Automatic Generation of the Production System Control

Francisco Yastami Nakamoto
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.
UNIFIEO - Universidade Fundação Instituto Ensino para Osasco, São Paulo, Brazil.
e-mail: francisco.nakamoto@poli.usp.br; yastami@unifieo.br

Paulo Eigi Miyagi
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.
e-mail: pemiyagi@usp.br

Diolino José dos Santos Filho
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.
UNIFIEO - Universidade Fundação Instituto Ensino para Osasco, São Paulo, Brazil.
e-mail: diolinos@usp.br

Abstract. *The recent market demand is very dynamic; consequently the product life cycles tends to be more and more short and a variety of products must be produced in a simultaneous manner. The simultaneous productions of different items request the project of control systems to Flexible Production System (FPSs) to manage the execution of multiple process simultaneously sharing a finite set of resources. One important aspect of FPSs is the complexity present in the global process. Another aspect is that the multiple and simultaneous execution of processes, sharing the system resources can lead the system to reach to a deadlock state. Deadlock is a phenomenon that occurs when the flow of the activities of the processes is permanently impeded due to the resource lack and/or information lack. This work presents a systematization of the resource control design, including the definition of deadlock avoidance rules. This systematization which applies an algorithmical approach enables the automatic generation of the resources control by computational tools. Important results of this work it the implementation of a prototype of a tool to automatic resources control generation.*

Keywords*: Deadlock Avoidance, Petri Net, Production System Control, and Resource Allocation.*

1. Introduction

The Flexible Production Systems (FPSs) are characterized by the simultaneous execution of several processes, sharing a same set of resources. The FPSs belong to the class of Discrete Events Systems (DES) (Ho e Cao, 1991; Cassandras, 1993), therefore the dynamic behavior of FPSs is based on the occurrence of events, on the eventual occurrence of parallelism and conflict of events. Besides, it is possible exist a certain indeterminism regarding the occurrence of these events in function of the time (lack of synchronism). Thus, to improve the efficiency of all processes in FPSs is required a larger control in the utilization of equipment. The control of one process is relatively simple since it is just necessary to guarantee the sequencing of each process. However, in a production system with multiple processes being executed simultaneously, where those processes share a finite set of resources, the systems may eventually freeze; that is called *deadlock* or *deadlock status*. This phenomenon occurs when the flow of processes is permanently interrupted and/or when the operations of the processes cannot be executed.

Several works propose solutions to deal the *deadlock* through a restriction model (Banaszak e Krogh, 1990; Viswanadham et all, 1990; Kumaran et all, 1994; Fanti et all, 1995; 1997; Santos Filho, 2000b). The implementation complexity of these methods is proportional to the complexity of the system, and it may become unfeasible due to the great effort computational required. Another important aspect is related to the resources allocation control. In Banaszak e Krogh (1990), Kumaran et all (1994) and Santos Filho (2000b) the control of the processes and the control of the resources are handled in the same level. In Fanti et all (1995; 1997) the strategies are applied in the resources control.

The method proposed in Santos Filho (2000a) for FPSs consists in dividing the control system into (i) control of the processes and (ii) control of the resources. The structure of two levels enables control of several processes through a restriction model, which determines how the resources should be allocated during the processes dynamics by the addition of control rules. The models of the control system are created using graphic and mathematical tools based on Petri Nets (Peterson, 1981; Reisig, 1985; Miyagi, 1996; Murata, 1998). This model of the dynamic behavior of the FPSs with this architecture of two levels avoids the *deadlock* in system. The tools used in this work are (i) Enhanced Mark Flow Graph (E-MFG) (Santos Filho, 2000a) and the (ii) Resources Allocation Graph (RAG) (Santos Filho, 2000a), associating the production rules to represent the set of restrictions to the control system.

This work aims to introduce the systematization of the technique proposed by Santos Filho (2000a), in other words, the objective is to introduce an algorithmical approach which allows to implement a computational tool for automatic generation of the control algorithm for FPSs. The algorithms supervise the processes regarding the resources allocation and it accomplishes decisions in real time to avoid the *Deadlock State*.

Section 2 provides a description of deadlock concepts in FPSs context. The section 3 characterize why is so difficult to control a FPSs, how to board the complexity of the FPSs, what are the modeling tools and the synthesis of the method to generate the resource control. The section 4 and section 5 introduces respectively the algorithmical approach, and illustrates one example using this method. The section 6 presents the conclusions and contributions of this work.

2. Deadlock in FPSs

In a FPSs each process executes a pre-defined of sequence of activities, or stages, and for the execution of each activity, it uses determined resources. The competition of resources by the processes can lead the system to deadlock, because two or more processes compete for a limited number of resources and/or information. For example, the processes are waiting for the liberation of resources, which are being used by other processes.

For *deadlock* occurrence in FPSs, the following conditions must be present (Banaszak e Krogh, 1990; Cho, 1993; Santos Filho, 2000a):

(i) *Mutual exclusion*;
(ii) *Retention while it waits*;
(iii) *No preemption and;*
(iv) *Circular waits*.

