ISSN 2229-5518

Operating modes of automated production systems: problems and assistance tool.

MABROUK Mohamed

Abstract— Competitiveness and production flexibility in terms lead to production systems more and more automated and efficient. This causes a higher level of flexibility for the operation of these systems and consequently the limits of the operator appear quickly before these complex and heterogeneous processes. The problems related to the Production Automated Systems operation (PAS) are apparent during phases of use and maintenance of these systems. Most notable are the problems associated with incompatible operating modes, including the reconfiguration of PAS that can intervene in case of a failure in the system, change production, recovery or reintegration. Faced with these problems, it is necessary to develop methods and tools for reconfiguration of PAS. It is our goal through this paper.

Index Terms— automated production system; Life Cycle; Operating phase; Operating mode; reconfiguration; HOOD; Petri nets; H-Expert.

1 INTRODUCTION

Competitiveness and production flexibility in terms lead to production systems more and more automated and efficient. This causes a higher level of flexibility for the operation of these systems and consequently, the operators limits appear quickly before these complex and heterogeneous processes.

In fact, the life of a Production Automated Systems (PAS) does not end at the end of its development. The operating phase is the final goal of any design. Its role is to live the production automated system. Its main objectives are to manufacture in quantity, quality and on time. In reality, the normal state of the system where production was uneventful is not the one and other states appear, against which the operator must react in an effective manner respecting human an hardware security constraints. Problems related to the operational phase are found during the use and maintenance phases of the system and require the development of aiding tools. To mention only the largest categories, we can consider:

• **Diagnostic problems:** which were the subject of several research studies [1][2][3]. These led to the implementation of the troubleshooting support systems. They are generally designed independently of the control system. Therefore, they are still inadequate. An integrated monitoring control is necessary.

• **Problems with incompatible operating modes:** these are crucial problems, including the reconfiguration [4] that can intervene if:

- passage in degraded mode after fault,
- recovery and reintegration; it is essentially to reinstate a machine after repair, requiring a specific state of control-command system and the operative part of the PAS.

These problems are the result of several factors, the most important are:

- The non integration of the operating needs in the life cycle of the system,
- The diversity of particiants in the development of automated production systems. In fact, each participant is expressed in language appropriate to its specialty. This absence of a common language between the various project participants can only cause problems at several levels (relations between the subsystems, relations between the user and the system ...),
- Lack of training of operators that are generally low skilled,
- The poor definition of system specifications (the client is not always the future operator of the system),
- Lack of operating documentation.

MABROUK Mohamed
Higher Institute of Applied Science and Technology, Sidi Massoud 5000 Mahdia-Tunisia
Monastir University
mabmohamed@gmail.com

2 RECONFIGURATION OF PAS: CONCEPT AND APPROACH

The reconfiguration of a PAS consists in a modification of its dynamic behavior (mode) and/or its operating mode so that it maintains the desired performance. The concept of reconfiguration can be summarized as follows:

"Given the current state Ea an automated production system, and following a parts flow reconfiguration order or the occurrence of a failure, what are the procedures (manual and / or automatic) to apply and what are the intermediate states through which it is necessary to transit the monitoring / control system and the operative part of an PAS to position it in an objective state Eo ".

All these modifications to be made following a defined protocol that provides a reconfiguration scenario. Each scenario is characterized by:

– Reconfiguration of operating modes: concerning the reconfiguration of the operating modes of hardware and software entities of the PAS.
– Reconfiguration of command level: concerning the reconfiguration of different control softwares of PAS.

2.1 An approach to reconfiguring operating modes.

The basic idea of our methodology is as follows: " All states of operation modes characterize a SAP entity can be modeled by a directed graph of states G. As the reconfiguration of an entity consists to evolve a ei operating mode state to another state ej, then this is equivalent to navigate a path "mode" in G having as initial state ei and as final state ej. This reconfiguration will be called "potential reconfiguration".

Therefore, the reconfiguration of PAS can be considered as a set of potential reconfigurations constraints between them. This means that reconfigure SAP consist to browse a set of "Operating Mode" paths constraints between them. Each path belongs to a Gi state graph associated to The entity Ci of PAS.

