15th Triennial World Congress, Barcelona, Spain

A MODELING AND SIMULATION APPROACH: TOWARDS TRUE MANUFACTURING FLEXIBILITY

Mercedes Narciso*, Miquel A. Piera*, Antoni Guasch.**

** Unitat d'Enginyeria de Sistemes i Automàtica. Dept. Informàtica. Universitat Autònoma de Barcelona.*
*** Dept. D'Enginyeria de Sistemes i Automàtica. IRI. Universitat Politècnica de Catalunya*
mnarciso@sunaut.uab.es, miquel@sunaut.uab.es, guasch@iri.upc.es.

Abstract: To produce a large number of different products in an efficient way, traditional industry production architectures have incorporated non-value-adding operations such as transporting, storing and inspecting to provide flexibility. These operations should be minimized, if not eliminated, to improve performance on automated manufacturing systems. Heuristic planning tools seem to offer a good methodology to deal with a sub-optimal scheduling policy of present flexible manufacturing systems. In this paper a simulation approach, based on Coloured Petri Nets, to evaluate and improve production planning policies will be presented.

Keywords: Flexible Manufacturing Systems, Scheduling, Coloured Petri Net, Simulation, Coverability tree.

1. INTRODUCTION

World-wide market competition, high product quality requirements, together with random demands instead of steady demand, are some key-factors which have forced traditional rigid and/or non-automated *production architectures* (such as Flow Shop, Job Shop) to change towards flexible manufacturing architectures (FMS) Tempelmeier (1993).

A flexible manufacturing system (FMS) is a production system consisting of a set of identical and/or complementary numerically controlled machines which are connected through an automated transportation system. In addition, each process in a FMS is controlled automatically by a dedicated computer. Under *ideal* operating conditions, a FMS is capable of processing workpieces of a certain workpiece-spectrum in an arbitrary sequence with negligible setup delays between operations. However, setup delays (*real operating conditions*) can decrease FMS performance results drastically if decision variables, such as processing, handling, storing and transportation, are not well coordinated.

Due to the considerable advances in the technological field, hardware flexible processing units (such as CNC's, robots, conveyors, etc) can be easily incorporated in present manufacturing systems to increase flexibility in the production system.

Nevertheless, true production flexibility requires not only flexible hardware mechanism, it is essential flexible software tools to manage, control and synchronize the flexible hardware components. Most present planning, scheduling and control software tools lack of flexibility to react efficiently in front perturbations. Thus traditional *production planning tools* are forced to evolve towards new heuristic based methodologies which could cope with a large amount of decision variables inherent to present FMS architectures.

The exact optimal solution of a FMS planning problem is quite complex and difficult, may be impossible to obtain. Furthermore, when system behavior is subject to random disturbances, system uncertainty should be described by the mathematical model (stochastic processes), which will increase considerably the complexity of the methodology to be implemented to determine the optimal solution.

Conventional planning tools that handle scheduling problems by using analytical techniques often fail to catch the appropriate level of detail when applied to FMS. For instance, Queuing theory methods can model the steady state operations, but they fall short to deal with transients. Hierarchical Planning Tools offer good results to deal with complex problems which might be decomposed in "independent" subtasks (ie. short, medium and long term production planning) Sethi (1994). However, these tools are inadequate to solve internal FMS planning problems, where machine scheduling can not be treated as independent targets.

Intrinsic FMS characteristics which constrains the use of some traditional production planning techniques are:

- Uncertainty in demand and time production: FMS production and transport units behave as discrete event oriented systems.
- Large decision variables amount. Note that while *flexibility* is essential to competitiveness, the amount of decision variables which should be coordinated and synchronized in an efficient way is a major drawback.
- Quick solutions to react in front perturbations: Most optimal planning techniques are CPU intensive (time consuming) which make them unsuitable to be used for re-scheduling purposes.