The *deadlock* will occur if the four terms are true. Thus, it is enough guarantee that at least one of the above conditions is never satisfy. In a FPS the first three conditions are always true, that is, two or more processes cannot use a resource simultaneously and only the process that allocates a resource can free it. Consequently, the retention occurs while the process waits the available resource; therefore, circular waits remain being the condition that can be controlled.

In this paper, we consider the problem of *Part Flow Deadlock* (Cho, 1993) in *Single Resource Allocation* (Lawley et all, 1997) using the *deadlock avoidance* method. Therefore, the processes are in a circular situation, waiting for the resources liberation allocated by other processes and to execute the process stage is necessary only one resource type.

3. System Control of FPSs

The control system accomplishes pre-established functions to reach a pre-defined objective. In FPSs context, such functions are the execution of activities and/or operations through several control levels (Santos Filho, 2000a). This section characterizes the complexity of FPSs and how to board the complexity of the FPSs, what are the modeling tools and the synthesis of the method of resource control.

3.1 Complexity of Flexible Production Systems

The concept of complexity can be defined in many ways and in different contexts. However, there is no general definition to every situation (Calinescu et all, 2000). The term "*complexity*" comes from the Latin, 'complexus', which means interlaced or twisted together; i.e., it takes two or more different parts or components which are in some way linked to each other, forming a stable structure (Palazzo e Castilho, 1998). According to Edmonds (1995), the complexity is the property of representation in which the global behavior is hard to be formulated even when all information about the components and their interrelations are provided.

In FPSs context, the complexity is a property that system models may have (Santos Filho, 2000a), in other words, the complexity can arise when a certain dynamic behavior is imposed to PSs.

Imposing a dynamic behavior for a single process is relatively simple, because it involves only the guarantee of the sequencing of the predetermined stages of the process. In global context, it is not possible to determine the behavior of the system in a manner purely sequential. That is, in the global context the system becomes complex because it is not possible to pre-define the desired dynamic behavior. The complexity of the global process is due to two levels of indeterminism, which start coexist (Santos Filho, 2000a):

- Indeterminism regarding time: it is not possible to determine when a certain event is going to occur, because the FPSs are SEDs so it is not possible to predetermine the instant in which a certain event will occur;
- Indeterminism regarding the sequence of events: it is not possible to determine which event precedes another, because the execution of simultaneous processes sharing a finite set of resources.

Consequently, despite knowing the individual behavior of each process, that is, a set of pre-determined sequential stages, it is not possible to describe the system behavior as a whole since several processes are being executed simultaneously. This characteristic is present in complex systems.

3.2 Architecture of the Control System

In FPSs the number of combination of reachable states can be exponential depending on the number of processes, resources and other variables, which complicates control. The solution adopted in this work contemplates three fundamental aspects.

The first one is based on restriction models (Santos Filho, 2000a). The restriction models are proposal because there are cases where it is not possible to determine all the reachable states of the system. However, through additional rules of control, it is possible to avoid that the system reaches determined undesirable state.

The second aspect is related to the human element. The model adopted in this work was the anthropocentric control systems proposed in Santos Filho (2000a); in other words, there is the need to make the human element a part of the system. The anthropocentric systems are man-made systems (Ito, 1991; Santos Filho, 2000a). They are systems that incorporate new technologies in which the human element, possessor of the knowledge, participates in the requisite specification, development, implementation and monitoring and, whenever necessary, executes actions which interfere in the dynamics of the system.

Another aspect refers to the division of the control system in two-module (Santos Filho, 2000a) (figure 1).

- Control of processes: Responsible for the sequencing of the stages of the processes, allocating available resources;
- Control of resources: Responsible for managing the utilization of the resources by the processes. In fact, it is a supervisory control.

The control of resource will monitor the stages of the processes regarding the utilization of resources according to a set of additional control rules. Both module levels have distinct semantics: when a process requests a certain resource, this solicitation is made to the control of resources. If this resource can be allocated, the control of resources indicates that the resource is being used. The next step, the control of resource sends a signal to the control of processes indicating that the resource can be allocated. Thus, the utilization of the resource in control of resource determines the allocation of the resource in control of process.

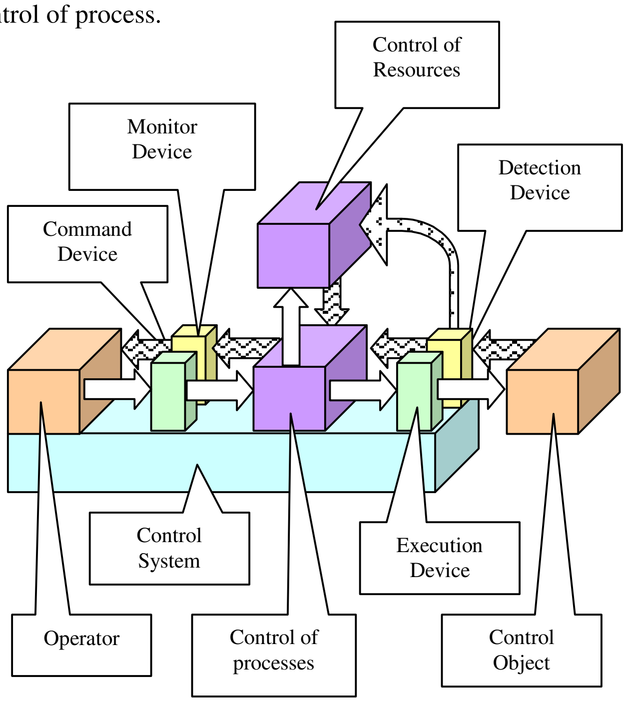


Figure 1 Control system in two hierarchical levels.

