# Qualitative Analysis Of Enterprise Processes With Autonomous Petri Nets

**Christine Covès, Didier Crestani and François Prunet**

Laboratoire d'Informatique de Robotique et de Microelectronique de Montpellier
Département de Robotique
161 rue Ada
34392 Montpellier Cedex 5
FRANCE
e-mail: coves@lirmm.fr

**Abstract :** Enterprise modelling is used to deal with the complexity of industrial world. However, most of the high level enterprise models cannot easily deal with formal analysis. In this paper a method based on autonomous Petri nets allowing to behaviorally analyze an enterprise process is proposed. The study of qualitative properties like liveness, boundedness and invariants, permits the detection and interpretation of some dysfunctionings of the modeled process.

**Keywords :** enterprise modelling, enterprise process, autonomous Petri nets, qualitative analysis

**Christine Covès** is a Ph.D student at the University of Montpellier, France. She develops her research activities at the LIRMM, French Laboratory of Robotics, Micro-electronics and Computer Science. Her areas of interest are the discrete event systems modelling and analysis.

**Didier Crestani** is Assistant Professor at the University of Montpellier, France. He develops his research activities at the LIRMM, French Laboratory of Robotics, Micro-electronics and Computer Science. His areas of interest are enterprise modelling, discrete event systems modelling and analysis by using Petri nets.

**François Prunet** is Professor at the University of Montpellier, France. He develops his research activities at the LIRMM, French Laboratory of Robotics, Micro-electronics and Computer Science. His areas of interest are the discrete event systems modelling and analysis.

## 1. Introduction

From early in the 20ies, when taylorism appeared, on many methods aiming at improving the behaviour of an enterprise have been proposed. In the 60ies, the first methods of management like MIS or 5M have been developed. The MIS (Management Information System) method allows to elaborate management strategies for banks or financial institutions. The 5M theory ("men", "money", "machines", "methods", "markets") proposes to represent the enterprise functioning according to these five factors "M". In the 70ies, methods such as the BBZ (Budget Base Zero) dealing with the reduction of enterprise structures, or the PDCA ("Plan, Do, Check, Act") method for a quality approach have been applied. Finally, in the 90ies, due to the emergence of new technologies for information and communication, new organisation structures have been imagined. This technological eruption breaks out the traditional planning structures of processes inside the enterprise [1].

Nowadays, the enterprise is inevitably subject to competition. So the improvement of the quality, the flexibility and the productivity of its processes and the conformity to ISO standards turn essential in order to offer the best products and services at the best prices within the shortest time schedule. Effectively, for every manufacturing or services enterprise, it is in the research on the value and quality that the customers may establish the rules of acceptability of the products and services. These customers reward then the efficiency and the performance of enterprises which provide them with the best products and services at competitive prices and delays. Along the years, the need for quality at the product level has extended to the need for quality at the enterprise process level [2].

The costs of a lack of quality, flexibility and productivity in an enterprise urge the finding of possibilities for an improvement of the processes which compose it. So, a better understanding and evaluation of the enterprise functioning is needed to anticipate or minimise the different problems which may come up.

Process improvement requires to be able to describe in detail the process organisation and all the entities involved in its functioning. From this description an analysis must proceed to improving the process efficiency.

In order to describe such systems, a number of enterprise models has been proposed [3]. Although these models have in general a good description power, they cannot be used to realise a fine analysis of the modeled processes. So, it is necessary to enrich the enterprise models by using formal models more adaptable to an in depth analysis but less easy to handle for non specialists. A solution is to automatically translate the enterprise model into the formal model adapted to the expected analysis. Here, the results of a study based on such an approach are presented.

Firstly, the context and goals of the study and the proposed methodology are presented. Then the enterprise model MOVES [4] is briefly described. Finally, the analysis possibilities of autonomous Petri nets [5], which have been adopted as formal models are shown. Each step of the approach is detailed. The translation of each enterprise model entity into an associated Petri net and their connections are described. Then, all the phases of analysis are presented. Before concluding, the approach of processes analysis is illustrated by a simplified industrial example.

## 2. The Proposed Methodology

This work has been based on the study of a quality handbook describing all the processes of an industrial enterprise. The quality handbook is a manual which has been worked out by production engineers and which is requested for an ISO9002 certification.

From a representation of the enterprise functioning, the proposed methodology suggests the following steps :

- firstly, the user describes the process by using an enterprise model from his own point of view;
- once the process is modeled, it is automatically transformed into a similar one, using the formal model tailored to the analysis to be done ;
- the analysis is realized, and the studied properties are interpreted to extract information about potential modeling or process design errors ;
- finally, this information reverts to the user who can modify or precise his modeling.

This sequence is repeated until the modeled process is error free. Once this goal is reached, all the realized modifications must be applied on the real process. The proposed methodology is illustrated in Figure 1 and is explained in more detail in Section 6.

## 3. The Enterprise Model Used

### 3.1 Enterprise Models

Over the last twenty years, several approaches for enterprise modeling have been developed. Thus, the enterprise model IDEF (ICAM Definition Methodology) originates from the program ICAM (Integrated Computer Aided Manufacturing) started in the USA in 1978 by the United States AirForce. This model has known several versions. The first version, IDEF0, resides on the concept of activity and derives from the SADT (Structured Analysis and Design Technique) method [6]. The IDEF1 and IDEF2 later versions integrate the concept of information and the simulation aspect. Then IDEF3 allows to briefly represent the behaviour of enterprise processes. Finally, IDEF3x which permits to take into account the semi-structured processes, has been proposed in the ACNOS project [7].

In 1980, the GRAI (Graphs with Results and Interrelied Activities) method [8] has been developed for systems design in industrial automation. In this modelling method, the conceptual model of the production system is based on the representation of physical, decisional and informational entities. This is a hierarchical model which can be decomposed according to the nature of systems and activities. In 1987, the method GIM (GRAI Integrated Methodology) [9] has been proposed. This method adds the functional viewpoint to the GRAI conceptual model.

The CIMOSA (Computer Integrated Manufacturing Open System Architecture) architecture was developed in 1986 by the Consortium AMICE within the ESPRIT project

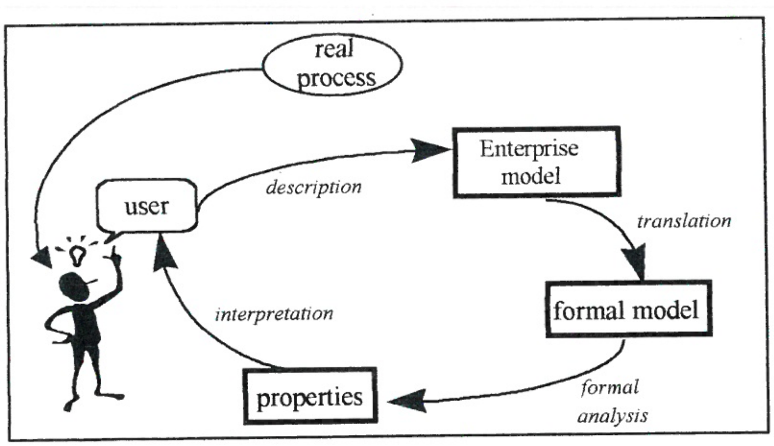


**Figure 1. The Proposed Methodology for Processes Qualitative Analysis**

in order to construct and analyse integrated systems of production [10]. This architecture consists of three fundamental components : an enterprise modelling framework, an integrating infrastructure and a CIM (Computer Integrated Manufacturing) system life cycle [11]. The architectural framework is represented by a cubic structure : the well known CIMOSA cube.

In 1989 the PERA (Purdue Enterprise Reference Architecture) architecture, inspired by the works on CIMOSA, was developed at the Purdue University (USA) [12]. The PERA architecture is represented by a layering structure where each layer corresponds to a phase task and where the set of such tasks matches with the life cycle of an industrial entity.

The ARIS (Architecture for Information Systems) architecture, developed at the Saarbrücken University in 1990 [13], can also be mentioned. ARIS, whose architecture is close to CIMOSA, focuses on software engineering, database design and on organisational aspects for integrated enterprise system design.

Finally, in 1990, in order to enlarge the possibilities of the main enterprise models, IFAC/IFIP Task Force work group [14] developed the GERAM (Generalised Enterprise Architecture and Methodology) method. GERAM is a more complete model and has been created by coupling the more representative architectures (CIMOSA, GRAI-GIM, PERA).