In the following, we characterize the state of the operating modes E of PAS by the N_uplet (e1, e2, ..., ei, ... en) of states of operating modes associated entities. The initial state (or current) and the final state (or objective), characterizing a reconfiguration of the system, are designated respectively by Ea = (Ea1, Ea2, ..., Eai, ..., Ean) and Eo = (Eo1 , Eo2, ..., Eoi, ... Eon). These two states are assumed to be known in the process.

Define all scenarioes that allow SAP to evolve from Ea state towards Eo state consist to define $Sr_{n(r-1)m}$ nodes of a tree structure as shown in Fig 1. The tree has as a starting node Ea and as happening node Eo.

This structure is divided into ranks. Each rank contains a set of operating mode states that PAS can take. Ranks (0) and rank (n) contain respectively the current state Ea, and objective state Eo .

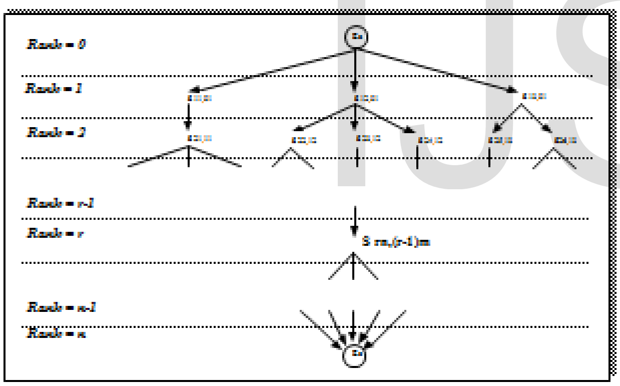

Fig 1: Tree structure

Each intermediate node $S_{rn,(r-1)m}$ characterizes a state of PAS operating modes state. It is defined by:
- An N_uplets of states of potential operating modes (e1, e2, ..., ei, .., en),
- the rank (r) of the state, characterizing the order of the state in the scenario,
- The number (n) in the rank,
- The number (m) and rank (r-1) of its predecessor.

A scenario corresponds to a path belonging to the tree having as starting node Ea and as arrival node Eo. It is described by a set of PAS states. Each state belongs to a rank (r) of the decomposition.

To enable the construction of the tree, the following three hypotheses are retained in our approach:
- only stable states are considered in our study. A stable state is a controllable state from which recovery is possible.
- potential targets states must be compatible. If is not the case, then the requested reconfiguration is questioned.
- current potential starting states are considered compatible. If not, then a pre-reconfiguration phase is needed to make compatible the starting states of operating modes. the establishment of the pre-reconfiguration is part of the operator's experience. For example, if a robot fails during the machine loading, the pre-reconfiguration consists of a clearance, in manual mode, the robot out of the machine.

The approach, we propose, consists of three steps:
- Specification of the operating modes states of the system entities,
- Specifying constraints between operating modes states,
- Specifying reconfiguration scenarioes of operating modes,

a) Specification of the operating modes states of the system entities

The purpose of this first step is to specify the operating mode states of system entities. This specification is established from the document design and use of entities. The state graph is a formal tool used for this specification. To each system entity is associated with a Gi state graph representing its different operating modes.

b) Specifying constraints between operating modes states

The objective of this second step is to specify constraints between operating modes states of each pair (Ci, Cj) of system entities and to derive their compatibilities. Two or more states of operation modes are said to be compatible if they can persist at the same time in the production system. Otherwise they are called incompatible.

Constraints are links that exist between the entities of the production system, and are introduced by the imperatives of advanced automation of production. They are mainly linked to the security aspect (Machine-machine and/or Human-Machine) and quality of the product. These constraints are used to define and specify multiple dependencies that exist between the entities and their tasks. Accordingly, they introduce an implicit relationship compatibility and incompatibility between these states. The main types of constraints are:
- structural constraints [5]
- functional or operating constraints [6], and
- malfunction constraints [7]),
- security constraints.

A matrix representation (Fig 2) [8] [9] [10] is used to specify constraints between operating modes of the system entities.

This type of modeling is retained thanks to its richness of expression and implementation simplicity.

So, we associate to each pair (Ci, Cj) of system entities a matrix of compatibility MCij. The elements MCij[e_{ik}, e_{jl}] of this matrix are boolean type defined by MCij [e_{ik}, e_{jl}] = 1 if the two operating mode states e_{ik} and e_{jl} respectively of Ci and Cj are compatible, else, MCij [e_{ik}, e_{jl}] = 0.