Furthermore, non-value-added operations (such as transporting, storing and inspecting) incorporated in manufacturing systems to allow a higher flexibility level, are the main operations which should be minimized, if not eliminated by the planning tool. These opposed goals force to look for new planning methodologies which could lead with a good weighted compromise between different economic objectives to be satisfied.

To be able to control the production efficiently, the controller must have an appropriate model of the manufacturing system, as well as a model for all the products manufactured. Each product description, together with the resource requirements, proffers a route through the system.

To achieve a truly flexible manufacturing system, it is essential to design a control system that could determine the best scheduling policy by experimenting with both:

- ▪ Model of the manufacturing resources
- ▪ Model of the manufactured products.

To deal with such a planner, an Object Oriented Coloured Petri Net Simulation Tool which could be used as a generic planning tool to cope with the scheduling and routing problems of a FMS has been developed. Section II introduces the needs to describe the logic constraints which appear between FMS units, and the advantages of using Coloured Petri Nets as a tool to formalize the model. Section III presents a time consuming methodology to determine the optimal plan. Finally, section IV summarizes the benefits of the implemented tool to evaluate and improve heuristic planning tools.

2. INTEGRATED MODELING OF MANUFACTURING RESOURCES AND MANUFACTURED PRODUCTS IN CPN

Coloured Petri Nets (CPN) have shown to be successful tools for modeling FMS's due to several advantages such as the conciseness of embodying both the static structure and the dynamics, the availability of the mathematical analysis techniques, and its graphical nature (Jensen 1997, Silva 1989, Zimmerman 1996). Furthermore, CPN are very suitable to model and visualize patterns of behavior comprising, concurrency, synchronization and resource sharing, which are the main characteristics of a FMS.

Despite a PN model might be suffice to describe the logic constraints between a list of resources (processing machines, transport units, local stocks) a list of operations, and their precedence relationships, it lacks of information data representation independent of the system architecture, which is essential to deal in an efficient way with the best production policy for a given system state Zhou (1999).

A CPN can be used as the specification of the system that we want to build up, or as a compact presentation of a system whose behaviour want to be improved. The process of creating the description and performing the analysis usually gives the modeler a improved *understanding* of the modeled system, Jensen (1997). The behavior of a CPN model can be *analyzed*, either by means of simulations or by means of more formal analysis methods.

Therefore the CPN formalism allows to describe both the complete structure of the system together with its behavior, and the information about the system state by means of a compact representation, which facilitates the maintenance of the model , Piera (2001).

2.1 . Example

A small factory has been chosen to illustrate some usual FMS production planning characteristics, such as:

- A FMS production planning method should allow for various hierarchical decision-making levels, from simple deterministic conditional branches to multicriteria, multiconditional branches.
- The proposed system provides a source for evaluating several alternative designs, and correct

the heuristic kernel to choose a desired sequence of transitions.

– The problem shows a dynamic structure, which allows the investigation of several different decision strategies.

A manufacturing system has been designed to process three different types of pieces (each one with different manufacturing operations) at any available manufacturing unit, and to assemble them at specified sequence order according to customer specifications.

Figure 1 shows the main elements that assemble the whole FMS:

1. A round conveyor with 8 positions
2. A Load/Unload unit: A shared robot (Ro=1) is used to load pallets at position 1 from row material stock (stock i=1), and unload them at position 7 to final product stock (stock i=6).
3. Two independent manufacturing units: each one is assembled by a local Robot (Ro=3 & Ro=5), a local stock (i=3, & i=5), a local CNC machine (j=10 & j=11) and a local assembly machine (p=3 & p=5) each one with 3 slacks for each type of piece.

Finite capacity of local stocks constrains both: the material flow and the lotsizing scheduling policies. This characteristic together with the performance differences of each CNC force to run several simulations in order to determine the best production architecture. It should be noted that each simulation force the modeler to change some code in order to describe the new experiment. When using the simulation approach to deal with the best planning policy of a real industrial production system, simulation model maintenance usually becomes a huge problem that constrains the nº of experiments that can be performed.