This allows the situation of the enterprise model MOVES (Model for the Organisation by Validation of Enterprise Structures) which, in many aspects, is close to the IDEF3 model [4].

### 3.2 The MOVES Enterprise Model

An enterprise can be seen as a complex organisation whose functioning is based on the interaction of human and material resources during operation. An enterprise model must have the capacity of translating these different components.

The MOVES enterprise model is based on the definition of four basic entities : the step, the actor, the information and the function.

The step is the entity in which the activity is executed. Then, it plays an essential part in the description of a process. It is characterised, among others, by the set of skills necessary for its good execution. The other three entities revolve around it.

The actor is the entity which allows to define any type of resource (human, material) necessary for the execution of a step. Every considered skill has its own characteristics (skills for human actors and functional capacities for material actors).

The information is the entity which allows to define all the data transiting with the environment or between steps.

The function is another entity which can be distinguished in :

- the constraint function, which allows to describe execution conditions for the associated step,
- the output function, whose role is to calculate the information at the end of an activity according to the input information,
- the added value function, which allows to associate performance criteria to the realized activities and then authorizes a fine quantitative analysis of the process.

The formalism representing the base of the enterprise model is illustrated in Figure 2.

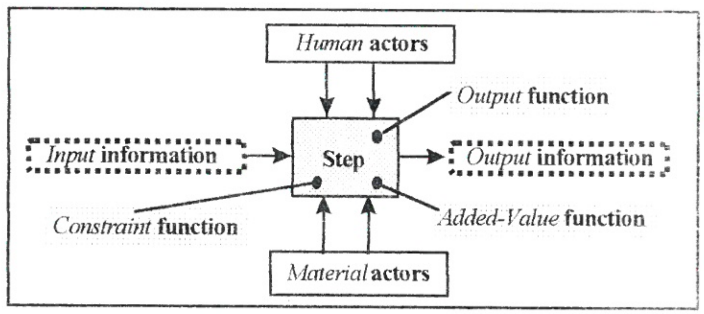


**Figure 2. Entities of the MOVES'Model**

## 4. Petri Nets : the Chosen Formal Model

### 4.1 Specification of the Formal Model

The choice of a formal model is directly oriented by the goals set out for the analysis process. In our case the aim is to realise both a structural and a behavioural analysis. This can be done by checking qualitative properties of good functioning.

Many models are able to analyse discrete event systems. Some of them like diode algebra [15], are only dedicated to behaviorally restricted systems. Some others like the Grafcet [5], easier to use, do not have sufficient analysis capacities. Finally, some models, like the ISM [16], even if very efficient for the analysis, are rather difficult to implement.

So, due to their significant power of description and validation, the choice has fallen on the autonomous Petri nets.

## 4.2 Petri Nets

Petri nets [17] allow to express the relations describing the dynamic behaviour of discrete event systems. They are associated with a graphical representation and with behavioral rules. Petri nets are composed of two types of nodes : places (symbolised by circles) and transitions (symbolised by lines). These nodes are linked by oriented arcs.

The set of places allows to represent the state of the modelled system. Events are associated with transitions. Places play the part of state variables which are linked to a marking (associated with the presence or absence of tokens in places).

The evolution rules of Petri nets give them a dynamics. These rules allow to modify the number of tokens in the Petri net when a transition is fired.

For ordinary Petri nets, the condition for a transition to be fireable is that all its upstream places must contain at least one token. The firing of a transition consists in removing a token in each upstream place of the fireable transition, and adding one in each downstream place.

The initial marking of the Petri net expresses an interpretation of the initial conditions of the modelled process.

## 4.3 Properties and Autonomous Petri Nets

Many behavioural properties and structural characteristics can be checked on autonomous Petri nets.

The behavioural analysis of a Petri net can be done with several tools such as the coverability graph, the incidence matrix and the state equation. The marking graph allows to enumerate all the states and events of the modeled system. The coverability graph allows to analyse systems with an infinite number of states by reducing the marking graph to a finite graph. This compression is obtained by using the symbol $\omega$ which expresses the infinite marking of a place. The graphs complexity can be decreased if reduction rules [18] are used on the studied Petri net.

Behavioral properties which can express some good functioning properties are associated with the behavioral analysis. Among these behavioral properties, the quasi-liveness, the liveness, the boundedness and the reversibility can be distinguished [17].

The property of quasi-liveness of a transition expresses the existence of an evolution of the system allowing the execution of a given action. A Petri net is said to be quasi-live if all its transitions are quasi-live. The quasi-liveness for a system represents the feasibility of executing at least once each one of its functions.

The property of liveness of a transition expresses the possibility of firing this transition later on. A Petri net is said to be live if it is possible to ultimately fire any transition of the net by progressing through some further firing sequences whatever the marking which has been reached to from the initial marking. The liveness for a system represents the availability of all of its functions.

The property of boundedness represents, schematically speaking, the maximal quantity of tokens which can be present in the places. A Petri net is said to be k-bounded or simply bounded if the number of tokens in each place does not exceed a finite number k for any marking reachable from the initial marking. The boundedness represents then the limit of accumulation of matter in a system.

The property of reversibility for a system expresses the possibility of reiterating the set of the accomplished functions, or more simply the possibility of coming back to the initial state of the system. A Petri net is said to be reversible if the initial marking Mo is reachable from M for any marking M of the net.

These properties depend on the initial marking of the Petri net.

Concerning the structural properties, they can be studied by using linear algebra techniques based on a structural representation of the net with matrix. The main identifiable properties are the P and T invariants. P-invariants are marking invariants which express that the sum of markings of some places of the Petri net remains constant. T-invariants characterizes cyclic firing sequences of the Petri net.
The structural properties do not depend on the net marking.

# 5. Representation of the Enterprise Model By Petri Nets

## 5.1 The Basic Entities

### 5.5.1 The Step

The Petri net associated with the step (Figure 3a) is composed of one place linked to a set of input and output transitions. The marking of this place corresponds to the execution of the associated activity. Every upstream transition expresses one execution possibility of the activity. Every downstream transition is associated with one of the possible evolutions.

The step net can be decomposed into three places and several transitions if it is more detailed. Hence, in Figure 3b, the first place $Pi_a$ corresponds to the checking on the conditions under which the activation of the step is possible. The second place is the activity itself. Finally, the third place $Pi_d$ models the condition for disactivating this step. This more detailed Petri net can be used to detect the origin of a problem more precisely.

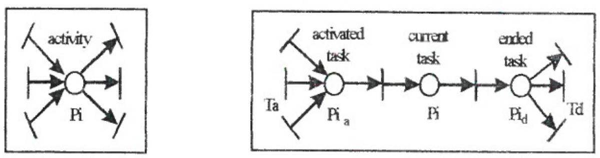


**Figure 3 a). Step Petri net.b) .Detailed step Petri net**

**5.1.2 The Actor**

The actor Petri net (Figure 4) takes into account that an actor must naturally be available, but he also needs the characteristics required by the task to be operational on it. These characteristics closely depend on his skills and are described in terms of knowledge, experience and behaviour if we consider a human actor.

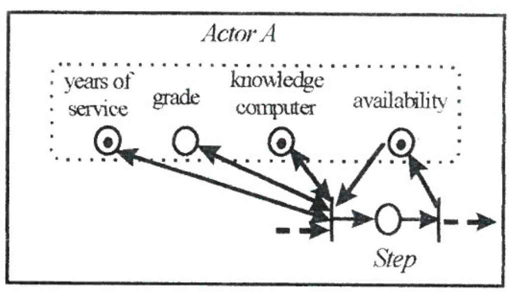


**Figure 4. Actor Petri net**

So, in terms of Petri nets, an actor can be represented by several places expressing the required skills. Moreover another place models his availability. If he possesses the required skills, the associated places are marked. If not, for example if an actor does not have a sufficient grade to do an activity which needs a decision making, the associated marking will be empty. When the actor is used for a work, the token of the place "availability" is removed at the beginning of the task execution and retrieved at the end of it.

## 5.1.3 The Information

The information is the entity representing every data item exchanged with the environment or between steps. The associated net (Figure5) is constituted of two places linked to one or several transitions. The first place "presence" corresponds to the existence of the data, and the second "availability" corresponds to their availability. Several configurations can be associated with the information. Among these there can be distinguished creation, use, modification, destruction, "recording", "unrecording", which are represented in Figure 5, and the information duplication, which is represented in Figure 6.

