 8 Timing Diagram for Machining Operation (based on Fig. 7)

A timing diagram can be plotted based on the use of the MPN modeling approach as shown in Fig. 8. This timing diagram clearly indicates synchronization events associated with the concurrent processes which will determine the level of utilization of each machine involved in the manufacturing cell (as illustrated in Fig. 7). In the above example, machining process Machine 1 can be done while robot picks up other part for Machine 2.

When using the MPN modeling and simulation environment, Job Schedule 1 and 2 can be prepared to generate the simulation depicted in Figs. 8(a) and (b) respectively.

The establishment of alternative scheduling and dispatching rules can thus be supported by the creation of modeling tools which capture the structural, functional and behavioural views

of a system yet are easy to understand and change. Also, such a facility can allow analysis and decision making in respect of the real-time control of machine systems. Thus, Fig. 8 compares timing analysis following such a change. It is observed that the total manufacturing time can be reduced by using the operation sequence described by Fig. 7.

#### D. Synchronization and Choice of Tasks

A workstation computer typically issues commands to machines to achieve: synchronization of operations among the tasks involved; and control of the individual sequence of each machine. The control task involved is a mixture of sequential control and real-time synchronization. Synchronization activities are important to reduce waiting times in any system and thus maximize production rate [30]. The example model shown in Fig. 7 demonstrates how a MPN modeling approach can describe synchronization among different tasks which have a deterministic sequence of operations.

Choices between different tasks are normally required when two or more machines are ready for an incoming part. This is illustrated in Fig. 7 where Machine 1 and Machine 2 are ready for incoming part type 1 and type 2 respectively. This is considered by Joshi and Smith [30] as a parallel mutual exclusion type of structure where a resource (i.e. robot) is shared by several processes or machines. In this case, one must make a choice about the dispatching of parts to machines and their sequence of operations.

An example for the choice of task between sequence and concurrent operation is illustrated in Fig. 8 and their corresponding results are depicted in Table III.

### VIII. CONCLUSION

The paper has focused on the combined use of architectural frameworks, which structure the definition of the real-time elements of machine control systems for the growing complexity of manufacturing machines; and modeling and simulation environment that support the design and construction of machines and their control systems. The deployment and development of MIMCA environment presents a new approach to the functional decomposition, modeling and simulation of machine systems and associated system-wide manufacturing operations. It provides the basis of a modeling and analysis approach based on Petri nets techniques which place emphasis on concepts of modularity, integratability and reconfigurability for real time control systems. A significance of the model is in its contribution towards the understanding of flexible and reusable control methods and provides a means of decomposing complex machine systems into reusable parts.

This paper has highlighted constraints when modeling based on the generalized Petri nets modeling approach. This understanding led to concepts which underpin the author's modular Petri nets (MPN) modeling and simulation environment. This was implemented to support the design and construction of distributed machine systems as potentially

TABLE III  
PRODUCTION ANALYSIS

Analysis	Concurrent operation	Sequence operation
Total simulation time (sec)	36	58
Total defect outputs (assumption) (part)	0	0
Total part completed / parts	2	2
Production Rate (part/min.) $= \frac{\text{Total Part Complete (part)}}{\text{Total Simulation Time (sec)}} \times 60 \text{ sec/min}$	3.33	2.07
Throughput (1hour operation) (parts) $= \frac{\text{Total Part Complete (part)}}{\text{Total Simulation Time (min)}} \times \text{Interest Time (min)}$	200	124
Percentage of Utilization $= \frac{\text{Processing Time In One Cycle (sec)}}{\text{Total Simulation Time (sec)}} \times 100\%$	58.33%	68.97%
Machine Speed Efficiency (%) $= \frac{\text{Processing Time per Cycle (sec)} - \text{Machine Time (sec)}}{\text{Processing Time in One Cycle (sec)}} \times 100\%$	28.57%	62.50%
Quality Rate (workdays = 8hr/day) $= \frac{\text{Actual Production Volume (pc/day)} - \text{Defect Output (pc/day)}}{\text{Actual Production Volume (pc/day)}}$	1	1

it has an inherent ability to solve deadlock and conflict issues while maintaining the flexibility and reusability of control software and hardware machine components. This should enable the design and control of distributed machine elements by facilitating the implementation of reusable software processes in a spectrum of machine control specification areas.

The concept of modular and scalable manner of the modeling architecture of MIMCA which has been based on designed MPN. Through utilizing a suitable simulation software program has been indicated and proven that a new production plan or any modification can be carried out quite easily. Thus, the MPN modeling approach can visualize changes in operating sequences and in a manageable and

scalable manner. The consequence simulation results can then be executed to evaluate machine, performance, allowing decisions to be made prior to actual implementation.