These matrices are used in the rest of our approach as a compatibility control tools between the operating modes states entities. The criterion "*couple of entities*" is retained in order to

ISSN 2229-5518

facilitate the matrices manipulation. In against part, the number of manipulated matrices can be high.

Operating modes of Cj component

Operating modes of Ci component

Ci \ Cj	ejl		ejk		ejn
eil					
eil			MCij [ejl,eilk] = 0 ou 1		
eip					

Fig 2: Compatibility matrix between the operating modes states of entities Ci and Cj.

c) Specification of reconfiguration scenarioes of operating modes

The purpose of this step is to specify the sequences of N_uplets (E1, E2, .., Ei, .., En) of operating modes states that the system must be taken to achieve the objective state Eo regardless, in a first phase, associated actions. It is therefore to specify the tree structure through the identification of $S_{rn,(r-1)m}$ nodes associates. To do this, the state graphs of system entities and compatibility matrices between operating modes states of these entities are used for the specification of the tree. This stage is divided into to four phases:

Phase 1: Determinating the paths "operating modes".
This phase consists of research, for each system entity, all paths "operating modes" allowing it to move from its current potential state Eai towards potential state objective Eoi. These paths can be a priori considered as potential reconfiguration scenarioes associated with the entity Ci. The goal through the search paths is to depart from the approach the statements of operating modes from which no potential objective can not be achieved.

Phase 2: Determination of the possible states of operating modes.
This phase involves researching for each Ci entity EPo (Ci, e_k, D_rang + 1) set of e_i states ($e_i \in$ D_rang + 1) it can, in principle, be taken immediately if it is in the e_k ($e_k \in$D_rang and $e_k \neq$ Eoi) state. These states (e_i) are said *"possible states of operation modes from e_k"*.

In fact, the possible states from ek are its successors in owned paths associated to Ci (specified in the previous phase1). In the case where e_k is the objective potential state Eoi of the entity Ci then the possible state from e_k is itself. This means that the entity Ci has achieved its reconfiguration objective potential state.

The EPo(Ci, e_k, D_rang + 1) states found in this second phase are states of operating modes that can take The entity if it were in the e_k operating mode state. But the existence of constraints between operating mode states of the system entities, implies that only certain states of EPo (Ci, e_k, D_rang + 1) may be authorized at any given time and depend on the operating modes states of other entities. These states are called "*permitted states of operating modes*".

Phase 3: Search permitted states of operating modes.
This phase involves research, EPo (Ci, e_k, D_rang + 1) of all authorized state EPe (Ci, e_k, D_rang + 1) modes of operation from the state e_k. control compatibility of each *possible state of operation mode* el EPo (Ci, e_k, D_rang + 1) with states permitting use patterns Cj entities (i ≠ j) belonging to D_rang must be achieved. These controls are established using the compatibility matrices. Several results can arise from these checks. Most features are:

- A set EPe (Ci, e_k, D_rang + 1) associated with an entity Ci is equal to the empty set: this means that no possible state from e_k and belonging to the rank (D_rang + 1) is permitted for the Ci entity. in this case, Ci must retain its last permitted state belonging Epe (Ci, e_k, D_rang).
- All EPe(Ci, e_k, D_rang + 1) sets associated with the entities of the system are empty and the system has not achieved its objective state Eo = (Eo1, Eo2, ..Eoi .. EoN). Then, two cases can be considered:
  - the requested reconfiguration is not possible: In this case, we must define another objective state for the system. This new target must be set by driving function through a redefinition of the modifications to be made on the flow of parts. Once defined this objective, the reconfiguration process should be resumed from the third step.
  - It is no longer possible to change one potential state operating mode. In this case, check whether it is possible to change simultaneously (in parallel) some states of potential modes of operation of certain system entities. These statements, if any, are entitled "*states allowed parallel order r*" (2 ≤ r ≤ n with n is entities number).

Phase 4: Determination of parallel allowed states order r.