3.3 Modeling Tools

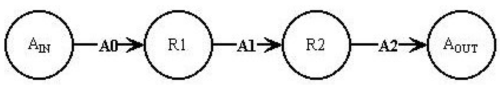
The modeling of the resources control, according to Santos Filho (2000a), can be accomplished through the use of:

- Resource Allocation Graph (RAG) and
- Enhanced Mark Flow Graph (E-MFG).

3.3.1 Resource Allocation Graph (RAG)

Resource Allocation Graph (RAG) is an unmarked graph deduced from Place/Transition Petri Nets. A RAG represents the production processes through the resources. The graph consists of (fig. 2a):

- Nodes, that represent resources of the system, depicted by circles;
- Arcs, that represent the stages of the processes, depicted as oriented arcs.



a)

$$G = \begin{array}{c|cccc} & R1 & R2 & A_{IN} & A_{OUT} \\ \hline R1 & 0 & A1 & 0 & 0 \\ R2 & 0 & 0 & 0 & A2 \\ A_{IN} & A0 & 0 & 0 & 0 \\ A_{OUT} & 0 & 0 & 0 & 0 \end{array}$$

b)

Figure 2 Example of RAG and adjacency matrix

The representation of a RAG is made through the adjacency matrix, be G a RAG with dimension n. (fig. 2b). RAG is used to determine the condition of circular wait which, in this work, corresponds to the Circular Wait Loops (CWL).

3.3.2 Circular Wait Loops (CWL)

The condition of circular wait can occur in FPSs where there is sharing of resources (Nakamoto et all, 2001a; Nakamoto, 2001b). This condition of circular wait in the model RAG is called Circular Wait Loops (CWL). The figure 3 presents an example.

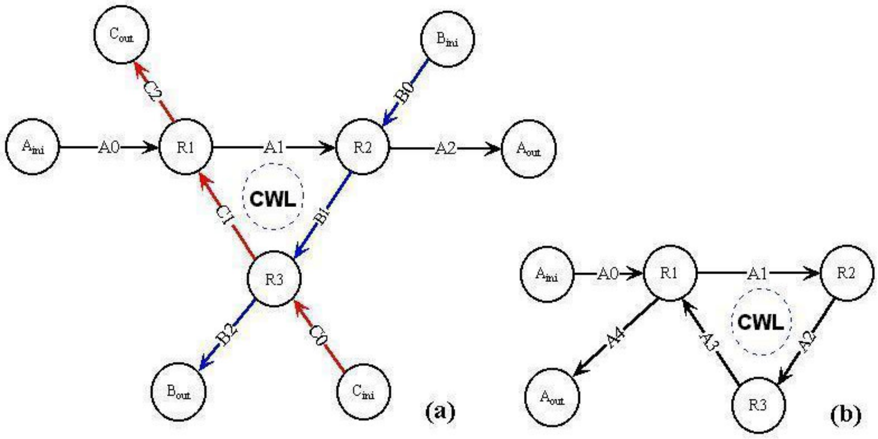


Figure 3 The example of a CWL.

3.3.3 Enhanced Mark Flow Graph (E-MFG)

The E-MFG is an extension of MFG. The MFG is a graph that represents the necessary fundamental characteristics for the structured representation of complex sequential systems, considering the functional behavior of the system and the accomplishment of the control (Miyagi, 1996; Santos Filho, 2000a). It is a subclass of nets deduced from Petri Nets (Peterson, 1981; Reisig, 1985; Miyagi, 1996; Murata, 1998). The characteristic of MFG is that the system can be easily interpreted by observing the model. However, this capacity becomes limited when MFG is used to model larger manufacturing systems, which need a more elaborated representation of control strategies. To overcome this limitation, Santos Filho (2000a) proposes an extension to MFG with the goal of increasing the modeling capacity to the tool.

The E-MFG is an interpreted graph with individual mark (Santos Filho, 2000a), and it was introduced to design system control and execute it. The difference between MFG and E-MFG is that the latter has an individuality and composition of the marks and additional rules of control that enable the transitions. The E-MFG enables a more adequate and consistent description of the characteristics of production system driven by events. In figure 4 is present the graphic elements of E-MFG.

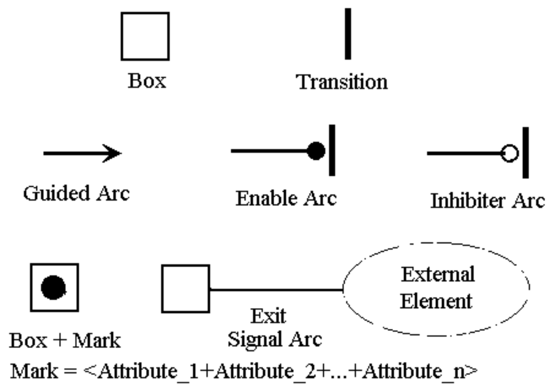


Figure 4. The graphic elements of E-MFG.