To avoid this type of problems, the modeling formalism should allow in an easy way both:

- To specify which is the new h/w system configuration
- To specify the flexibility of the system in such a way that the best configuration could be evaluated automatically from the simulation model.

This academic FMS can be used to illustrate the huge number of possible transition sequences that could be fired to process a certain amount of material, according to different hardware configurations. Consider for example (Narciso 2001) :

1. Single pallet and processing time in the units greater than the transportation time throughout the conveyor.
2. Three pallets and processing time in the units greater than the transportation time of pallets.
3. Processing time in unit less than the transportation time of pallets.

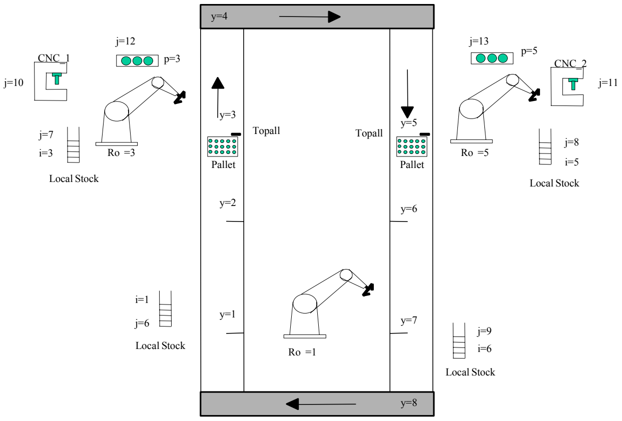

Figure 1 A Flexible Manufacturing System

Different alternatives of allocation of tasks should be found each time a parameter of the system is changed, as for example they are: time of load and unloading of employee by each robot, number of processing units, processing time of each unit, number of pallets, etc. Nevertheless, to reach the best solution will depend on a faithful representation of the system and its behavior when it is modeled by means of the CPN.

Figure 2 and Figure 3 shows the CPN model of the conveyor subsystem and the load/unload subsystem respectively. Table 1 outlines the Places description.

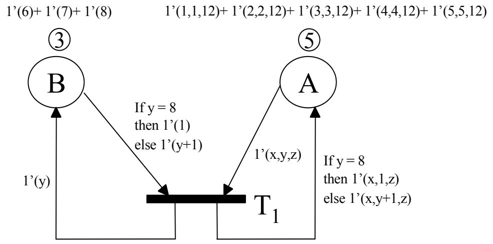

Figure 2 : CPN of the conveyor subsystem

A plan for this system will consist to define an ordered sequence of transitions (or actions) in such a way that firing this sequence will drive the system from its initial state to a certain final state. The best heuristic planning tool will be the one that will reach the final state in a minimum time. Note that the time to drive the system from an initial state to the final state can be computed by specifying to the CPN model the time associated with each transition.

Table 1: Colours and Places description

Place	Description
A	It represents the information associated with the pallets: pallet identifier, position in conveyor, free spaces number.
B	It indicates the conveyor's free positions.

C	It indicates the state of each robot.
D	It represents the information associated with each piece: Type of piece=(1..3: Type of original piece, 4: Type 2 processing, 5: Type 1 processing, 6: Assembled piece 123, 7: Assembled piece 13) and Position=(1..5: in pallet, 6: in stock_0, 7: in stock_1, 8: in stock_2, 9:in stock_3, 10: in CNC1, 11: in CNC2, 12: in assembler_1, 13: in assembler_2)