This phase consists, therefore, in search of all r_uplets of the operating mode enabled parallel states of order r. We will check the compatibility between the possible states of operating modes EPo (Ci, e_k, D_rang + 1) of each entity Ci with all possible states of operating modes EPo (C, e_k, D_rang + 1) of each Cj (≠Ci) entity of system. We denote the set of *states allowed parallel order r* bearing the rank (D_rang + 1) by EPp (r, D_rang + 1).

the set of states allowed parallel order r bearing the rank (D_rang + 1) associated with the entity Ci when the state permit mode operating ek owned D_rang is designated by EPp (r , Ci, e_k, D_rang + 1).

The states allowed parallel should be synchronized when changing operating modes. This means that entities associated with the r_uplets potential states constituting the state allowed parallel in question, take their states belonging at r_uplets to the same instant.

2.2 An approach for the control level reconfiguration.

At the command level, the state graph, modeling the operating modes of an entity of PAS, is seen as an operating program. To Each state is assigned a program module which may be represented by the Petri nets [11] (Fig 3 for example). We denote "*program module*" a part of the operating program that can be run by itself. The timed Petri nets constitue the formal tool used for modeling the control command. The timing is established on network states.

Each module can transmit and/or receive a message to/from another module of a another system entity (for example, two robots exchange messages during an assembly operation). It can also receive a report of a sensor (position of the part for example). In both cases, the module may be subject to blockage. Reconfiguring the control level is therefore in resolving deadlocks with synchronization points [12].
Our proposal is the use of modules entitled "*recovery Module* ". Petri nets are the tools used for the specification of these modules. Fig 3 shows an example of recovery module.

Each recovery module is associated for receptivity of synchronization in the the behavior model of material entity and / or software from PAS. It must describe the behavior of the entity if the corresponding point synchronization is blocked.

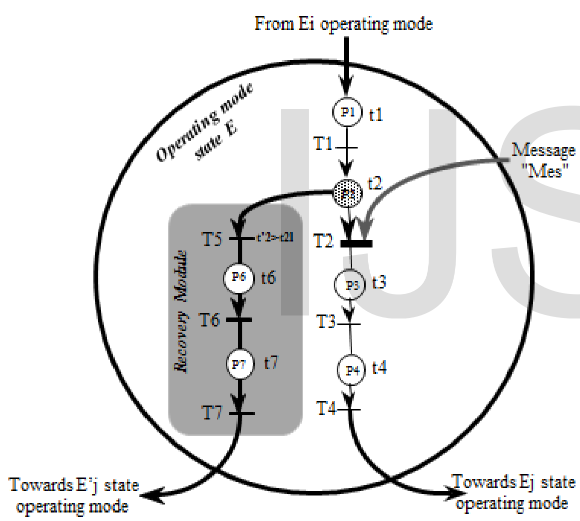

Fig 3: example of control level reconfiguration associated to the reconfiguring of the operating modes (passage from Ei state to E'j state)

3 An Assistance Tool for the Reconfiguration of PAS

3.1 Functional architecture.

The various functions characterizing the operational phase of a PAS can be organized in the form of the general structure shown Fig4. This structure is based on three levels:

- A "**USER**" level for the consideration of the various stakeholders for modeling and editing templates (generic and/or special) of a PAS and involvement of operators.

- A "**FUNCTIONAL**" level to the development of specific models to an application (PAS) for the reconfiguration, modification and operation of these models. This level should support the functions to elaborating of the above mentioned model, research, evaluation and application of modifications on the operation modes for reconfigure the PAS.

- An "**INFORMATION**" level with the aim of supporting generic data structures of SAP and instantiated specific objects to an application. It must also support the rules and procedures relating to the functions of the operating and, in particular, those associated with the functions for reconfiguration of the production system.

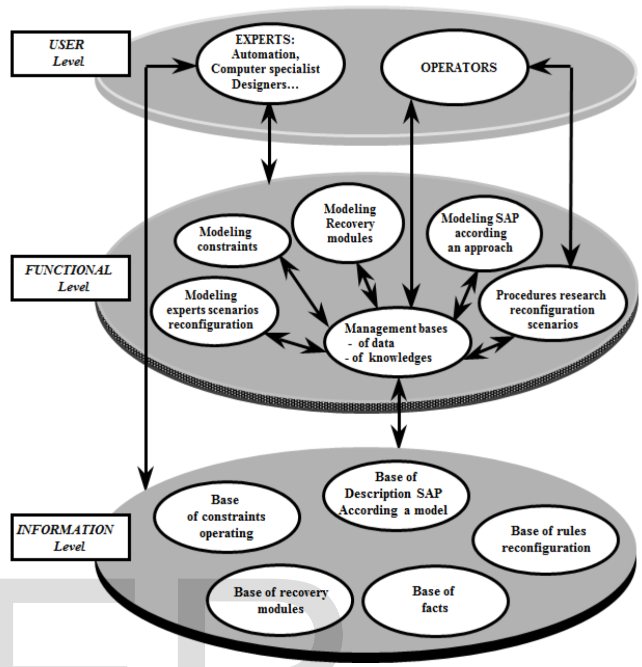

Fig 4: General structure of an assistance tool for the reconfiguration of PAS.

