A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

Modelling of cognitive activity during normal and abnormal situations using Object Petri Nets, application to a supervision system

H. Ezzedine and C. Kolski

Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines
LAMIH UMR CNRS 8530,
Le Mont-Houy – University of Valenciennes and Hainaut-Cambrésis
F-59313 Valenciennes Cedex 9 – France

Email: {Houcine.Ezzedine, Christophe.Kolski}@univ-valenciennes.fr
Fax. (+33) (0)3.27.51.13.16

Abstract – This article presents a method for the modelling of cognitive activity using object Petri nets. The method includes the recognition of the various classes of situation (normal and abnormal) which human operators are likely to meet whilst performing their tasks. Each of these classes is described according to the characteristics of the state of the system. We will present the various mental representations used during the control/command activity according to the main aims set by the operator. The examples given come from a project dealing with the integration of a supervision system in a railway traffic regulation room.

Keywords: Petri Nets, activity modelling, design, human-machine systems, specification, supervision, human-machine interaction.

1 Introduction

The progress made in the field of HCI design environments (Meyers 1995; Penner 1993; Fekete and Girard, 2001) has led to a breakthrough as regards the possibilities of modelling and prototyping (Penner et al. 2002; Jacko et al. 2002). In addition, as regards the task of interface specification from the user's point of view, current trends use interactive system architecture models such as the PAC (Presentation, Abstraction, Control) model (Coutaz et al. 2001), the MVC (Model, View, Controller) model (Goldberg 1984), the Arch model (Bass et al. 1991) or variations on these models, consisting in designing the interface using associations between the application's data structures and the interface presentation objects. It is therefore a matter of interface specifications from an implementation point of view, which can be distinguished from the specifications from a cognitive point of view.

As pointed out by Bainbridge (Bainbridge 1981), the behaviour of a human operator in a complex control/command environment depends on the context in which the human and the machine are to be found (Jones et al. 2002). What is more, the operator tasks can prove to be extremely variable in such environments (Rasmussen 1986; Moray 1997).

Based on these observations, one of our research directions is to suggest a method for the modelling of cognitive activity which can be a starting point for the object oriented specification of an HCI. This direction takes into account the different classes of situation the operators are likely to meet. Each one of these classes is described according to the characteristics of the process state to which it corresponds. We then describe the main objectives set for the operator for each of the classes. Following this, the various mental representations used by the operators in their control/command activity must be defined. The resulting model encompasses two parts:

- the first part consists in describing the different goals of the operator when faced with a specific situation (we therefore concentrate on the cognitive aspect of the human operator's work),

- the second part describes the actions triggered by the operator in order to achieve his/her goals and sub-goals.

The description of these two parts is based on object Petri nets. To illustrate our method, we will give examples which come from an industrial project.

2 Global method targeted

2.1 Classic modelling approaches and dynamic industrial environments

The most classic approaches for the modelling of human tasks (or activities), such as HTA (Hierarchical Task Analysis) (Duncan 1981; Stammers et al. 1990; Shepherd 1993), MAD (Method for Analytical task Description) (Scapin et al. 1990; Scapin et al. 2001), Diane (Tarby et al. 1996), CTT (ConcurTaskTree) (Paternò 2000), TKS (Task Knowledge Structure) (Johnson et al. 1991), etc., generally define a task as being a set of goals and states, along with a set of constraints linked to the technical environment. The description of a task is generated using the breakdown of goals into sub-goals and the way in which these goals are achieved. This description is often represented either by a grammar or by one or several models inspired by Artificial Intelligence techniques or software engineering techniques (Diaper et al. 2003; Hollnagel 2003). Several criteria make this approach difficult to use in the dynamic industrial environments which interest us in this article:

- In the existing methods, the task description can be limited to a sequential performance of actions, which is far from being the case in dynamic systems (Cellier et al., 1992).
- In the classic task models, it is most often presumed that the information necessary to achieve a specified goal is included in the description of the problem faced by the user, and that this information does not change dynamically. However, in dynamic situations, the situation can change with time, for example following the appearance of alarms specifying malfunctions (Stanton, 1994).
- Consequently, the fact that the dynamic aspect of the system is not taken into account in the tools available to the operators can influence their activities and make them commit errors (Reason 1990; Dorner 1997).

2.2 Main starting point : Rasmussen's model

The well-known model proposed by J. Rasmussen in 1980 (Rasmussen 1980; Rasmussen 1986) was the reference model we used as our main inspiration for our research work.