3.4 Synthesis of the Method of Resource Control

Given a set of resources and the sequences of utilization of the resources by the processes, it is presented a synthesis of the method of resource control. The steps of this method are:

- Obtaining the individual RAG for each process;
- Obtaining the global RAG ;
- Determining the circular wait loops (CWLs) in the global RAG ;
- Generating the control rules for resources allocation based on CWLs information;
- Mapping the E-MFG from the RAG (Santos Filho, 2000b) and inserting the additional rules of control.

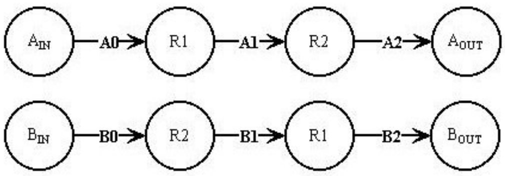
4. Algotithmical Approach

This section introduces the algorithmical approach. The algorithmical approach is applied for the first four steps of the method to the generation of the control of resources introduced previously (Nakamoto, 2002b).

4.1 Adjacency Matrix Representation of RAG

The sum of the resources types quantity and quantity of processes, in which each process owns an entrance and exit, define the dimension of the adjacency matrix.

For instance, a certain process *A* and process *B* (fig.5a). Process *A* uses two sequential resources: *R1* and *R2*; Process *B* uses two sequential resources: *R2* and *R1*. There is an entrance (A_{IN} and B_{IN}), an exit (A_{OUT} and B_{OUT}) for each process. Thus, the complete sequence for process *A* is: A_{IN}, *R1*, *R2* and A_{OUT}; and three stages: *A.0*, *A.1* and *A.2*. Thus, it creates a matrix for each process is defined. Then, the matrix of the global RAG is obtained through the coalition of the common nodes of individual processes (fig. 5b).



a)

$$G = \begin{array}{c} \\ R1 \\ R2 \\ A_{IN} \\ A_{OUT} \\ B_{IN} \\ B_{OUT} \end{array} \begin{array}{c} \begin{array}{cccccc} R1 & R2 & A_{IN} & A_{OUT} & B_{IN} & B_{OUT} \end{array} \\ \begin{bmatrix} 0 & A.1 & 0 & 0 & 0 & B.2 \\ B.1 & 0 & A.2 & 0 & 0 & 0 \\ A.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

b)

Figure 5 Individual RAG of process (a) and matrix of global RAG (b).

4.2 The E-MFG Matrix Representation

The E-MFG matrix representation is derived from the adjacency matrix of RAG. This matrix representation of the E-MFG is necessary to determine the additional rules to avoid *Deadlock State* in FPSs, which will be introduced in the next item.

The total numbers of stages (of all processes) correspond to the total quantity of lines of the matrix.

The dimension of the adjacency matrix RAG is the total quantity of columns of the matrix, that is, the sum of the quantity of resources type and the quantity of the processes. Each element of matrix E-MFG cells receive values 0, 1 or -1 according to RAG on illustrated in figure 6 and 7.

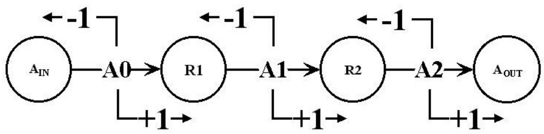


Figure 6 Determination of the values of the cells.

$$E = \begin{array}{c} \\ A.0 \\ A.1 \\ A.2 \\ B.0 \\ B.1 \\ B.2 \end{array} \begin{array}{c} \begin{array}{cccccc} R1 & R2 & A_{IN} & A_{OUT} & B_{IN} & B_{OUT} \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Figure 7 The E-MFG matrix representation.

4.3 Determination of the CWLs

The determination of the CWLs is based on search algorithm. This algorithm is an adaptation of the depth-first search algorithm (Aho et all, 1974), applied to data search in a tree structure. This algorithm, presented in figure 8, takes each node of the graph as a root of a tree and starts a depth first search. Each node that is visited is added to a list. If it visits a node that has been previously found, then a CWL has been discovered, and this is stored in a List. When all nodes of the RAG matrix have been searched, it returns to the previous node. The algorithm is finished when it returns to the root.

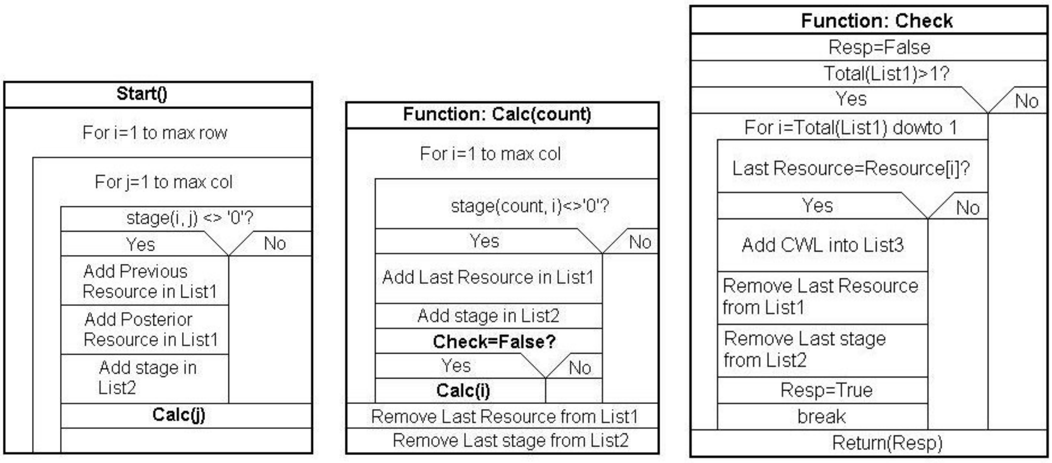


Figure 8 Algorithm for CWLs determination.