In case of creation (Figure 5a), these two places are linked to the downstream transition of the step. This transition points out the end of execution of an activity and then allows the creation of output information.

When a piece of information is used (Figure 5b), the two places are linked to the downstream transition of the place associated with the step which needs use it.

The modelling of information modification (Figure 5b) transforms the initial information via an activity carrying out this modification. The upstream transition of the step allows to empty the two places of the information model. Once this step is executed, its downstream transition returns tokens to the information model.

In the case of destruction (Figure 5d), the two places are linked to the upstream transition of the step. The firing of this transition removes the present tokens that are not retrieved at the end of the step.

The "recording" of a piece of information (Figure 5e) uses only a single place. The two places associated with the information are linked to transitions whose firing makes the "recording". In fact, the recorded information is compressed into a unique marking place where the availability is suppressed.

The "unrecording" (Figure 5f) consists in re-creating the initial information from a "recorded" one. This is done via a transition whose firing retrieved the basic two places model.

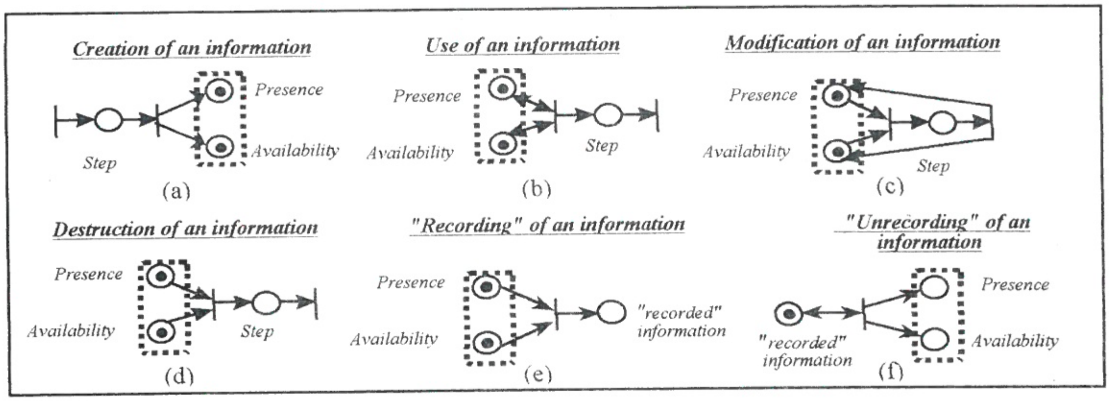


**Figure 5. Different Configurations for Information Treatment**

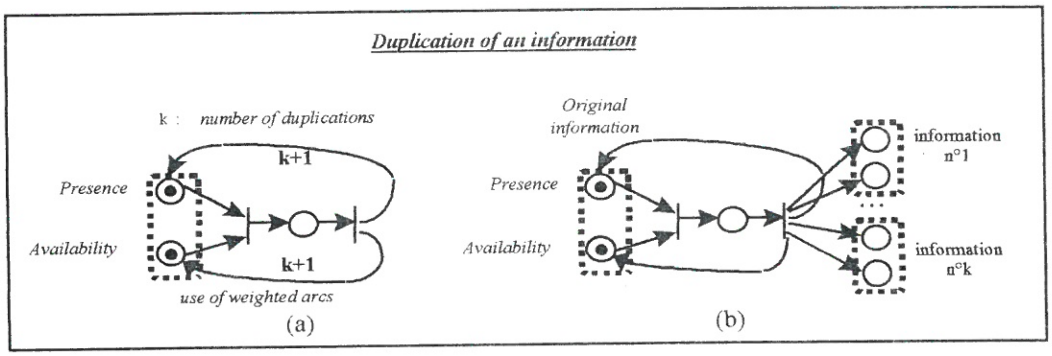


**Figure 6. Two Possible Configurations for Information Duplication**

The duplication of an information consists in creating many copies of an existing information. Two kinds of modelling are considered. In the first case, the creation of k copies can be modelled by using weighted arcs (Figure 6a). This modelling does not allow to physically distinguish the duplicated information. In the second case (Figure 6b), many basic models are associated with each duplicated piece of information to physically distinguish each copy. This modelling noticeably increases the number of places of the Petri net and its complexity.

### 5.1.4 The Function

The function is the entity for which the added-value function, the constraint function and the output function defined in the MOVES'model are distinguished. Only the latter are translated into Petri nets. The constraint and output functions are expressed by using production rules. These rules allow to define the execution conditions of a step via the constraint function

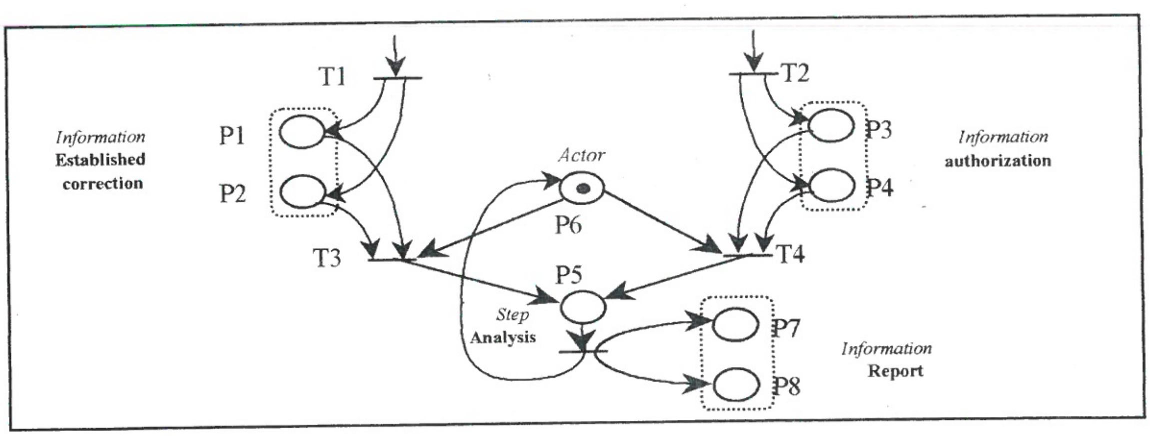


**Figure 7. Constraint Function Example Using A Logical OR Petri Nets Structure**

and the generation of the corresponding output information.

The generic form for a production rule is:

**if** [constraint function]

**then** [output function]

**else** (production rule)

Figure 7 illustrates the example of a step "Analysis" which generates an information "Report". This step can be executed if and only if it has an input information "Established correction" or "Authorization", and if the actor is available. The two types of input information "Established correction" and "Authorization" are supposed to be naturally exclusive, that means that they may not be present at the same time. The places P1-P2, P3-P4 and P7-P8 are respectively associated with the information "Established correction", "Authorization" and "Report". Place P5 represents the step "Analysis". The actor is modeled into a single place P6 in order to simplify the example.

It can be translated into the form of the following production rule :

**if** [(information "Authorization") OR (information "Established correction")] **then** [(create (information "Report")) AND (destroy (information "Authorization"), (information "Established correction"))].

Initially, the actor is supposed to be available and to have the required skills. Place P6 is then marked. The firing of the transition T1 expresses the creation of the information

Due to the exclusive rights of this information, only transition T3 or transition T4 is fireable, thus allowing the execution of the step "Analysis" which finally generates the information "Report".

## 5.2 The Behavioral Structures

The different entities previously defined allow just to translate the local behavior of a process around a step. In order to take into account the whole behavior of a process, it is necessary to identify all high level behavioral structures that can be met with in a real process. This task has been achieved through a complete analysis of the quality handbook describing the processes of an enterprise.

This study highlighted two classes of behavioral structures. On the one hand, the ordinary structures met with in any dynamic system modeling are parallelism (Figure 8), synchronization (Figure 9), iteration (Figure 10), periodicity and basic logical structures allowing the translation of elementary Boolean conditions (OR, AND, ...). On the other hand, in enterprise processes, specific structures occur. For example, the generation of outputs which are mutually exclusive, characteristic of most enterprise processes, has been emphasized. To translate complex behaviours by logical functions, the combination of basic behavioural structures has also been considered. Finally, links between processes are necessary to represent the interaction between processes.

