



**HAL**  
open science

## Use of interpreted petri nets for human-machine dialogue specification in process control

Meriem Riahi, Faouzi Moussa, Christophe Kolski, Mohamed Moalla

### ► To cite this version:

Meriem Riahi, Faouzi Moussa, Christophe Kolski, Mohamed Moalla. Use of interpreted petri nets for human-machine dialogue specification in process control. ACIDCA'2000, Mar 2000, Monastir, Tunisia. hal-03409109

**HAL Id: hal-03409109**

**<https://uphf.hal.science/hal-03409109v1>**

Submitted on 14 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Use of Interpreted Petri Nets for the dialogue specification in process control

Meriem RIAHI\*, Faouzi MOUSSA\*, Christophe KOLSKI\*\*, Mohamed MOALLA\*

\* PIE : Productique et Informatique d'Entreprise - Faculté des Sciences de Tunis - Département des Sciences de l'Informatique - Campus Universitaire, 1060 Le Belvédère - Tunis, Tunisie  
Tél : (216) (1) 872 600, Fax : (216) (1) 885 190 , email : moussa<maddouri@ensi.rmt.tn>

\*\* LAMIH : Laboratoire d'Automatique et de Mécanique Industrielle et Humaine, URA CNRS 1775 - Université de Valenciennes et du Hainaut Cambrésis, Le Mont Houy, 59304 Valenciennes Cedex, France  
Tél : (33) (0) 3 27 14 12 42, Fax : (33) (0) 3 27 14 12 94, email : kolski@univ-valenciennes.fr

## Abstract :

This paper studies an approach for designing graphical Interfaces for process control. This approach is based on a formal modelling of the Human-Machine System behaviour. This modelling allows the deduction of the user requirements and then, the identification of the interface objects. The formal aspect permits the validation of the generated specifications before proceeding to the automatic generation of the interface.

A formalism using Interpreted Petri Nets is here proposed for modelling the Human-Machine dialogue.

**Key words :** User Interface (UI), Human-Machine System (HMS), Formal specification, Interpreted Petri Nets, Dialogue modelling.

## Introduction

With the considerable evolution of computer sciences and the important use of software designed to control automation's applications with high degree of security, the role of the User interface (UI) becomes preponderant. Many researchers were tended towards design and development of UI. Within these researches, several models and approaches were defined. Many tools for interface generation were also proposed. Our works can be divided into two main points : (i) the choice of a rigorous formal tool adequate to the application's context ; (ii) the perfection of a design approach for graphical user interfaces.

This paper starts with a brief critical synthesis of related works on these two points. Then, it describes the proposed approach for UI design, focusing on the modelling of human-machine dialogue.

## Related Work

Researches, nowadays, are oriented more and more to formal approaches. In fact, only formal techniques allow the designers to describe the external behaviour of any system without considering the implementation context. These techniques permit also the validation of specifications before proceeding to the effective UI generation. They are recommended specially for complex and critical systems such as industrial processes [13]. So, it is very interesting to use a formal technique for specifying UI for process control. But, which technique must we choose?

In literature, we can find several formal specification languages based on different theories: universal algebra, typed logic, theory of sets (Z, VDM, B) or automata theory (state transition diagrams, Petri Nets,...). Each of these families, has a predilection domain (data processing, real time, protocol management,...), a community of researchers and users and a number of variants or disciplines [7].

Particularly, in the field of human computer interaction, some formal methods were proposed by different researchers. The state transition diagrams were proposed since the 60's, for modelling the screens chaining [10]. State transition diagrams were then proposed by Jacob for direct manipulation interfaces [5]. They were followed by Petri Nets [14][6], to which we can associate the ICO formalism [9], then, temporal logic and LOTOS [11], the Z formalism [4], the LUSTRE language [2] and recently B method [1].