4.4 Deduction of the Additional Rules of Control to Avoid the *Deadlock*

For each CWL, based on flow-in-suppression *deadlock* avoidance policies, it is necessary to evaluate the entrance nodes for the considered cycle. The *pre-deadlock* condition is characterized when all resources, except one, of CWL are occupied by processes (processes which belong to CWL). The rules are based on the entrance inhibit of the process, that is, the resource can not be used because the system will be able to evolve to *Deadlock State*. Consider the example of figure 9. If the resource *R5* is being used by process *E* and resource *R3* is being used by process *D*, then the entrance of process *C* in resource *R6* must be inhibits.

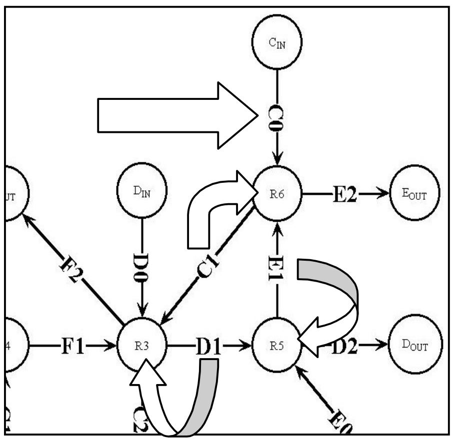


Figure 9 Example to determine the rule to avoid *deadlock*.

The algorithm creates "IF... THEN..." type production rules. The condition "IF" describes the *pre-deadlock* conditions. The action "THEN" prescribes the action of the entrance of new processes in the cycle. The figure 10 presents the algorithm to deduce the additional control rules. The CWLs list obtained at the previous step is used as an input parameter of this algorithm.

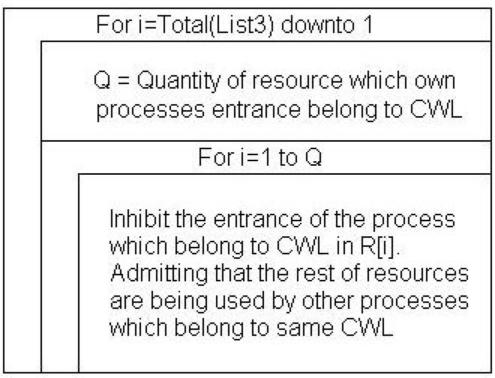


Figure10 Algorithm for deduction of the additional control rules to avoid *deadlock*.

5. Example

Consider the following system (Fanti et all, 1995): processes (*A*, *B*, *C* and *D*) and resources (*R1*, *R2*, *R3*, *R4*, *R5*, *R6*, *R7*, *R8* AND *R9*). The resources utilization sequence by the processes is presented in the table 1. These informations are inserted in the computational tool, which generates automatically the adjacency matrix RAG, the rules to avoid *deadlock* and the E-MFG matrix representation.

Table 1 Resource utilization sequence by the processes.

Process	Resource Utilization Sequence
A	R1, R8, R2, R3, R4 and R9
B	R5, R4, R3, R6 and R9
C	R5, R6, R3, R7 and R9
D	R7, R3, R2, R1, R8 and R9

Step 1 and 2: Obtaining the RAG

It is determined the global RAG based on the individual RAG (fig.11a and fig.11b).