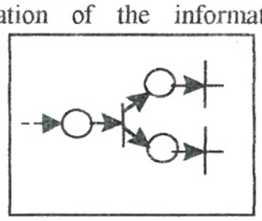

**Figure 8. Parallelism**

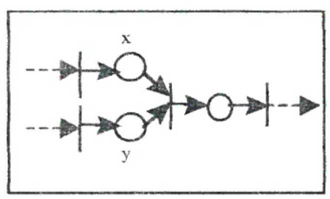


**Figure 9. Synchronization**

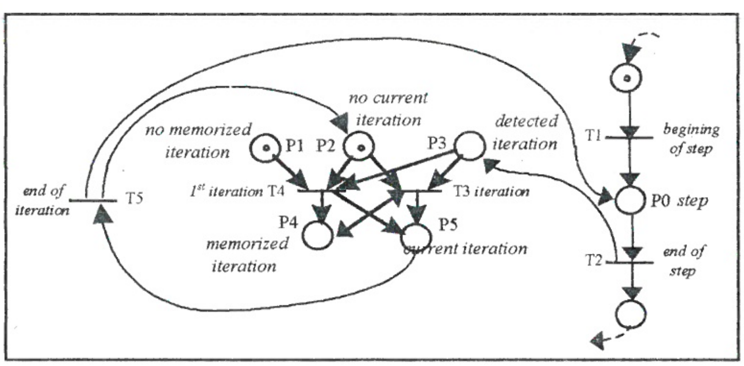


**Figure 10. Iteration**

"Established correction", and the one of T2, the creation of the information "Authorisation".

Figure 8 expresses the parallelism by an associated Petri net. The parallelism can imply some problems due to the use of resources when an actor is used in two "parallel branches" at the same time. Figure 9 expresses the synchronisation of activities. For example, the creation of an information item can be the result of the synchronization of several activities (x and y in Figure 9). The steps associated with these activities are then linked by a same output : this corresponds to a synchronization.

However some behavioral structures are more complex to model. Figure 10 expresses in a simple example the net part used to manage an iteration. Two particular states must be modeled: the first signifying that an iteration is under execution, and the second signifying that an iteration has already been executed. Each state is represented by two complementary places : P1 and P4, for the memorization of the first iteration, and P2 and P5 for the identification of the iteration which is under current execution. The detection of an iteration is expressed by the marking of place P3. At the initial state, P1 and P2 are marked in order to translate the absence of an iterative behavior. The first iteration induces T4 firing. It removes mark from P1 and adds mark to P4. During the next iterations, only transition T3 will be fireable, then allowing the marking of P5 during an iteration. An example using the iteration structure is given in Figure 13.

# 6. Modeling and Analysis of An Enterprise Process

In order to model a real enterprise process, the user has to identify the set of tasks, resources and behaviours which are respectively represented by steps, actors and constraint functions. Then a transparent translation of this model into an equivalent Petri net is done by observing the transformation rules previously defined. From this net an analysis is realized. During this analysis phase, structural and behavioural properties are checked. These properties are interpreted by taking into account what the modeled entities really represent. Then their possible origins are translated to the user.

Petri nets analysis needs first to precise the initial state of the studied process, that is the initial marking. Following the preliminary task, the analysis is done in two steps. The first one, associated with phase 1 of the analysis, consists in checking the Petri net properties when this latter one is considered without global feedback. The second one, associated with phase 2 of the analysis, corresponds to the Petri net properties study when a global feedback is structurally added. The proposed approach and the properties checked are illustrated in the chart of Figure 11 and explained in the following.

## 6.1 Preliminary Task : The Initial Marking

The definition of the initial marking of the process associated Petri net needs to know a large amount of data about the process structure and the entities used. For example, the information present at the beginning of the process (primary input information) must be known, and the characteristics, skills and availability of the used actors must be defined.

Moreover, some places used to translate the process behaviour need be initialized. These data can be extracted from the enterprise model, from database, or asked for the user. Once the initial marking defined, the analysis process can be carried on in two phases.

## 6.2 Phase 1 of the Analysis

Phase 1 of the analysis is realized from an equivalent Petri net model of the process without general feedback. The checked properties are, first, boundedness, quasi-liveness, deadlock or conflict detection which are based on the

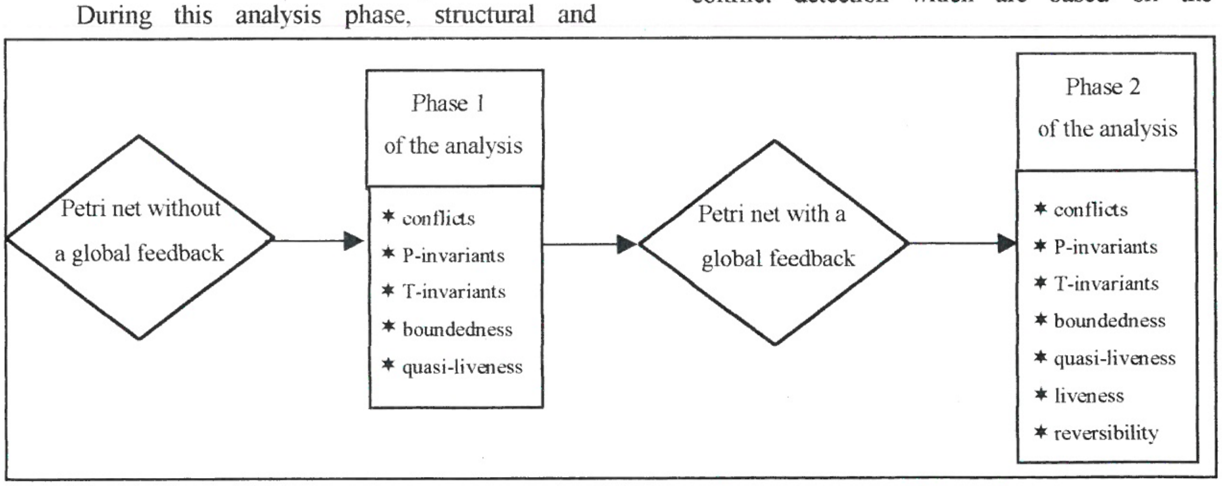


**Figure 11. The Proposed Analysis Approach**

coverability graph study. Then linear algebra techniques can be used to determine the places and transition invariants. Finally, the checked or unchecked properties are interpreted to put an emphasis on their origins. The latter come generally from an incomplete or bad translation by the user of the real process structure or behaviour. But the detected problems can also come from a mismanagement of the entities composing the studied process.

This first analysis phase is carried on until all the studied qualitative properties deduced from the coverability graph are come across. At this point, to draw a final conclusion on the good organisation of the process, a global feedback must be added to the initial process Petri net model.

### 6.3 Phase 2 of the Analysis

This global feedback allows to check that the studied process activities are still available whatever the process behaviour. To be realized correctly, this feedback must consider all the final execution possibilities of the process to restore the initial marking. It is a non obvious process which needs to take into account the coverability graph, the initial state and the net structure. Finally, if the process has been correctly described, the coverability graph associated with the new net will not present leaves.

Moreover, from this new net, the previously studied properties such as liveness, and reversibility can be investigated during phase 2 of the qualitative analysis.

### 6.4 Interpretation of the Properties of the Analysis

Considering a well structured process the following properties must be verified on the equivalent Petri net model:

- The Petri net enterprise model must be bounded to 1 if the possibility to model a duplication activity by using generalised Petri nets is excluded. That is to say that during its functioning, a place must contain at most one and only one token. So the boundedness property must be checked.
- Quasi-liveness during phase 1 of analysis, and liveness during phase 2 of analysis must be checked. In case of non liveness some deadlocks can be brought to light.
- All the T-invariants must cover the set of Petri net transitions if the net is live.
- The reversibility properties must be obtained at phase 2 of the analysis level.
- The conflicts can be detected from the coverability graph analysis. However it is important to point out that a conflict does not necessarily reveal a problem. It may also correspond to a real imprecision in the process, for example when a decision is made without knowing its real value.
- Finally the P-invariant study informs about the internal iterative process.

The non satisfaction of one of these properties reveals some problems of management modeling. Some examples of properties interpretation are explained in the following. However it is clear that this is a difficult work because it must take into account the non checked property, its origin in the Petri net, its signification, but also its context. In fact, it is necessary to establish a database from which a symptom and its context can indicate to the user a set of possible interpretations on the enterprise model. The user will then take the possibility which seems to be the most probable to him.