What one can reproach, mainly, to techniques like Z, lotos, or Lustre, is the difficulty that exists in reading and manipulating their notations. This limit is removed with graphical techniques like state charts and Petri nets. The latter (technique of Petri Nets) seems to be particularly adapted (suitable) for modelling the dynamic aspect of human-machine dialogue in the particular context of process control. Petri nets are expressive for event's aspect, synchronism and parallelism, as much pertinent criterias for constraints of interactive graphical interfaces. Moreover, Petri nets have been already used for human machine dialogue modelling [9] [14] or human task description [6]. They are, here, proposed for modelling human-machine system (HMS) behaviour and interface objects, as explained further.

For methodological viewpoint, we can notice that most of the approaches proposed in literature, apprehend only one aspect of the interface design problem. Considering, especially, the researchers focusing on using Petri Nets, some observations should be notified :

- Palanque propose to model the interactive application's interface using the ICO formalism (Interactive Cooperative Objects) [9]. This provides a formal specification of the UI and its behaviour which allows a functional analysis and a possibility of the specification's validation. But, his approach is based on an object oriented method. This method doesn't explicit the principle of interactive objects' identification. He leaves that to the common sense of the designer.
- With the TOOD method [6], Mahfoudhi propose a design approach based on an hierarchical decomposition of the HMS in tasks modelled using Petri Nets. This allows the identification of user requirements. An object design method of the UI is then proposed. But, there is many problems, not yet resolved, to which the designer is faced for automatic generation of the interface.
- No one, of the authors, take into consideration, the dysfunctionning modes in their HMS analysis. Nevertheless, it is a crucial aspect for the critical systems.
- Concerning the particular industrial applications, the Ergo-Conceptor system [7] presents an ergonomic approach for the interface design allowing an automatic interface generation. However, in this approach, there is no indication to how identify the user requirements. It allows only a static interface generation. It didn't consider the dynamic aspect of the interactive objects and the human machine dialogue.

In view of these limitations, our objective is to propose a most possible completed design approach for automatic interface generation for process control. A synthetic presentation of this approach, is first, advanced in this paper. Then, we focus on the formalisation of the human-machine dialogue using Petri nets.

## Proposed approach

In the proposed approach, we try to benefit from the advantages of different previous works. We aim, especially, to satisfy these points :

- analyse different functioning contexts (normal and abnormal functioning) as well as the operator's tasks,
- identify the user requirements from this analysis,
- automate the deduction of interface objects,
- benefit from formal techniques to specify the interface, the objects and their behaviours,
- use guidelines for objects' presentation,
- verify the specifications before automatic generation.

The proposed approach is made up of six steps (figure1).

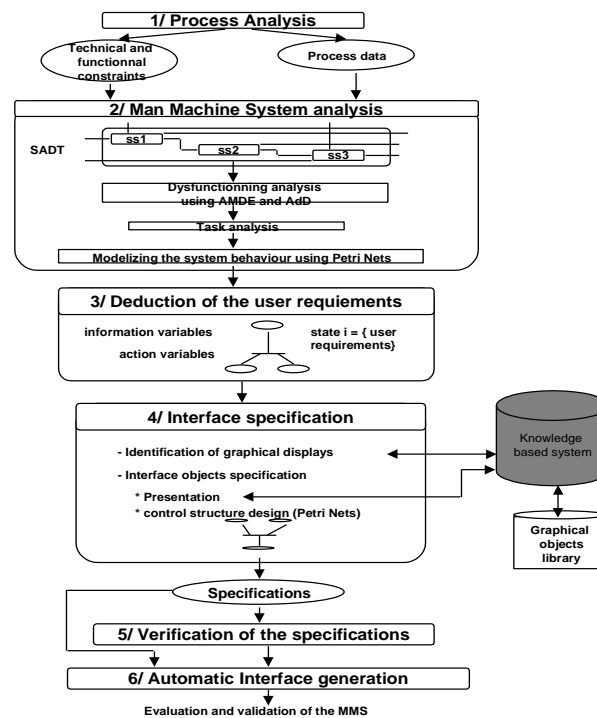


Figure 1: Methodological framework of the proposed approach

First, a step of process analysis is needed. It provides a document containing the process data and the different technical and functional constraints. Then, a second analysis of the hole HMS in terms of the process, its command system and the operator's tasks, is carried out. The dynamic aspect of the HMS is modelled using Petri nets. The third step concerns the identification of the user requirements in terms of interface objects from previous analysis. The interface specification is then ensured. We benefit, afterwards, from the formal technique used to verify the generated specifications. The last step of this approach is dedicated for the automatic interface generation.