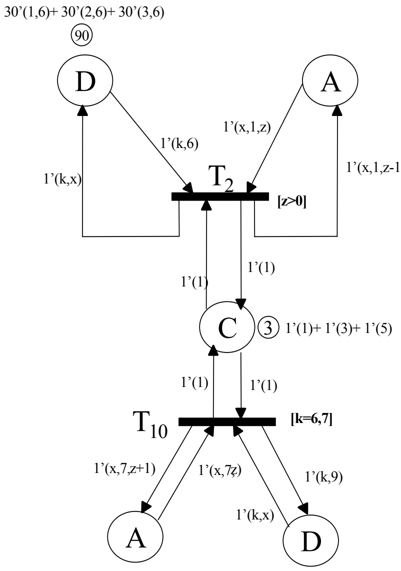

Figure 3 : CPN model of the load/unload subsystem

2.1 The Coverability Tree

The coverability tree allows to describe both (Jensen 1997, Proth 1996) :

- All the FMS states (markings) which can be reached starting from a certain initial system operating conditions M_0.
- The transition sequence to be fired to drive the system from a certain initial state to a desired end-state.

The main disadvantage of Coloured Petri Nets lies in the size of the marked graphs (coverability tree) produced by modeling very complex discrete event systems, such as flexible manufacturing systems. For these large nets, although analytic techniques are available for their processement, the computation involved in analyzing them is quite substantial, and in many cases, it may not be practical.

The scheduling goal will consist to find the sequence of operations that will allow to drive the system from its original state to the final state (M_f) . However, Since it will not always be possible to build the overall coverability tree, two solutions can be adopted:

- Choose a higher abstraction level to describe FMS particular characteristics.
- Use some heuristic to prune the tree.

The former option can reduce the size of the coverability tree considerably, however, it is not possible to afford a plan with appropriate detail to generate a planning that could contemplate all the production architecture decision variables. Although the latter option does not guarantee that the optimal solution will be found, it offers a good answer to Industry scheduling and re-scheduling requirements which are forced (due to time constraints) to accept a prompt quick sub-optimal solution rather than a delayed optimal solution. Note that a fast re-scheduling is essential to quickly react when disturbances appear.

3. A TOOL TO DETERMINE THE OPTIMAL PLAN

The coverability tree of a FMS can grow in exponential form with respect to the number of events that could be fired in parallel. Due to computer restrictions, it is not possible to build, analyze and maintain the complete coverability tree in a computer memory.

An algorithm has been implemented to go over the whole coverability tree just by storing in memory a static structure (objects) describing the FMS Coloured Petri Net, and a dynamic structure (binary tree search) which stores in each element (node):

- Marking description: Information associated with tokens
- Information about the transition fired.
- Time information of tokens used by the fired transition.
- Time information about the marking
- Cost of reaching that state
- An identifier that indicates the marking from which was generated the present marking (father node).

The algorithm begins with an initial marking, and determines all the enabled transitions for this state. A transition is selected, a new state is generated, and its existence in the binary tree is checked. If the new marking has been generated previously in some other level of the coverability tree (old node), the algorithm will not explore the enabled transitions associated with the new state.

Nevertheless, if the arrival time to the new marking is shorter than the arrival time of the old node, the algorithm updates his associate times by the one of the generated node. The same update is made for all the markings generated from the old node. On the other hand, if the generated marking corresponds to a new state non reached previously, a new node is added to the tree, and all the enabled transitions from this new state are computed and processed in a similar iterative way again.

When a generated marking agrees with the goal state, the branch of the tree is stored in a file from the initial state to the reached final state, with the corresponding information on the transitions, times and costs of each marking of this route.

It should be noted that for each node of the binary tree, the identification of the father node together with the arrival transition to the marking is all the information required to to backward to the previous marking and analyze another marking by selecting another enabled transition. Thus, it is possible to generate the entire coverability tree of a system and to determine the best sequence of actions that drives the FMS from an initial state to a goal state.

Nevertheless, due to computational restrictions it can be impossible to generate the whole tree, thus the algorithm allows to divide the construction of the tree as a set of subtrees: from the initial marking the entire tree up to a specified deep (level **l**) is built. All the nodes at this specific level are evaluated, and the best **n** nodes are selected as the root of the new **n** coverability trees of deep **l** levels. This procedure is applied iteratively until obtain a final state.