## 7. Representation and Analysis of An Enterprise Process Example

### 7.1 Example of Enterprise Process

In this Section, the example of a simple enterprise process (Figure 12) which allows (or does not) to produce a product after its ratification, is briefly presented.

Its behavior can be explained as follows. First, the production service (Prod. Serv.) of the enterprise analyses on whether some problems can occur during the production of the product and then it establishes an expert report (Exp. Report). The latter is used as a basis for the ratification of the product by the production and technical services (Tech. Serv.). In case of non conformity, some corrective actions are proposed. These modifications are also analysed by the production service. If the conformity is validated (Production Request), the authorization (Decision) of production is then decided by the two services.

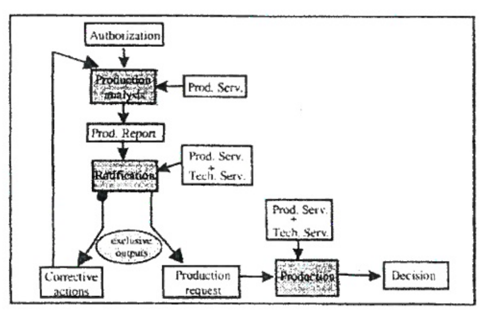


**Figure 12. Example of Enterprise Process**

## 7.2 Petri Net Translation

The process represented in Figure 12 is translated into the Petri net illustrated in Figure 13 by making use of the structures presented at Section 5. Its construction takes place during several steps which have been presented in manage the evolution of the process. The latter ones concern the places which are used for the management of the reiteration but also the places of sequence (S1 to S5) which control the consistent evolution of the process.

Let us note that in order to simplify the Petri net, the actors involved in the process have the

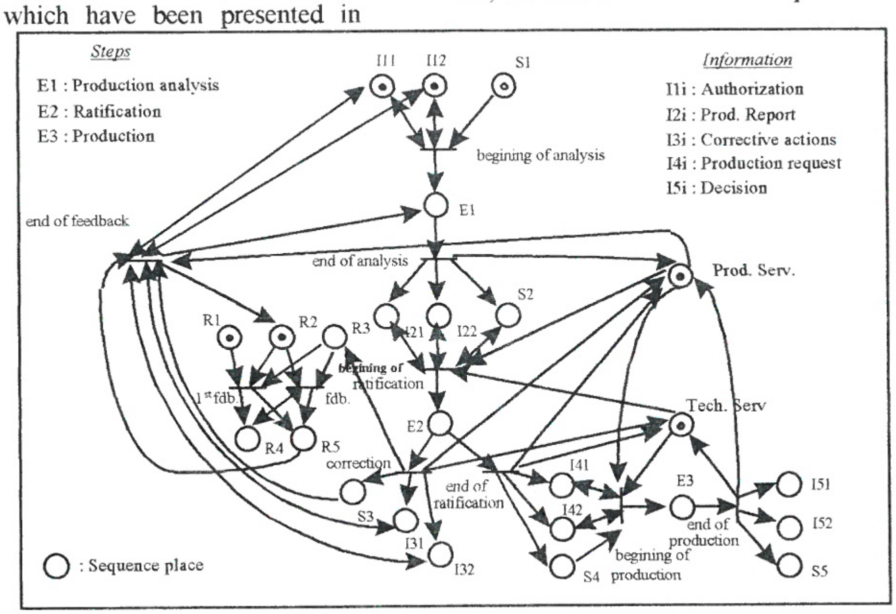


**Figure 13. M1: Petri Net of the Studied Process**

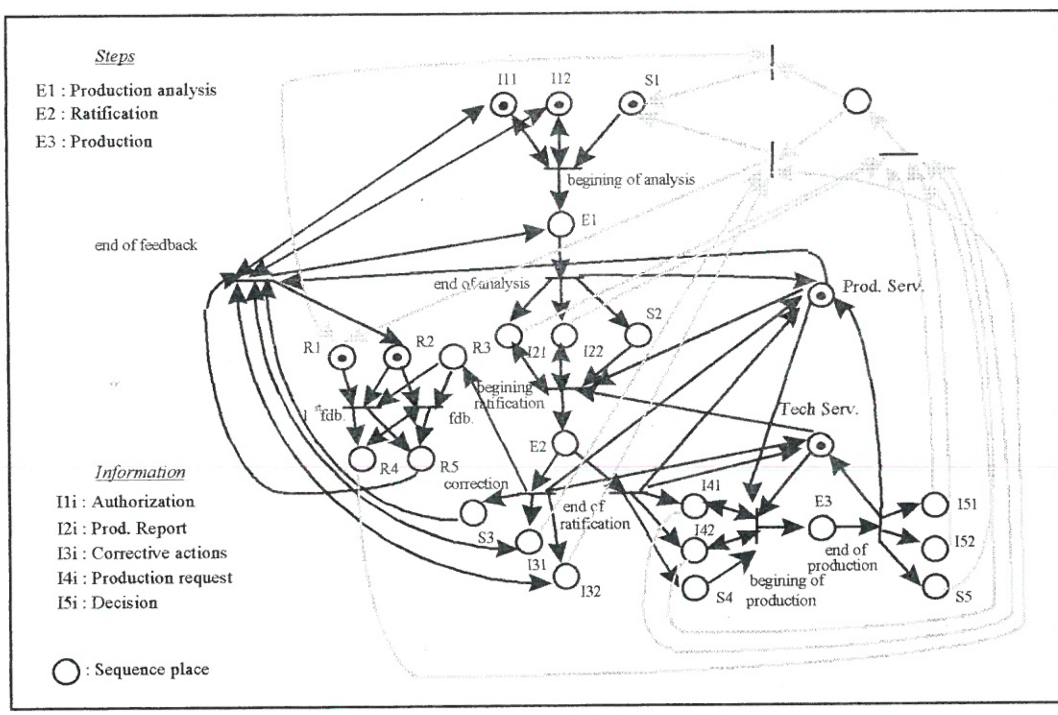


**Figure 14. M2 : Petri Net of Reiterated Process**

Section 6.1. First, its structure is generated, and the initial marking is assigned. The latter allows, on the one hand, to specify the available information at the initial state (Authorization), and, on the other hand, to initialise the places which will be used to required skills in this example. So, they can be modeled by a unique marked place.

This Petri net must be completed in a second step by a global feedback which is to retrieve the initial marking from all possible final states. This Petri net is illustrated in Figure 14.

### 7.3 Example of the Analysis Procedure

Table 1 summarizes the analysis by checking the qualitative and structural properties for every step of the modeling. This analysis implies the examination of four successive Petri nets : M1 (Figure 13), M2 (Figure 14), M3 (Figure 15) and M4 (Figure 16). As this first analysis highlighted some modelling problems, the initial Petri net M1 was modified into the Petri net M3 taking into account the first analysis results obtained from M1 and M2 analysis. For every step of the modeling, if the analysis shows an unbounded Petri net, the number of the states associated with the