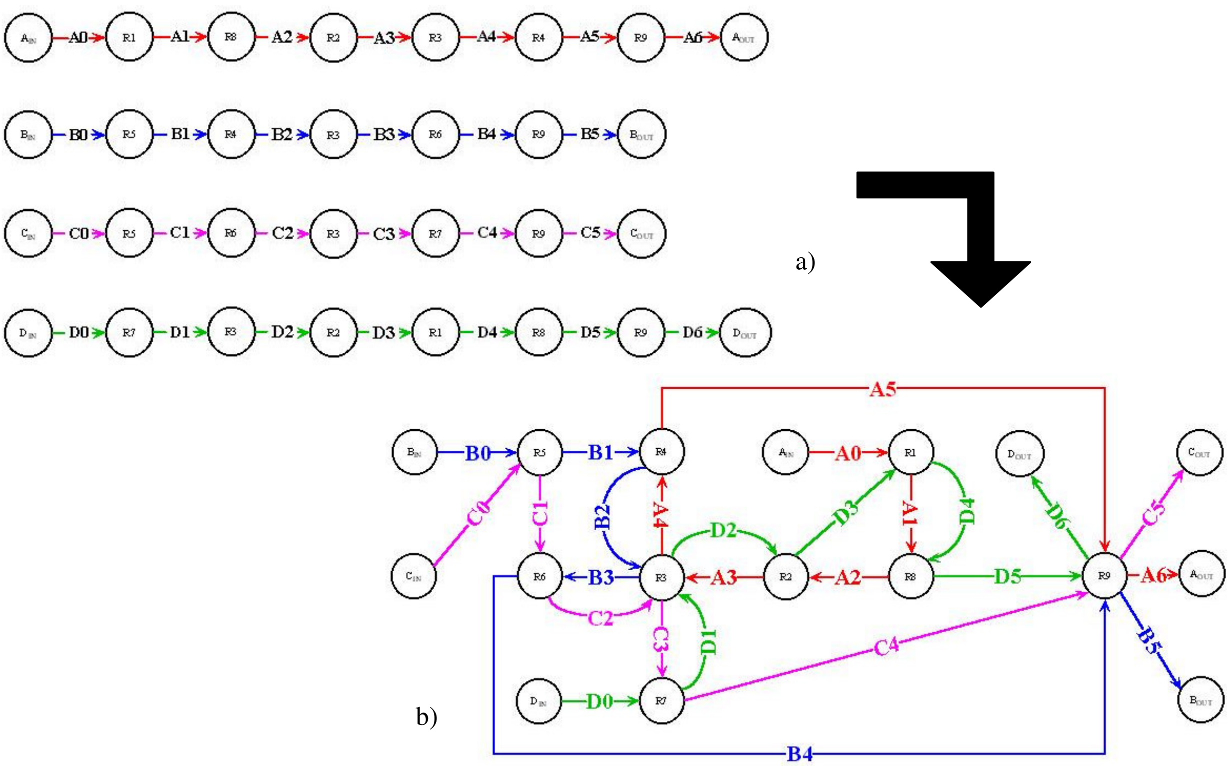


Figure 11 Individual RAG (a), global RAG (b).

Step 3: Determining the CWLs

The determination algorithm of CWL (fig. 8) is executed in the global RAG (matrix representation) and the figure 12 show determined CWLs.

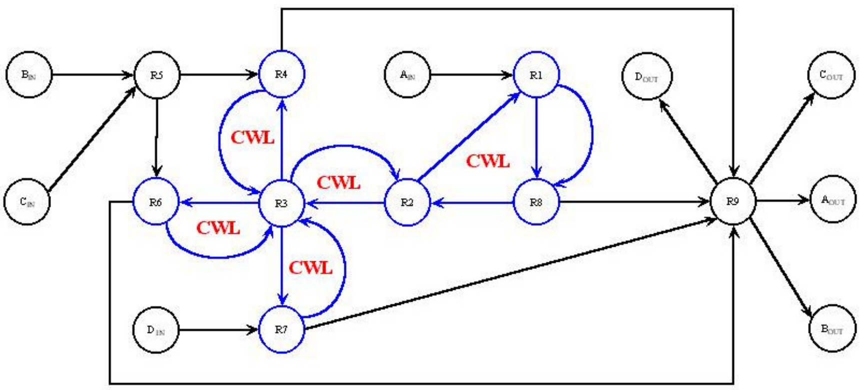


Figure 12 the CWLs determined by the algorithm (b).

Step 4: Generating the Control Rules to Avoid D*eadlock*

The rules to avoid the *deadlock* are deduced by the algorithm introduced in the figure 8. Table 2 presents the CWLs and the respective rules to avoid the *deadlock.*

Table 2 CWL and rules do avoid deadlock.

CWL	Rules to avoid *deadlock* state
D2/A3	IF (R2=A) THEN D1=FALSE
	IF (R3=D) THEN A2=FALSE
B2/A4	IF (R3=A) THEN B1=FALSE
	IF (R4=B) THEN A3=FALSE
C2/B3	IF (R3=B) THEN C1=FALSE
	IF (R6=C) THEN B2=FALSE
D1/C3	IF (R3=C) THEN D0=FALSE
	IF (R7=D) THEN C2=FALSE
D3/A2/A1+D4	IF (R8=A) AND ((R1=A) OR (R1=D)) THEN D2=FALSE
	IF (R2=D) AND (R8=A) THEN A0=FALSE

Step 5:Mapping the E-MFG From RAG and Inserting the Additional Rules of Control.