3.2 Internal conceptual model of the assistance tool for the reconfiguration of PAS

The model of automated production system is used in our study is the MESAP (Model of Operations of Automated Production Systems) [14].

The hierarchical object oriented design (HOOD) [15] is the method used for the description of the internal conceptual model of the proposed tool for reconfiguration. As presented fig5, besides the objects MESAP, two other classes of objects are highlighted: the class "scenario" that includes several other subclasses, and the class "constraint". These two classes are included in the class "Station" of MESAP.

3.3 Implementation

The implementation of the assistance tool for the reconfiguration of PAS must take into account the different levels characterizing its functional architecture presented below. It must therefore enable the creation and the operating of the different models and concepts described above. Thus, we have defined the general architecture implementation of a tool for the design and reconfiguration of PAS as shown Fig 6. This architecture is based on two models:
- An **internal model**: support "information" level. It therefore contains the different classes of objects MESAP model and the assistance tool for the reconfiguration and their instantiations.

- A **conceptual model**: support "functional" and "users" level of assistance tool for the reconfiguration. It allows to act on the objects creation, modification, deletion and use.

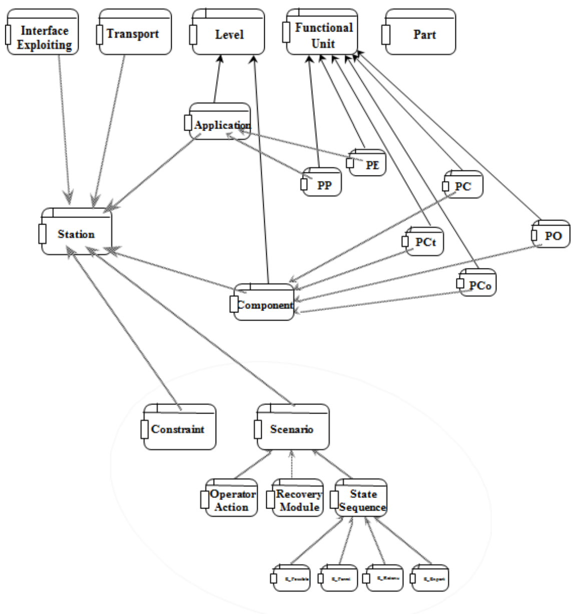

Fig 5: Internal conceptual model of the assistance tool for the reconfiguration of PAS.

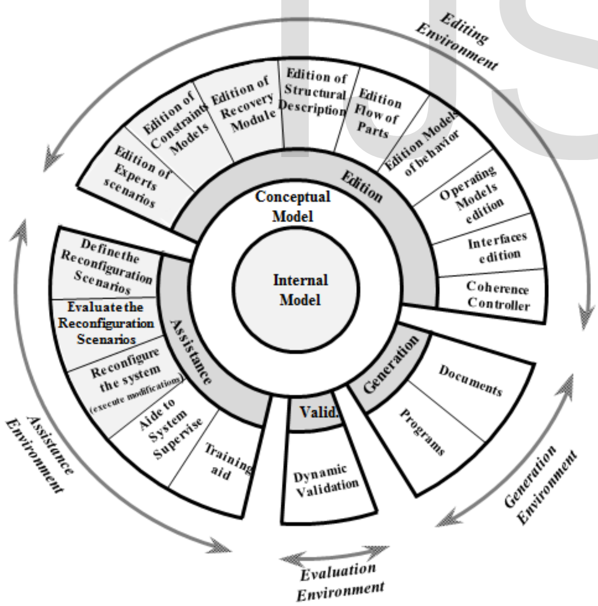

Fig 6: General architecture implementation of the assistance tool for the reconfiguration of PAS.