The current research still looking forward into concurrent and real-time control machine systems in more scalable and generalized in solutions when dealing with the growing complexity of manufacturing environment.

## REFERENCES

- [1] Albus, J. S., McGain, H.G., Lumia, R. 1989, NASA/NBS standard reference model for telerobot control system architecture (NASREM), Nat inst. Standard and Tech., Tech. Report 1235, Gainstherburg Md 1989.
- [2] Albus, J.S., 1991(a), Outline for a theory of intelligence, IEEE Transaction on Systems, Man, and Cybernetics, Vo. 21, No. 3, May/June, 473-509.
- [3] Albus, J.S., Quintero, R., Lumia, R., Rerman, M., Klmer, R. D. 1991(b), A reference model architecture for ARTICS, Manufacturing Review, Vol. 4, No. 3, Septermber.
- [4] Albus, J. S. 1991(c), Concept of reference model architecture for real time intelligent control system (ARTICS), NIST/NBS TECH NO. 1277.
- [5] Albus, J. S. 1992, RCS: A reference model architecture for intelligent control, IEEE Computer M, 56-59.
- [6] Angermuller, G., Niedmemayr, E. 1989, Off-line programming and simulation of flexible assembly, Assembly Automation, 9(2), 97-102
- [7] Ariffin, S., Weston, R. H., Harrison, R. 1995, A modular Petri nets Approach to the design of distributed machine control system., Procs of the 31st Int. MATADOR Conf., UMIST, U.K., April, 621-626.
- [8] Ariffin, S. 1996, Modeling and Simulation in Support of the Design and Construction of Modular Machine Control System, PhD Thesis, Loughborough University, U.K.
- [9] Beck, C. L., Krogh, B. H. 1986, Models for simulation and discrete control manufacturing system, Robotics and Automation IEEE Int. Conf. 1986, 305-310.
- [10] Ben-Arieh, D. 1991, Concurrent modeling and simulation of reactive manufacturing systems using Petri nets, Computers Industrial Engineering, Vol. 20, No. 1, 45-58.
- [11] Booch, G 1994, Object-Oriented Analysis and Design with Application, (The Benjamin/Cummings Publishing Company), 2nd Edition, 515.
- [12] Bostelman, R., Dagalakis, N., Albus, J. 1992, A robotic crane system utilizing the steward platform configuration, ISRM 1992 Conference, Santa FC, NM, November, 10-12.
- [13] Brussle, H. V 1990, Planning and scheduling of assembly system, Annals of the CIRP, Vol. 32/2, 637-644.
- [14] Carrot, A. J., Moore, P. R., Weston, R. H, Harrison, R. 1995, The UMC software environment for machine control system integration, configuration and programming, IEEE Trans . on Industrial Electronics, August.
- [15] Cho, H., Wysk. R. A. 1995, Intelligent workstation controller for computer – Integrated manufacturing problems and model, Jou. Of Manufacturing Systems, Vo. 14, No. 4, 252-263.
- [16] Dotan, Y., Ben-Arieh, D. 1991, Modeling flexible manufacturing system the concurrent logic programming approach, IEEE Trans on Robotics and Automation, February, Vol. 7, No. 1, 135-149
- [17] ESPRIT/AMICE 1993, CIMOSA Research Report, Project 688/5288, Vol. 1, Open Architecture for CIM, Spring-Verlag.
- [18] Goh, A. S. 1995, Step toward the design and control a new generation of machine, PhD Thesis, Loughborough University of Technology.
- [19] Harmonosky, C. M. 1990, Implementation issues using simulation for real-time scheduling, control and monitoring, Procs of the 1990 Winer Simulation Congerence, 595-598.
- [20] Harrison, R. 1991, A generalized approach to machine control, PhD Thesis, Loughborough University of Technology.
- [21] Hassan M. M. D. 1994, Machine layout problems in manufacturing facilities, Int. J. Prod. Res., vol. 32, No. 11 2559-2584
- [22] Hollocks, B. W., 1990, Improving manufacturing operation with WITNESS computer simulation AT &T Technology, Vol. 6, No. 1, 16-21
- [23] Hon, K. K. B., Ismail, H. S. 1991, Application of Transputers for simulation of manufacturing system – preliminary study, IMech E, Procs Instrumentation Mech Engineers, Vo., 205, 19-23.
- [24] Heidelberg, P. 1990, Discrete event simulation, Communication for the ACM, October , Vol. 33, No. 10, 29.