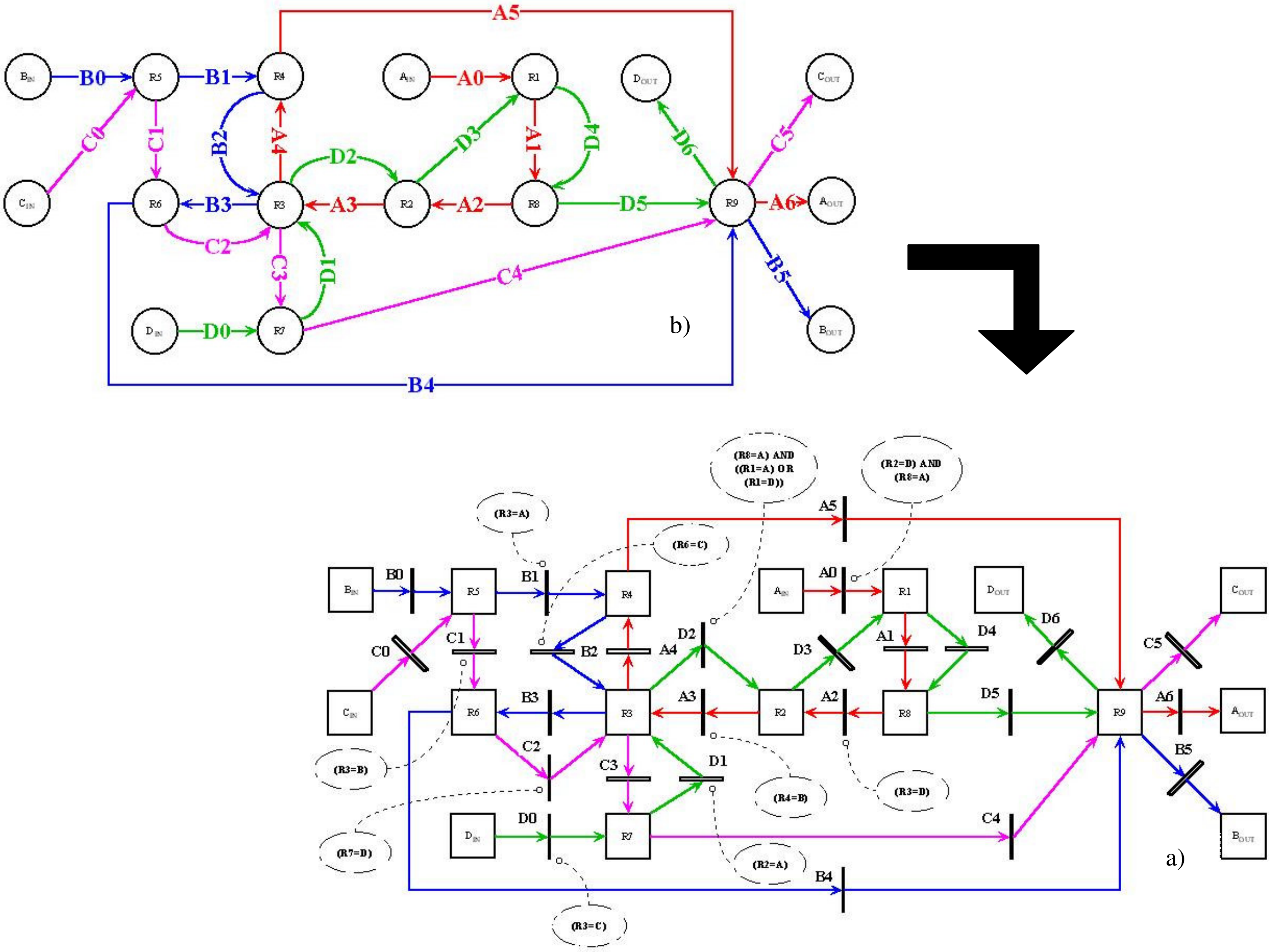


Figure 13 The mapping the E-MFG with rules (a) from RAG (b).

The Mapping is accomplished inserting the rules in the transitions of the E-MFG (Santos Filho, 2000a; Nakamoto, 2002b). The figure 14 show a detail of CWL formed by the resource R1, R2 and R8, and the additional rules to avoid *deadlock*. The transition is inhibited if the condition of the rule is true.

More examples detailed of the application of the method are introduced In Nakamoto et all (2001a; 2001b; 2002a) and Nakamoto (2002b). In Nakamoto et all. (2001a, 2001b) was introduced an initial algorithmical approach for the elaboration of the computational tool. In Nakamoto et all (2002a) was proposed a new algorithm for determination of

CWL and it was accomplished a comparison of the original and proposed algorithms. In Nakamoto (2002b) was introduced the last version of the computational tool

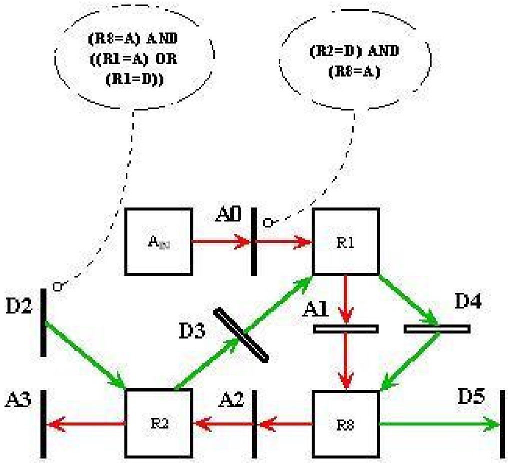


Figure 14 Detail of CWL.

6. Conclusion

The results of this work show that it is possible to automatically generate the control algorithm in a simple and fast way. In other words, it is just necessary to insert the information about the resources and the sequence of resource utilization.

Based on this algorithmical approach, it was implemented a computational tool that will integrate a computer tool for design of control system of FPS. The main contributions of this work are:

- Systematization of resources control design for FPSs;
- Automatic generation of the additional *deadlock-avoidance* rules of resource control in FPSs;
- Possibility to obtain reactive control systems, once the automatic generation of control algorithms can be accomplished in run time.

7. Acknowledgements

The authors would like to acknowledge CAPES and CNPq for the support to this work.