3.4 Validation

The validation presented here is done by simulation. For this, we developed a software prototype to:

- editing of various models of the SAP,
- Search of reconfiguration scenarios.

This latter characteristic is established by interaction with the human operator. The expert system generator H-Expert [16] is used for this achievement.

The cell used as a support of the application is that of the University of Valenciennes. The hardware architecture of the cell is shown in Fig 7. This provides the machining cell and cylinder assembly.

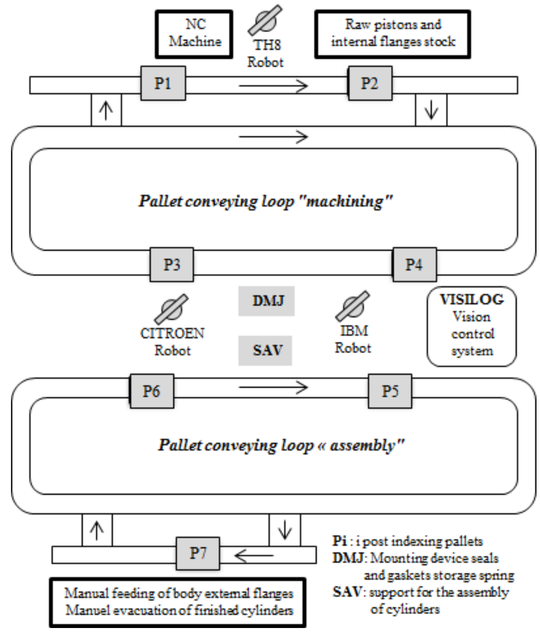

Fig 7: Hardware architecture of the cell.

The operating model of the cell according MESAP and the associated object model are shown respectively in the fig8 and fig9.

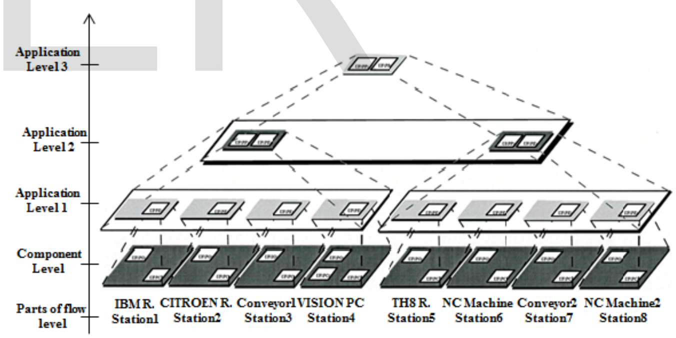

Fig 8: MESAP model of the cell.

Fig 10 presents the results obtained during the course of a search session and selecting a cell reconfiguration scenario. This reconfiguration is requested following the break of the cylinder rod during its machining. It involves work on the NC Machine lathe to clear the broken part and then resume normal production of the cell.

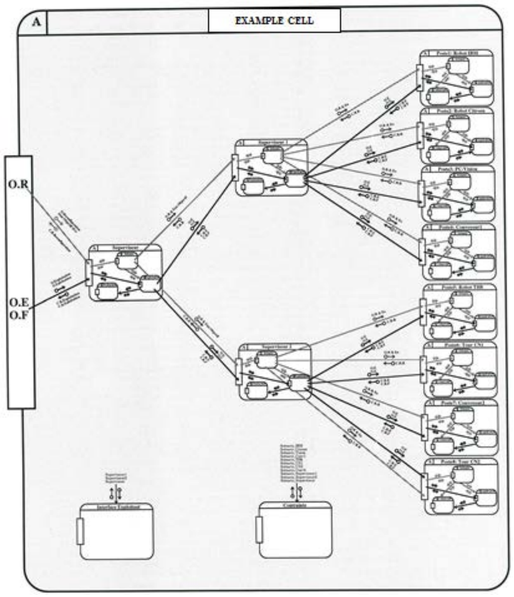

Fig 9: Cell Object Model.