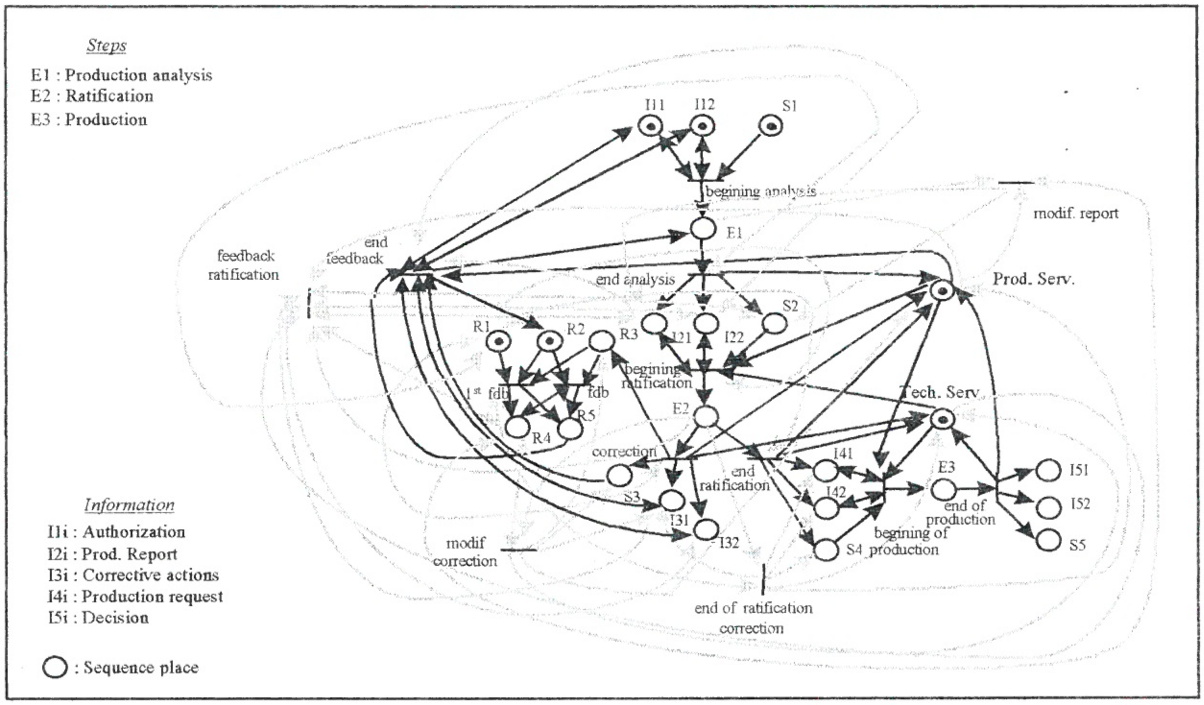


**Figure 15. M3:Modified Petri Net Without Global Feedback**

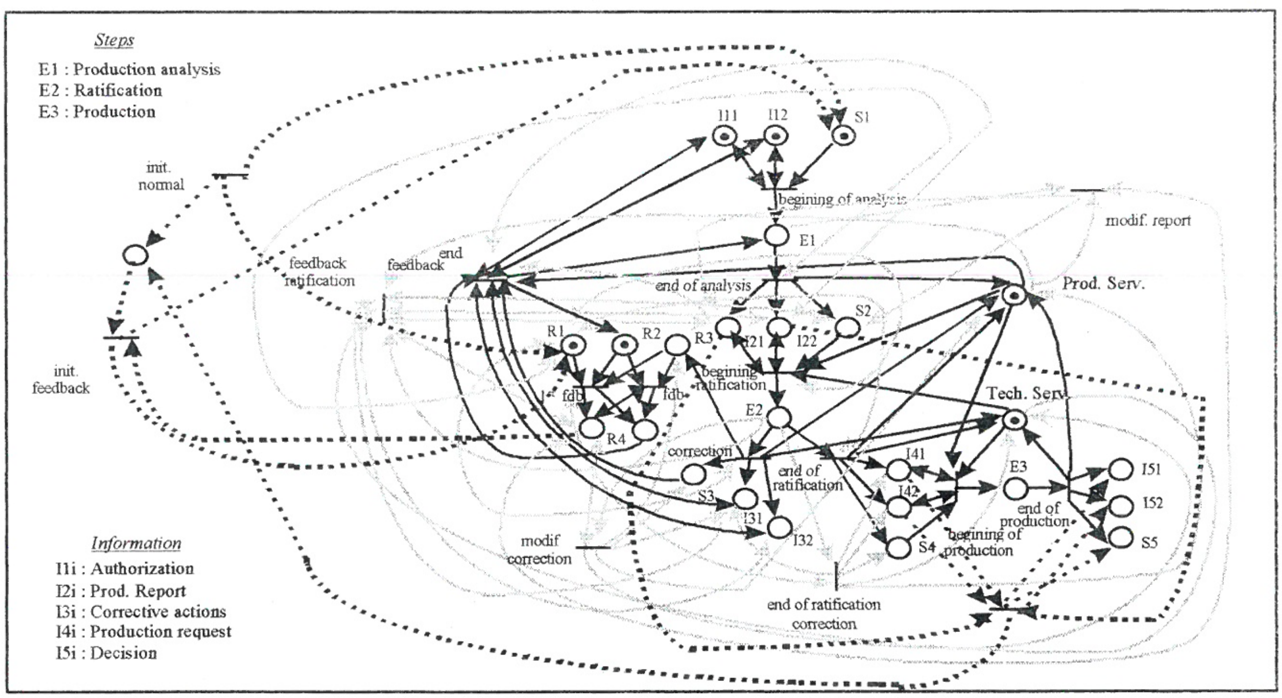


**Figure 16. M4 : Modified Model With General Feedback**

coverability graph, which allows the analysis of such kind of a situation, is represented in Table 1.

user will have to modify its description of the process functioning at the enterprise model level to take into account the analysis which has been done.

**Table 1. Results of the Process Qualitative Analysis**

| | | | characteristics of Petri nets | | | | properties | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Petri Net | step of analysis | phase of analysis | nb. of places | nb. of transi-tions | nb. of arcs | nb. of states | con-flicts | boun-dedness | quasi-liveness | liveness | T-inv. |
| M1 | 1 | 1 | 25 | 10 | 62 | 24 | yes | no | yes | no | no |
| M2 | 2 | 2 | 26 | 13 | 80 | 54 | yes | no | yes | no | yes |
| M3 | 3 | 1 | 25 | 14 | 96 | 16 | yes | yes | yes | no | yes |
| M4 | 4 | 2 | 26 | 16 | 114 | 18 | yes | yes | yes | yes | yes |

In the following, every step of this study is explained.

### 7.3.1 Step 1 : Analysis of the Initial Model M1 Without A Feedback (Phase 1)

From the Petri net M1 a first analysis is conducted. The structural properties are studied without considering the marking of the Petri net. Hence, the search for conflicts on M1 shows the existence of a conflict on the downstream transitions of place E2 associated with the step "Ratification". This conflict means that after E2, a decision must be made and the user is not aware of it. Then the process is not deterministic. So, this detection gives signal evidence of a real process problem.

The research of invariants shows an absence of T-invariants. That expresses that in this case the internal loop of the process has not been detected. This is due to the unboundedness of the Petri net. Effectively, the study of the qualitative properties by the coverability graph of M1 (illustrated in Figure 17) shows an unboundedness on places I21 and I22 associated with the information which is generated before the step "Ratification", when a correction produced. This points out a more serious problem which can be accounted for by a bad description of the process functioning during the design of the enterprise model.

The Petri net represented in Figure 13 expresses that a new account report of analysis and corrections is generated for each process iteration. In fact this report is to be modified when a new product checking is required. The user will have to modify its description of the process functioning at the enterprise model level to take into account the analysis which has been done.

### 7.3.2 Step 2 : Analysis of the Initial Model With A Feedback : M2 (Phase 2)

The addition of a global feedback to the previous Petri net M1 description (Figure13) allows to obtain the Petri net M2 (Figure 14) based on which a new analysis is carried out. At this point, the structural analysis allows to detect a T-invariant when the functioning of the process does not imply a correction procedure. So it can be concluded that this part of the process seems to be correctly constructed.

The qualitative analysis shows that the Petri net M2 is quasi-live and already unbounded. For reasons of dimension (54 states), the coverability graph associated with the Petri net M2 is not represented here. To try to reach a bounded and live Petri net, it is necessary to take into account the analysis results obtained at Phase 1. The user must then state precisely the modality of creation of the expert report in the initial enterprise representation. This new information handling will then be modeled into the equivalent Petri net.

### 7.3.3 Step 3 : Analysis of the Modified Model Without A Feedback (Phase 1)

To take into account the corrections induced by the unboundedness at the description level of the process functioning (model M1 in Phase 1 of analysis), it is necessary to construct a new Petri net M3. This net, without a general feedback, is represented in Figure 15.