The principle of HMS analysis and user requirements deduction is explained in [12]. This paper focus on the human machine dialogue specification from the HMS's behaviour modelling using Petri nets.

To be lucid, we develop this point, in the sections below, through an industrial pedagogical example. This example is now presented.

## **Presentation of the example**

As first application, we have chosen a simple industrial process which consist on manufacturing metallic products. The process is controlled by an operator in the control room.

It consists on : (1) producing the metallic sheets, then (2) covering these sheets with a chemical solution. This solution is obtained by mixing two products C and D. To prepare the product C, we have to mix two other solutions A and B, having variable temperatures and flows ( $t_A$ ,  $t_B$ ,  $f_A$  and  $f_B$ ). The result C must be at fixed temperature and flow ( $T_C$  and  $F_C$ ). The operator's task consists, then, in controlling the mixer functioning and intervene when a dysfunctionning appear to restore the normal functioning state of the system.

The step 2 of the proposed approach consists in applying the SADT method to decompose the HMS and identify the appropriate elementary sub-systems. Four sub-systems can be identified in this example: (1) a first sub-system for preparing the different components (metallic sheets, products A, B and D), (2) a second sub-system for preparing the mixture C (the mixer), (3) a third sub-system is needed for preparing the chemical solution and (4) the last is for covering the sheets with the chemical solution. We choose to study the second sub-system (the mixer). We suppose, then, that an analysis using the classical and complementary methods (FMEA and FAT) has been carried. This analysis identify two possible panes: (1) deviation of the temperature or the flow of the product C, and (2) the overflow of the storage tank.

Thus, we deduce, principally, three states of this system : a normal functioning state, a dysfunctionning state relative to the deviation of the temperature or the flow of the product C, and a second dysfunctionning state relative to the overflow of the storage tank.

We suppose also, that an operator task analysis was carried out. This analysis identify three essential operators' tasks relative of the three states: a supervision task in normal functioning, a correction task consisting in adjusting temperature and flow of the product C, manipulating different parameters of the system ( $t_A$ ,  $t_B$ ,  $f_A$  and  $f_B$ ), and an other correction task relevant to the second pane. It consists to empty the storage tank.

We consider, principally, the two latter tasks and we suppose that a task analysis was carried out. This allows the identification of the actions procedures. The arguments of these procedures define the variables which should be manipulated by the operator. They will constitute, afterwards, the user requirements.

This preliminary analysis of the HMS aims to identify the different functioning states of the system and possible interventions of human operator. The purpose is to specify adequate interfaces for the system control. The HMS behaviour is, then, modelled using Petri nets. The principle of this modelling is exposed below.

## Modelling the HMS behaviour

The specification of the dialogue component of the UI is the most delicate part in the specification task. In fact, the human-machine dialogue modelling require the detailed description of all of the possible and acceptable states of the interactive application according to possible evolutions [9].

In order to succeed this task, we propose within this approach to start by modelling the HMS behaviour using Interpreted Petri Nets [3]. The places represent the HMS states. The evolutions between these states are modelled by the transitions. The Petri net in figure 2, illustrate the modelling of the tank sub-system. The place P1 models the normal functioning state. The places P2 and P3 model respectively the two dysfunctioning states. P4 and P5 model the valuation states after the operator's actions. These states imply nothing at the interface. That's why, they are considered as non observable places and will be ignored in the step of deducing the user requirements.