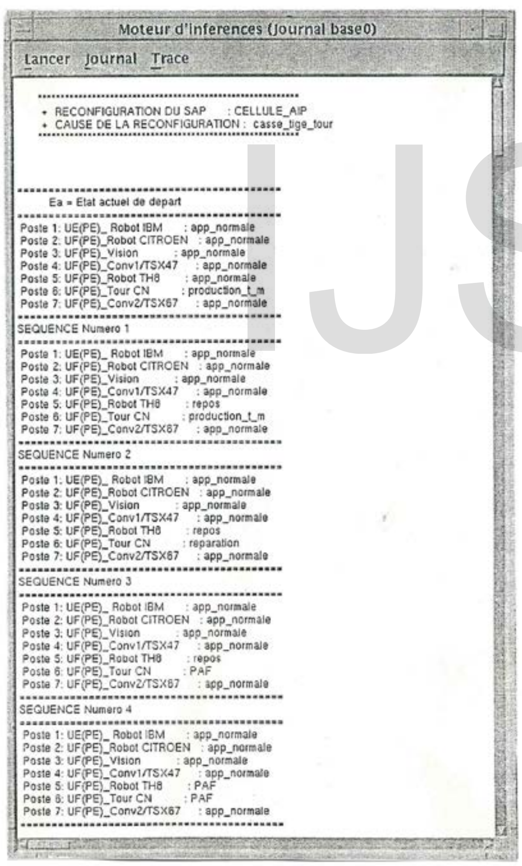

Fig 10: Cell reconfiguration scenario

4 Conclusion

In this paper, we proposed a generic method to define the set of changes to be made and the associated protocol to reconfigure an automated production system (PAS). Under this method, we specified a tool for the reconfiguration of SAP. After the presentation of its functional architecture, we have described the internal conceptual model of the aiding tool using HOOD method and its implementation architecture. To validate the proposed approach, we have developed a prototype of the support tool. This prototype has allowed us to study and simulate a case of reconfiguration associated with an automated production cell.

References

[1] Brunet J., "Reconfiguration et diagnostic prédictif de machines outils", Diagnostic Engineering Management, Quatrième congrès international sur les techniques de surveillance de diagnostic et de maintenance, Senlis- France- du 15 au 17 juillet 1992.

[2] El khattabi S., "Thèse de doctorat : "Intégration de la surveillance de bas niveau dans la conception des systèmes à événements discrêts : application aux systèmes de production flexibles". Université des Sciences et Techniques de Lille, 29 Septembre 1993.

[3] Shraoui A., Thèse de doctorat, " Contribution à la commande et la surveillance d'ateliers flexibles", Université Paul Sabatier de Toulouse (Sciences), Laboratoire d'Automatique et d'analyse des systèmes du CNRS, France, 1987

[4] Mabrouk M., "Proposition d'une méthode et d'un outil d'aide à la reconfiguration des systèmes automatisés de production," thèse de doctorat, Université de Valenciennes et du Hainaut Cambrésis, 1996.

[5] Bois S., "Thèse de doctorat : "Intégration de la gestion des modes de marche dans le pilotage d'un système automatisé de production"; Université des Sciences et Techniques de Lille Flandres Artois.

[6] Dindeleux E., Thèse de doctorat : "Proposition d'un modèle et d'un système interactif d'aide à la décision pour la conduite d'atelier", Université de Valenciennes et du hainaut-Cambrésis, 17 Janvier 1992.

[7] Rode G., " Les systèmes de production: modélisation et performance", Edition Hermès, 1989.

[8] Descanel F., Thèse de doctorat : " Pilotage d'une cellule flexible d'usinage",Faculté des Sciences et des Techniques de l'Université de Franche-conté, 17 Février 1989.

[9] Bois S., Thèse de doctorat : "Intégration de la gestion des modes de marche dans le pilotage d'un système automatisé de production"; Université des Sciences et Techniques de Lille Flandres Artois, 1991.

[10] Kermad L. and Gentina J.C., "Reconfiguration d'un système de production flexible et changement de mode de fonctionnement"; Journées d'étude S3 Sûreté, Surveillance, Supervision, Paris, 8 décembre1994.

[11] David R. and Alla H., "Du Grafcet aux réseaux de Petri",Éditions HERMES, Paris, 1992.

[12] Parayre T. " Le MESAP: vers une méthodologie d'exploitation des systèmes automatisés de production, " Université de Valenciennes et du Hainaut Cambrésis, 1992.

[13] HOOD. " HOOD reference manuel, version 3.0,1989.

[14] H-EXPERT. " H-EXPERT manuel de référence, version 1.3,1992.