From M3, the structural analysis detects the T-invariant relative to the reiteration of the

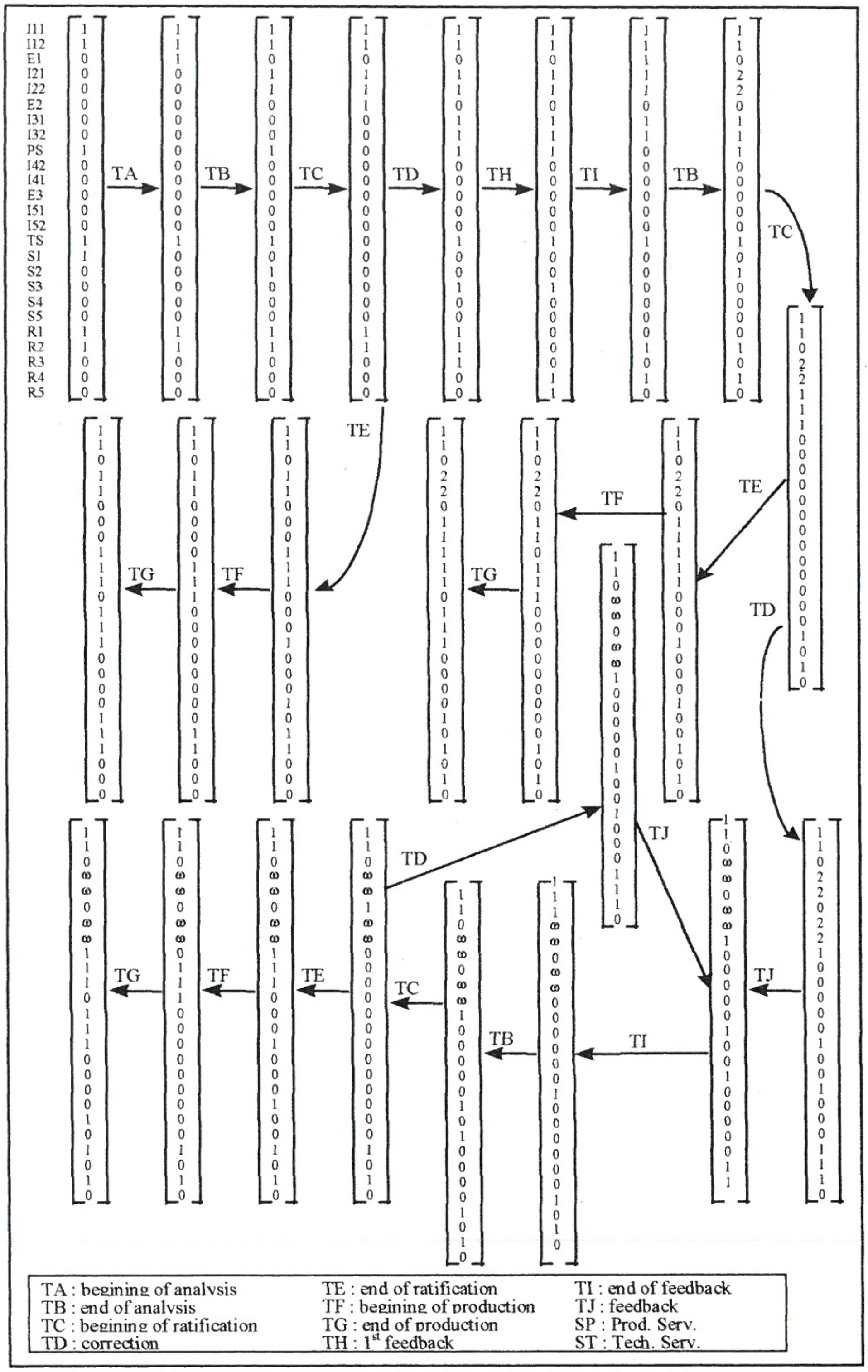


**Figure 17. M1 Coverability Graph**

correction procedure which is inherent to the process.

The checking of the boundedness property shows that all places of the net are 1-bounded, as one can observe on the marked graph of M3 illustrated in Figure 18. So, the problem detected during the analysis of the first model M1 seems to be solved. Here the non- liveness of the Petri net is absolutely normal because M3 is not totally looped. However let us note that M3 is quasi-live, which proves the possibility of realizing at least once every event described in the modeled process.

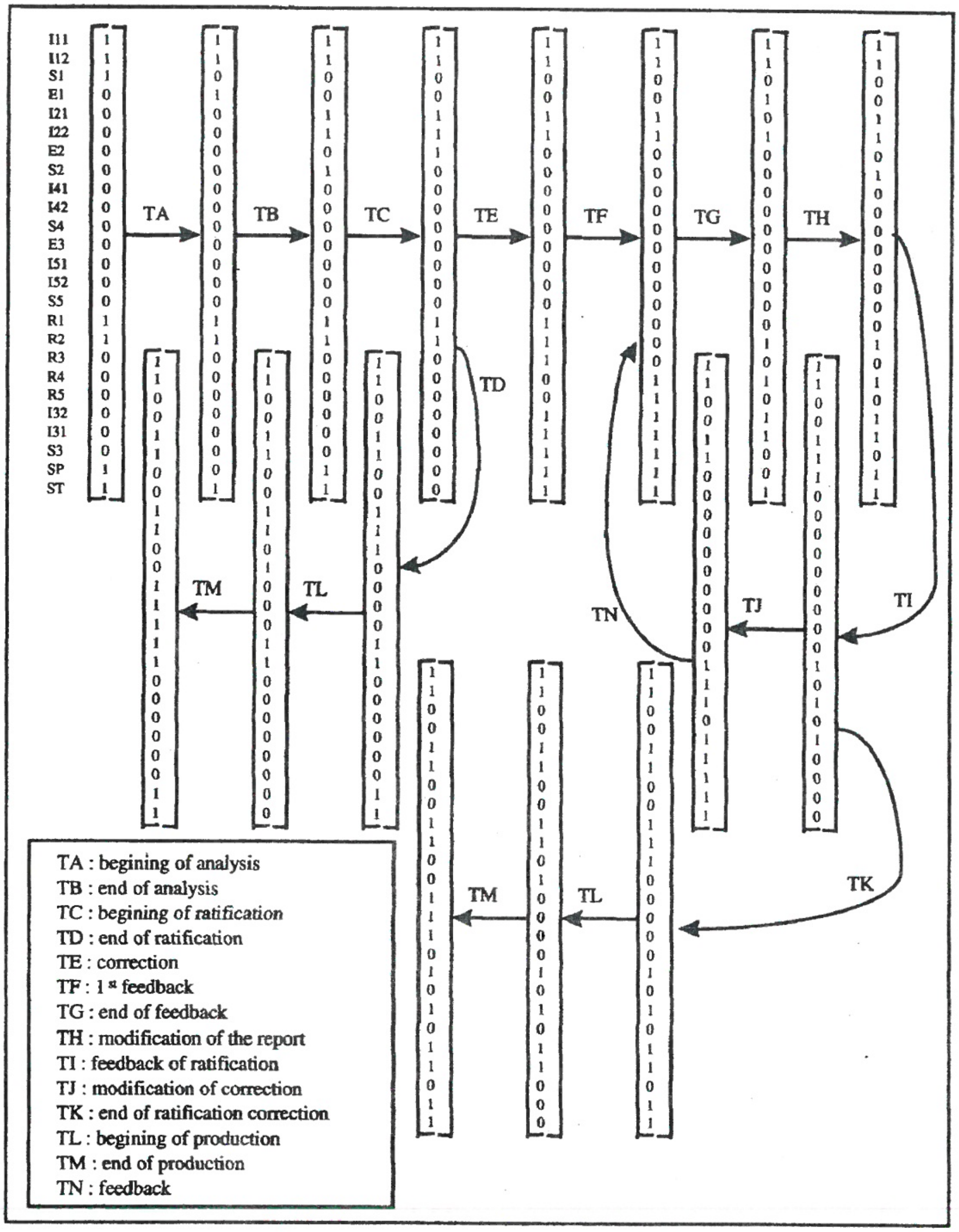


**Figure 18. M3 Marked Graph**

### 7.3.4 Step 4 : Analysis of the Modified Model With A Feedback (Phase 2)

The Petri net M4 (illustrated in Figure 16) matches the Petri net M3 which a general feedback has been added to. It allows to take into account the iteration capacity of the modeled process and then to proceed on the final analysis.

The analysis is based on the marked graph of the Petri net M4 represented in Figure 19. It shows an 1-bounded and live Petri net. Moreover, the structural analysis allows to notice that all transitions belong to T-invariants. Finally, the obtained Petri net is reversible.

Then all the expected properties of good behavior of the Petri net are checked. This allows to conclude that the process which has been described with the enterprise model seems to be absolutely coherent.