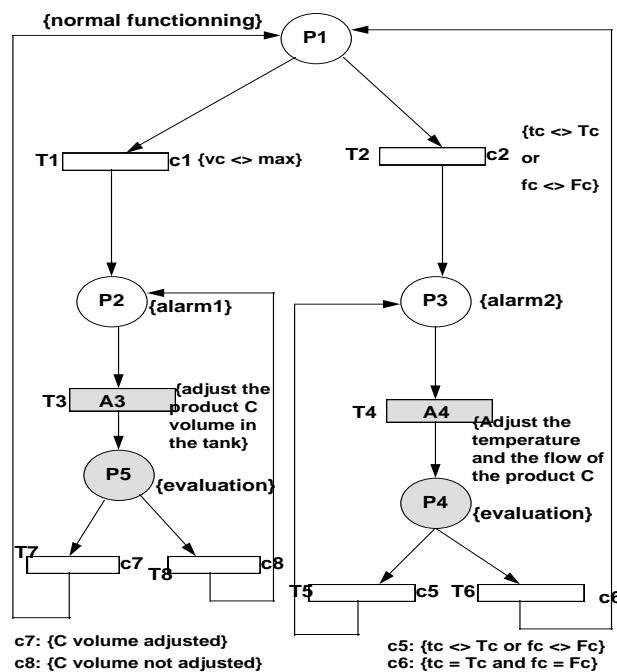


Figure 2 : Dynamic modelling of the sub-system "Mixer"

T1 and T2 model the system state changing from the normal functioning state to the alarm states, under the passing conditions c1 and c2. T3 and T4 model the operator's interventions for regulating the system. These transitions aren't really passed unless the operator finish his actions. A3 and A4 are the action procedures associated to these transitions modelling the execution of the operator's tasks. T5, T6, T7 and T8 model the state evolution after the operator's intervention.

Our objective is to identify from the HMS analysis, the appropriate interface objects in first time, then models the human-machine dialogue. The idea is to associate to each state, the appropriate informational variables and to each transition, the necessary command variables. These variables identify the user requirements and then interface objects. This is explained in [12].

Two kinds of graphical objects will be considered : informational objects will be associated to the places of HMS model, representing the system's state ; command objects will eventually be associated to the transitions modelling the operator's tasks.

When a transition becomes passable, the action objects associated to this transition will become activated according to the conditions expressed in the action procedures. They will be deactivated after passing the transition. The informational objects associated to the places of the Petri Net, are displayed when a token is put.

## **Specification of the human-machine dialogue**

Once, graphical objects have been identified, the next step consists in specifying this interface in terms of presentation and dialogue. For interface presentation, we propose to benefit from the researches carried out since the eighties, by some researchers [15] concerning knowledge-based approaches for automatic evaluation and design of UI used in process control. In fact, there are many ergonomic rules available in literature. For instance, Gilmore and al. (1989), inspired by their works with the U.S. Nuclear Regulatory Commission (USNRC), have written a handbook of guidelines for user-computer interface design in process control. These guidelines were organised into four categories, namely, (1) video displays, (2) control and input devices, (3) control/display integration, (4) workplace layout and environmental factors. Each guideline was presented in a structured format. Hundreds of guidelines adapted to this particular field (process control) were listed in the handbook.

Several model-based tools for the interface development are described in the literature. The Ergo-Conceptor system, one of these tools, is oriented towards process control interactive applications. So, following the principle of Ergo-Conceptor, our system will be able to decide on the appropriate displays to associate to each sub-system. For that, it will take into consideration, on the first hand, the characteristics of each sub-system (the list of its functioning states, the user requirements associated to each state,...) and on the other hand, specific formalised guidelines stored in its knowledge bases.

Once, the different graphical displays are identified, we have to specify the presentation and the dialogue associated to each of them. Interface objects are identified according to the user requirements deduced in step 2 of our approach. Their presentations will, also, be chosen according to a knowledge based system, dealing with guidelines, in order to ensure an ergonomic quality. At this level of abstraction, guidelines are more frequent and it is easier to formalise them and to automate their inference.