3.1 The Cost Function

The formalization of an objective function to drive the program through the search space, will allow to summarize certain expert knowledge and express it in the mathematical formalism used by the search algorithms. The knowledge expressed through the objective function can be used to select those markings (states) within the solutions space which could lead to the optimal solution.

In industry, production requirements are defined usually as a compromise between time and cost. To assess a production process, the engineer has to be aware of performance indexes such as: total time that a part spends in a queue; total time that parts spend in transport systems; equipment utilization; proportions of time a machine is down (waiting for parts of a previous work station), blocked (waiting for a finished part to be removed), or undergoing setup operations, etc.

Attempting to group the production performance indexes, a corresponding cost function is defined. This cost function is formalize by two components: a place or `work in process' (WIP) component and a time component.

‘Work in process’ is the current number of pieces (or quantity of material, in the processes industry case) in the production line. In terms of Coloured Petri Nets, the WIP can be obtained by computing the sum of tokens in every place representing a stock. Thus, the cost that a company pays for pieces stored in particular queues (places), can be expressed mathematically by an objective *P-function* (see eq. 1), where P_i represents the internal values of the place i (number of pieces or tokens stored in queue i), and A_i is a weighting parameter defined by the user.

$$Jp = \sum_{i=1}^{n} A_i \times P_i \tag{1}$$

Note that the performance of the *P-function* depends on the tuning of the weighting parameters. Thus, the final user can penalize those places where tokens should not remain for long (row material places).

In the studied case, different policies have been studied by changing the weight of the tokens in places such as the number of pieces in stocks (in order to empty stocks as soon as possible), or the number of pieces in pallets (in order to use them for shorter or longer time), etc.

Figure 4 illustrates how the best sequences of actions to drive the system from an original state to a desired end state can be determined by pruning the coverability tree.

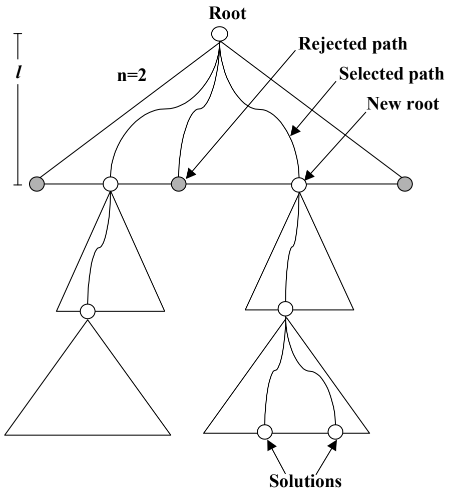

Figure 4 Heuristic Pruning Algorithm.

4. SCHEDULING RESULTS

The proposed methodology has been applied to the flexible manufacturing system described in example 2.1, obtaining following results .

Figure 5 shows the gannt diagram with the first sequence of operations to be made by the FMS when the transport system has 2 pallets to feed or remove material from the 2 subsystems. Table 2 outlines the meaning of each task.

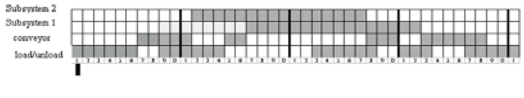

Figure 5 : Gannt diagram of the FMS with 2 pallets

Table 2: Sequence of task with 2 pallets

time	subsystem	task

interval		
[1-6]	Load/Unload	Load 6 pieces in pallet at P1
[7-11]	Conveyor	Transport pallet from P1 to P6
[7-8]	Conveyor	Transport pallet from P9 to P1
[9-14]	Load/Unload	Load 6 pieces in pallet at P1
[12-27]	Subsystem 2	Process pieces from/to the pallet
[15-16]	Conveyor	Transport pallet from P1 to P3
[17-32]	subsystem 1	Process pieces from/to the pallet
[28-30]	Conveyor	Transport pallet from P6 to P8
[31-33]	Load/Unload	Unload 3 processsed pieces
[33-38]	Conveyor	Transport pallet from P3 to P8
[34-36]	Conveyor	Transport pallet from P8 to P1
[37-42]	Load/Unload	Load 6 pieces in palet at P1

Figure 6 shows the gannt diagram with the first sequence of operations to be made by the FMS when the transport system has only 1 pallet to feed or remove material from the 2 subsystems. Table 3 outlines the meaning of each task.