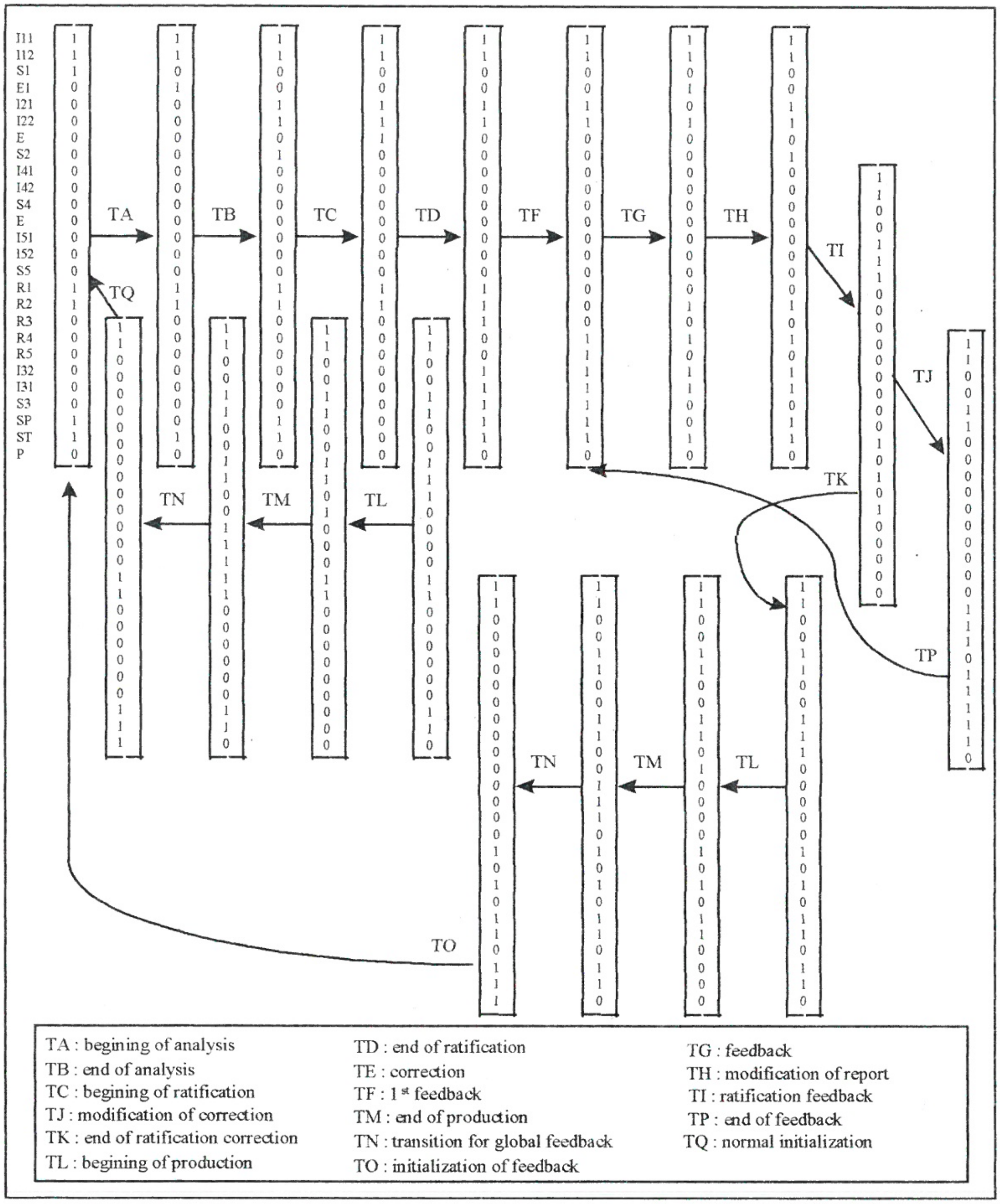


**Figure 19. M4 Marked Graph**

## 8. Conclusion

This paper proposes a method allowing a formal qualitative analysis of enterprise processes. This paper is based on an automatic translation of an enterprise model into autonomous Petri nets. From the definition of an equivalent Petri net for every entity of the enterprise model and of the behavioral structures which can be found in enterprise processes, an example of enterprise process has been studied. This translation is done in several phases relatively with the analysis method which is associated with it. The results of the study of the properties (conflicts, liveness, boundedness, reversibility, P- and T-invariants) are then interpreted on the enterprise model to guide the user in his analysis of dysfunctioning or modeling errors. This interpretation of qualitative properties in terms of model or management errors is difficult because of the influence exerted by the system context.

The next step of this work will be the transcription on an algorithmic form of the translation rules of the enterprise model to the

Petri net model. This phase is essential when dealing with real enterprise processes. Moreover, due to the high complexity of places and transitions in the obtained Petri net, some techniques of partial Petri net reduction must certainly be used to make the Petri net analysis less complicated. In this case, due to the loss of meaning of the Petri net description, these reduction techniques will be coupled with an iterative process which locally expands the model areas there where a dysfunctioning has been detected. It would also be attractive to extend the formal analysis possibilities to the accessibility state problem. Finally, an expert database will be built and enriched to interpret the results of the qualitative analysis of the Petri nets in terms of modeling or management errors.

## REFERENCES


1. MINTZBERG, H., **Grandeur et décadence de la planification stratégique**, ÉDITIONS DUNOD, 1994.

2. ECHARD, R., **Contrôle de la qualité, certification et concurrence**, **Certifier la qualité**, GEIST, PRESSES UNIVERSITAIRES DE STRASBOURG, 1998.

3. VERNADAT, F.B., **Enterprise Modeling and Integration : Principles and Applications**, CHAPMAN &HALL, 1996.

4. CRESTANI, D., PRUNET, F., CHAPURLAT, V., LARNAC, M., MAGNIER, J. and CHALVET, D., **A Generic Model Oriented Multi Criterion Added Value Analysis for Factory Modelling**, WMC'97, New Zealand, 1997, pp.159-165.

5. DAVID, R. and ALLA, H., **Petri Nets and Grafcet : Tools for Modelling Discrete Event Systems**, PRENTICE-HALL, New York, 1992.

6. ROSS, D.T. and SCHOMAN, K.E., **Structured Analysis for Requirements Definition**, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 1977, pp. 9-15.

7. ACNOS, **Intégration des activités non structurées dans la modélisation des systèmes de production**, Action initiative du DSPT8 en productique, Rapport final, 1997.

8. DOUGMEINGTS, G., **Méthode GRAI : méthode de conception des systèmes en productique**, Thèse d'état, Automatique, Université Bordeaux I, 1984.

9. ROBOAM, M., ZANETTIN, M. and PUN, L., **GRAI-IDEF0-Merise (GIM) : Integrated Methodology To Analyse and Design Manufacturing Systems**, COMPUTER INTEGRATED MANUFACTURING SYSTEMS, 1989, pp.82-98.

10. AMICE, **CIMOSA : Open Systems Architecture for CIM**, SPRINGER-VERLAG, Berlin, 1993.

11. FOULARD, C., **La modélisation en entreprise : CIM-OSA et ingénierie simultanée**, ÉDITIONS HERMÈS , 1994.

12. WILLIAMS, T. J., **The Purdue Enterprise Reference Architecture**, Instrument Society of America, Research Triangle Park, NC, 1992.

13. SCHEER, W. A., **Architecture for Integrated Information Systems**, SPRINGER-VERLAG, Berlin, 1992.

14. BERNUS, P. and NEMSE, L., **A Framework To Define A Generic Enterprise Reference Architecture and Methodology**, Proceedings of the 3rd International Conference on Automation, Robotics and Computer Vision ICARCV'94, 1994, pp. 88-92.

15. COHEN, G., DUBOIS, D., QUADRAT, J.P. and VIOT, M., **A Linear System Theoretic View of Discrete Event Processes and Its Use for Performance Evaluation in Manufacturing**, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, February 1998.

16. LARNAC, M., CHAPURLAT, V., MAGNIER, J. and CHENOT, B., **Formal Representation and Proof of the Interpreted Sequential Machine Model**, EUROCAST'97, Las Palmas, 1997.

17. MURATA, T., **Petri Nets : Properties, Analysis and Applications**, Proceedings of the IEEE, Vol. 77, No. 4, 1989, pp.541-580.

18. BERTHELOT, G., **Transformations de réseau de Petri**, TECHNIQUE ET SCIENCE INFORMATIQUES, Vol. 4, No.1, 1985, pp. 91-101.