We focus, here, on the specification of dialogue component. In fact, having the model of HMS behaviour, it is necessary to define the behaviour of the different graphical objects of the interface, in order to model the human-machine dialogue. In fact, each object has an intrinsic behaviour according to the services it offers. So, we have to model the different evolutions of its states in relation to the interface. This evolution (activation/deactivation and display/ mask) is related to the system's evolution. Thus, as the ICO model [9], we propose to define for each graphical object a control structure. This structure determine the services the object offers and its different possible states. This structure is described by an Interpreted Petri nets like those proposed for HMS behaviour.

Considering the two kinds of graphical objects identified (informational and command objects), we propose two generic control structure (figure 3). The first concerns informational objects. These objects have principally two states : displayed and masked. The control structure of such object will

have two particular transitions. The actions associated to these transitions will ensure the display and the mask of the object.

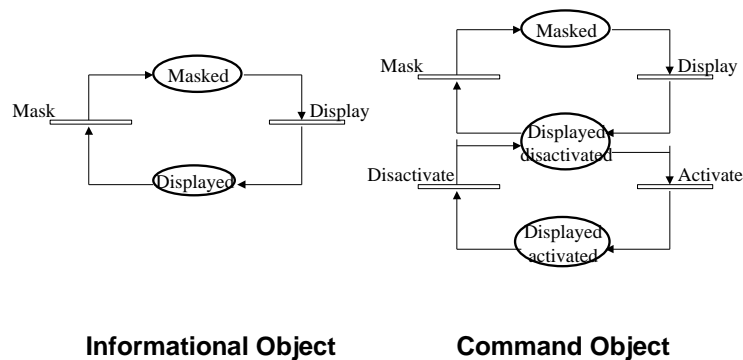


Figure 3 : Control structure of Interface objects

The second type of control structure concerns command objects. A command object has three principle states : masked, displayed/disactivated and displayed/activated. So, four particular transitions are proposed in this structure. One for the display of the object, a second for its mask. The two others are for the activation and deactivation of this object. The principle of dialogue modelling is described by these three points :

- 1- from initial marking of the HMS model, display all informational objects associated to the marked places.
- 2- When a transition becomes passable, all command objects associated to it will be displayed/disactivated. These objects will be activated and disactivated according to the evolution of operator's actions.
- 3- At the end of the execution of operator's task, all the command objects will be masked. The displays will be refreshed according to rule 1.

An operator task is a combination of actions executed sequentially, in parallel manner or alternatively. This is identified by the task analysis elaborated previously. This can precise the order of the activation and deactivation of command objects. In fact, a transition in HMS model, associated to an operator task, is a 'macro-transition'. It resumes a Petri net which is composed principally of a 'macro-place' and two transitions : 'beginning task' and 'ending task'. The command objects will be displayed and masked respectively on passing these transitions. The 'macro-place' is a Petri net modelling the synchronism and concurrency of the execution of different actions of the task (figure 4).

With such decomposition, and on passing the transition 'beginning task', events for displaying in disactivated state, all command objects associated to this task are generated. Then, these objects are activated and disactivated on passing the transitions 'beginning action' and 'ending action'. These objects are masked on passing the transition 'ending task'.



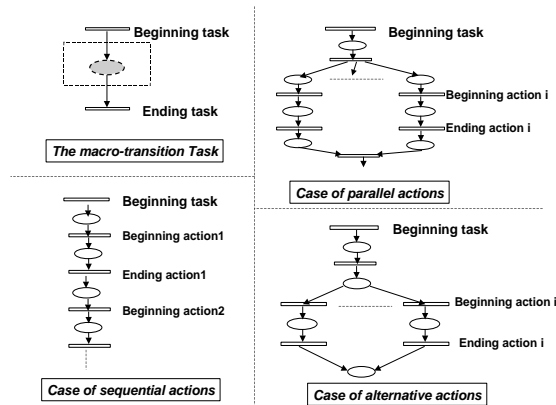


Figure 4 : Decomposition of the “macro-transition” task

The global dialogue is determined by different Petri nets proposed here : Petri net modelling the HMS behaviour and different Petri nets corresponding to the control structures of different graphical objects. The communications between all these nets is insured by production of internal events. The idea is to consider the orders of masking, displaying, activating and disactivating the interface objects as events generated when a token is put on a place or when a transition is passed. The events will, therefore, be produced by the net modelling HMS behaviour. The principle of generating these events is described by these points :