- [25] Hukan, H. M., Quintero, R. 1990, Task decomposition methodology for the design of a coal mining automation hierarchical real-time control system, 5th IEEE Proceeding Symposium on Intelligent Control, Vol. 2, 884-893.
- [26] Jeffrey, S. S., Sanjay, B. J., 1992, Reusable software concepts applied to the development of FMS control software, Int. J. Computer Integrated Manufacturing, Vol. 5, No. 3, 182-1961.
- [27] Jones, A. T., Mclean, C. R. 1996, A proposed hierarchical control model for automated manufacturing system, Jou. Of Manufacturing Systems, Vol. 5, part 1, 15-25.
- [28] Jorysz, H. R. Vernadat. F. B. 1990, CIMOSA Part 1: Total enterprise modeling and function view, Inst. J. Computer Integrated Manufacturing, Vol. 3, No. 3, 144-156.
- [29] Joshi. S., Wysk, R. Jones, A. 1990, A scalable architecture for CIM shop floor control, Procs of CIMCON 1990, NIST Gattersburg MD, 21-34
- [30] Joshi, S. B., Smith, J. S. 1994, Computer Control of Flexible Manufacturing Systems, Chapman & Hall, Chapter 10, 264-284
- [31] Knapp, G. M., Wang. H. P. 1992, Modelling of automated storage/retrieval systems using Petri nets, Journal of Manufacturing Systems, Vol. 11, No. 1, 20-29
- [32] Kumar, D. 1990, An approach towards distributed simulation of timed Petri nets, Procs of the 1990/Winter Simulation Conference, New Orleans, 428-435.
- [33] Levis, A. H. 1992, A coloured Petri nets model of intelligent nodes, Robotics and Flexible Manufacturing Systems (Elsevier Science Publishers), 369-379.
- [34] Miller, D. J., Lennox. R. C. 1990, An Object-oriented environment for robot system architecture, Robotics and Automation, IEEE Int. Cong., 352-361.
- [35] Peterson, J. L. 1981, Petri net Theory and the Modelling of Systems (Prentice Hall, New Jersey).
- [36] Pritschow. G. 1990, Automation technology – on the way to an open system srchitecture, Robotics and Computer Integrated Manufacturing, Vol. 7, No.1/2, 103-111.
- [37] Pritschow, G. Junghoms, D. G. Sperling, W. 1993, Open system controllers, - a challenge for thefuture of the machine fool industry, Annals of the CIRP, Vol. 42/1, 449-452.
- [38] Quintero, R., Barbera, A. J. 1993, A real-time control system methodology for developing intelligent control system, Robot System Division, NIST.
- [39] Schaeffer, C. 1991, Open motion control (OMC), Specification Document History V1.0 (Draft Version).
- [40] Shaharoun, A. M. 1994, A new approach to the life-cycle support of costing system for advanced manufacturing environment, PhD Thesis, Loughborough University of Technology.
- [41] Sorensen, S. 1992, Overview of a modular, industry standard based open architecture machine controller, CIMETRIX Inc. Report.
- [42] UMC 1992, Universal Machine Control (UMC) Summary Document Issue 3, Modular System Group, Loughborough University of Technology.
- [43] Weston, R. H. et. al. 1989, Universal machine control primitives for Modular distributed manipulator systems, Int. J. Prod. Res., Vol. 27, No. 3, 395-410.
- [44] Weston, R. H. et. al. 1992, Advanced architecture for universal machine control, Final ACME Report, GR.F 13492, Loughborough University of Technology.
- [45] Wiendahl, H. P., Garlicks, R, Zengtrager K. 1991, Modeling and simulation of assembly system, Annals of CIRP, Vol. 40/2, 557-585.
- [46] Wright, C., Harrison, R. 1994, IMDC: an integrated environment for the design and control of manufacturing machines, Procs. Advances in Manufacturing Technology VIII, 10th National Conference of Manufacturing Research, Loughborough University, September, 355-359.
- [47] Wright, P. K. 1995, Principle of open architecture manufacturing, Journal of Manufacturing Systems, Vol. 14, No. 3, 187-201.
- [48] Wysor, R. W., 1993, Open architecture in control system integration, Intelligent Motion, October 1993 Proceedings, 17-21.
- [49] Yim, D. S., Barta, T. A. 1994, A Petri net based simulation tool for the design and analysis of FMS, Journal of Manufacturing Systems, Vol. 13, No. 4, 251-261.
- [50] Zhou, M. C., DiCesore, F., Rudolph, D. 1990, Control of flexible manufacturing system using Petri nets, IFAC Symposia Series- Procs of a Triennial World, 47-52.