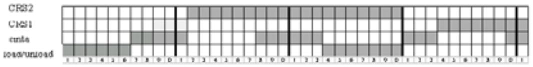

Figure 6 : Gannt diagram of the FMS with 1 pallet

Table 3: Sequence of task with 1 pallet

time interval	subsystem	task
[1-6]	Load/Unload	Load 6 pieces in pallet at P1
[7-11]	Cinta	Transport pallet from P1 to P6
[12-30]	CRS2	Unload from the pallet and process pieces from/to the local stock
[18-23]	Cinta	Transport pallet from P6 to P1
[24-30]	Load/Unload	Load 6 pieces in pallet at P1
[31-33]	Cinta	Transport pallet from P1 to P3
[34-52]	CRS1	Unload from the pallet and process pieces from/to the local stock
[39-41]	Cinta	Transport pallet from P3 to P6
[42-45]	CRS2	Unload 3 processsed pieces

It can be easely noted that when the FMS must be operated with only 1 pallet, the transport system can become a bottleneck, and consequently the best scheduling consist to optimize the pallet use increasing consequently the number of operations (unload from pallet to local stock) in each subsystem. However, when the FMS is operated with 2 pallets, the manipulator in each subsystem can become a bottleneck, and consequently the best scheduling consist to minimize the number of load/unload operations in each subsystem which forces to use the pallets as local stocks.

CONCLUSIONS

An Coloured Petri Net Simulator has been developed to generate a schedule policy in order to satisfy certain production goals on a flexible manufacturing system. The proposed framework allows to drive a Coloured Petri Net simulator from an initial state to a desired goal state.

The high number of decision variables in present flexible manufacturing systems, usually can lead to a huge coverabilty tree, which make practically impossible its computational handling. The CPN simulator has been developed in such a way that different heuristic algorithms can be implemented to drive the simulator to deal with the best scheduling policy.

At present, a distributed platform of this prototype is under consideration in order to improve the computational time to obtain the best scheduling policy of a high flexible production system.

REFERENCES

Tempelmeier and H. Kuhn, *Flexible Manufacturing Systems*, Wiley; 1993.

S.P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, Birkhauser, MA; 1994.

K. Jensen, *Coloured Petri Nets: Basics Concepts, Analysis Methods and Practical Use,* vol. 1,2,3, Springer-Verlag, Berlin; 1997.

M. Silva and R. Valette, “Petri Nets and Flexible Manufacturing”, *Lecture Notes in Computer Science*, vol. 424, Advances in Petri Nets 1989, pp.374-417.

A. Zimmermann, K. Dalkowski and G. Hommel, “A Case Study In Modeling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets”, *in Proceedings of the 8th European Simulation Symposium (ESS '96)*, pp. 282-286.

M. Zhou and V. Kurapati, Modeling, Simulation and Control of Flexible Manufacturing Systems: a Petri Net Approach*, Series in Intelligent Control and Intelligent Automation, vol. 6, World Scientific, NJ; 1999.*

M.A. Piera and T. Guasch, “Coloured Petri Net for Simulation FMS Model Maintenance”, *EUROSIM’01*, 2001.

M.E. Narciso and M.A. Piera, “Sistema Automático de Planificación de la Producción para una Célula Flexible de Fabricación”, *Workshop en Metodologías de Modelado y Simulación de Sistemas*, 2001.

J.M. Proth and X. Xie, Petri Nets. *A Tools for Design and Management of Manufacturing Systems*, Wiley, 1996.