- 1- the put of a token on a place generate events for displaying informational objects associated.
- 2- Passing a transition ‘beginning task’ produce the generation of events for displaying command objects disactivated.
- 3- Passing a transition ‘beginning action’ produce the generation of events for activating command objects needed for this action.
- 4- Passing a transition ‘ending action’ produce the generation of events for disactivating relative command objects.
- 5- Passing a transition ‘beginning task’ generate events for masking all informational and command objects.

Figure 5 illustrates this with the tank sub-system. It contains a Petri net modelling the HMS and 2 Petri nets modelling the behaviour of informational objects, “curve”, associated to the informational variables TC and FC. Two other Petri nets represent the control structure of command objects, “command button”, associated to the command variables fA and fB. The communication is expressed in this figure by arrows in dotted lines :

- .....▶ express the generation of events for displaying the interface objects,
- - - ▶ express the generation of events for activating command objects, and
- - - ▶ express the generation of events for masking informational objects or disactivating command objects.

The communication process between different Petri nets avoid the construction of a global net and the addition of places and transitions.

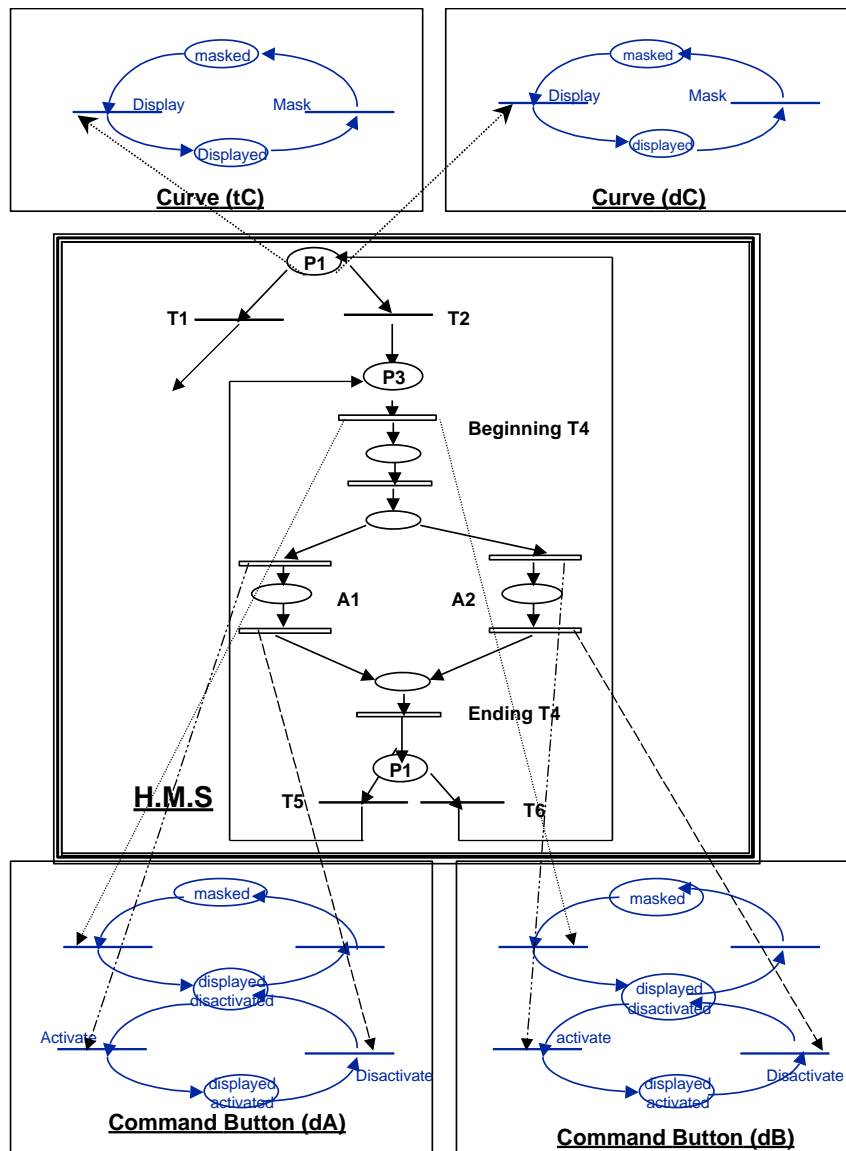


Figure 5 : Communication process between different Petri Nets

## Conclusion

We have focused in this paper on one of the more difficult steps of any global methodology for UI design : the modelling of human-machine dialogue. We have explained, through an industrial application, the principle of modelling the human-machine dialogue using Petri nets.

The next step of this approach concerns the verification of UI specifications before proceeding to the generation its self. We are concentrating, now, on this aspect. We are studying the verification of the Petri nets correctness with the proposed extensions.

## Bibliography

1. Ait-Ameur Y., Girard P., & Jambon F., A uniform Approach for specification and design of interactive systems : the method B, DSV-IS'98, Coesner's House, Abingdon, UK, June 3-5 1998, pp 333-352.