8. References

Aho, A.V., Hopcroft, J.E., Ullman, J.D., 1974, 'The Design and Analysis of Computer Algorithms", Addison-Wesley, Reading, Massachusetts, pp.172-251.

Banaszak, Z.A. and Krogh, B.H., 1990, "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows", in IEEE Transctions on Robotics and Automation, vol.6, pp.724-734.

Calinescu, A., Efsathiou, J., Sivadan, S., Schirn, J., Huaccho Huatuci, L., 2000, "Complexity in Manufaturing: An Information Theoretic Approach", in Proceedings of the International Conference on Complex Systems and Complexity in Manufacturing, Warwick University, Warwick.

Cassandras, C.G., 1993, "Discrete Event Systems – Modeling and Performance Analysis", Richard D. Irvin, Inc. and Aksen Associetes, Incorporated Publichers, pp.1-139.

Cho, H., 1993, "An Intelligent Workstation Controller for Computer Integrated Manufacturing", Doctor dissertation, Texas A&M University, pp.83-88.

Edmonds, B., 1995, "What is Complexity? – The Philosophy of Complexitu per se with Application to some Exemples in Evolution", Symposium: The Evolution of Complexitu, University of Brussels, Breussels, Belgium.

Fanti, M.P., Maione, B., Mascolo, S., Turchiano, B., 1995, 'Control Policies Conciliating Deadlock Avoidance and Flexibility in FMS Resource Allocation", INRIA/IEEE Conference on Emerging Technologies and Factory Automaion, Paris, France, pp.343-351.

Fanti, M.P., Maione, B., Mascolo, S., Turchiano, B., 1997, "Event-Bases Feedback Control for Deadlock Avoidance in Flexible Production Systems", in IEEE Transaction on Robotics and Automation, vol. 13, no.3, pp.347-363.

Ho, Y.C. and Cao, X.R. 1991, "Pertubation Analysis of Discrete Event Dynamic Systems", Kluwer Academics Publisherds.

Ito, Y., 1991, "A Diserable Production Structure Lookin Toward the 21th Century – Anthropocentric Intelligence-Bases Manufacturing", XI Congresso Brasileiro de Engenharia Mecânica, São Paulo, Brazil, pp.23-32.

Kumaran, T.K., Chang, W., Cho, H., Wysk, R.A., 1994, “A Structured Approach to Deadlock Detection, Avoidance and Resolution in Flexible Manufacturing Systems”, Int.J.Prod.Res., vol.32, no.10, pp. 2361-2379.

Lawley, M.; Reveliotis, S.; Ferreira, P. 1997, “Design Guidelines for Deadlock Handling Strategies in Flexible Manufacturing Systems”, The International Journal of Flexible Manufacturing Systems, vol.9, pp.5-30.

Miyagi, P.E., 1996, “Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos”, Ed.Edgard Blucher Ltda, São Paulo, Brazil, pp.1-5, 71-148.

Murata, T., 1998, “Petri Nets: Properies, Analysis and Applications”, in Procedings of the IEEE, vol.77,no4, pp.541-580.

Nakamoto, F.Y., Maruyama, N., Miyagi, P.E., Santos Filho, D.J., 2001a,“Regras de Controle para Alocação de Recursos em Sistemas Produtivos com Processos Concorrentes”, V Simpósio Brasileiro de Automacao Inteligente, Canelas, R.S., Brazil.

Nakamoto, F.Y., Maruyama, N., Miyagi, P.E., Santos Filho, D.J., 2001b, “Geracao Automatica de Regras Adicionais de Controle para Alocação de Recursos em Sistemas Produtivos com Processos Concorrentes”, XVI Congresso Brasileiro de Engenharia Mecânica, Uberlândia, MG, Brazil.

Nakamoto, F.Y., Miyagi, P.E., Santos Filho, D.J., 2002a, “Uma Proposta de Algoritmo para a Determinação dos Ciclos Fechados de Espera em Sistemas Produtivos Flexíveis”, XVI Congresso Brasileiro de Automática, Natal, RN, Brazil.

Nakamoto, F.Y., 2002b, “Sistematização do Projeto do Controle de Sistemas Produtivos”, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.

Palazzo, L.A.M., Castilho, J.M.V., 1998, “Sistemas Complexos e Auto-Organizacao”, VI Escola Regional de Informática, Curitiba, PR, Brazil.

Peterson, J.L., 1981, “Petri Net Theory and the Modeling of Systems”, Prentice-Hall.

Reisig, W. 1985, “Petri Nets, An Introduction”, Springer-Verlag, Berlin Heidelberg.

Santos Filho, D.J., 2000a, “Aspectos do Projeto de Sistemas Produtivos”, Tese de Livre Docência, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.

Santos Filho, D.J., Motohashi Matsusaki, C.T., Nelli Silva, E.C., Miyagi, P.E., 2000b, “Automatic Generation of Metamodels to Design Production Systems Control”, in Proceeding of Mechatronics 2000. 7th Mechatronics Forum International Conference, Atlanta.

Viswanadham, N., Narahari, Y., Johnson, T.L., 1990, “Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models”, in IEEE Transactions on Robotics and Automation, vol.6, no6, pp.713-723.