2. D'Ausbourg B., Durrieu G., & Rocher P., Deriving a formal model of interactive system from its UIL description in order to verify and to test its behaviour, in proceedings of DSV-IS'96, springer verlag, pp 105-122.
3. David R., & Alla H., Du GRAFCET aux Réseaux de Petri, Editions HERMES, 500 p, 1992
4. Harrison M., & Thimbelbey, *In formal methods in human computer interaction*, Harrison & Thimbelby (Eds) 90, cambridge university Press, 1990.
5. Jacob R.J.K., A state transition diagram language for visual programming, IEEE Computer, vol. 18 (8), pp 51-59 August 1985
6. Mahfoudhi A., TOOD : Une méthodologie de description orientée objet des tâches utilisateur pour la spécification et la conception des IHM : : Application au contrôle du trafic aérien. Thèse de doctorat de l'Université de Valenciennes et du Hainaut-Cambrésis, Juin 1997
7. Monin J.F., Comprendre les méthodes formelles : panorama et outils logiques, edition masson, Paris, 1996
8. Moussa F., Kolski C., & Riahi M., A model based approach to semi-automated user interface generation for process control interactive applications. *Interacting with Computers*, 12 (3), pp 279-292, 2000.
9. Palanque P., Spécifications formelles et systèmes interactifs : vers des systèmes fiables et utilisables. Habilitation à diriger des recherches, Toulouse I, 14 Novembre 1997.
10. Parnas D.L., On the use of transition diagrams in the design of user interface for an interactive computer system. 24 th ACM Conference 1969, pp 379-385.
11. Paterno F., & Faconti G. On the use of Lotos to describe graphical interaction. In proceedings of people and computer VII, HCI'92 conference, cambridge university press, pp 155-174.
12. Riahi M., Moussa F., Moalla M., & Kolski C., Vers une spécification formelle des interfaces homme-machine basée sur l'utilisation des réseaux de Petri. Colloque ERGO IA'98, Ergonomie et Informatique Avancée, Biarritz, pp. 196-205, 1998.
13. Turner K.J., Using formal description techniques : An introduction to ESTELLE, LOTOS and SDL, edition John, 1993.
14. Williem R., & Biljon V., Extending Petri Nets for specifying Man-Machine dialogues, *Int. J. Man-Machine Studies*, vol. 28 pp 437-455, 1988.
15. Kolski C., & Moussa F., Two examples of "Tools for working with guidelines" for process control field : Synop and ERGO-CONCEPTOR. *Third Annual Meeting of the International Special Interest Group on "Tools for working with guidelines"*, J. Vanderdonck (Ed.), Namur, Belgium, 4 June 1996.