To give a brief reminder, Rasmussen proposed a frame of reference called the “decision ladder” which can provide a general description of operator activity in a control room. Rasmussen's “decision ladder” has several sequential levels of information processing (Fig. 1). These include:

- The detection of abnormal events which alert the operator following an alarm or the observation of one or more variables with abnormal values;
- The observation of the set of variables which allows the operator to identify the state of the system given the previous system state and the goals of the operator;
- The identification of a general strategy which will tend to correct the system;
- A set of procedures to carry out the actions, and the actions themselves.

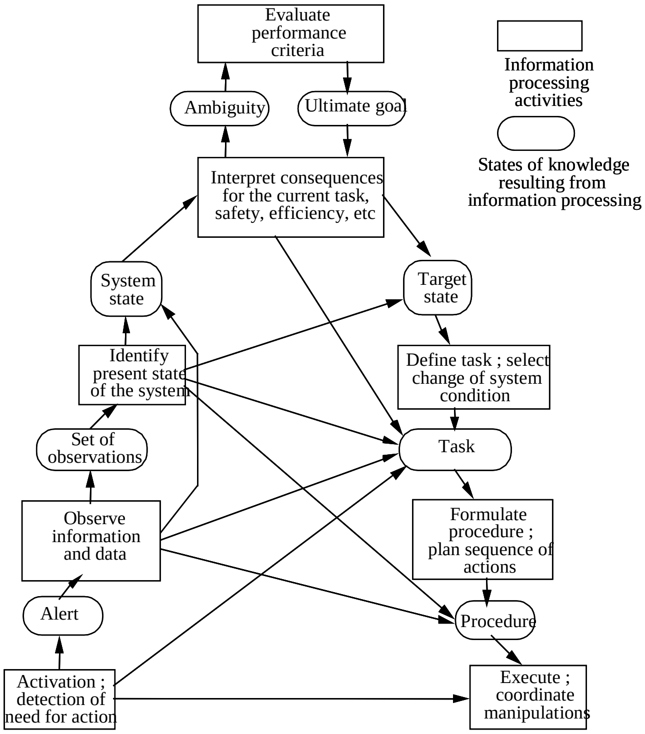

Fig. 1. Rasmussen's Decision Ladder (from Rasmussen 1983, simplified)

We should also remember that Rasmussen identifies three kinds of behaviour:

- The first is skill-based behaviour, the result of lengthy practice, in which the user responds almost automatically with corrective actions in relation to information perceived as signals. These signals provide information about the current state of the environment and allow the experienced user to respond with over-learned automatic actions. The user thus goes directly from an alerted state to the execution of a procedure.
- The second kind of behaviour is rule-based, in which the operator, having identified the state the system is in, chooses an appropriate rule from a set which then identifies the appropriate response. Boy suggests that the knowledge extracted from experts in order to implement expert systems is essentially of this type (Boy 1986).
- The third kind of behaviour is knowledge-based behaviour and is used by the operator when confronted with new or unforeseen situations. In such cases the decisions depend upon really intelligent reasoning, the generation of hypotheses, attempts to verify them and to predict the evolution of the system following whatever attempts are made to manage the situation. The procedures used tend to be memorised and used as rule-based behaviour if the situation recurs.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

Rasmussen’s “decision ladder” is a model characterised by a general cognitive architecture. Such an approach has often been criticised as being too reductionist, too linear and too fixed, in so far as it does not reflect the dynamics of the problems which have to be dealt with. These reasons have led to the creation of derived models. Among those currently involved in this kind of research are Hoc and Amalberti, who have proposed a new approach to modelling which includes diagnosis and decision making in dynamic situations. They have proposed a model involving the temporal dimension (Hoc and Amalberti 1995).

Despite its limitations, the initial Rasmussen model is extremely valuable for computer and automation scientists as it provides a concrete framework for reflection, and situates the global phases followed when solving a problem. This is why we consider it to be the under-lying model used in our research work.

2.3 Modelling process

In (Rasmussen 1993), Rasmussen defined the task as being relative to a situation and not a "thing". Given this point of view, it appears to us to be possible to break the system down into running situations (normal and abnormal) ; this is the aim of step #1 shown in figure 2. Indeed, in complex industrial systems, human operators often have to intervene in different situations which can be normal or abnormal. According to the situations, the information required and the human actions necessary can vary enormously (Millot et al. 1993; Moray 1997; Kolski 1997). It is therefore important that the analysts show up the different situations.

Following this classification, the operator’s task is broken down into two representation units (phase #2 in figure 2) :

- The first unit is called the *Operational Processing Unit (OPU).* In both normal and abnormal situations, it consists in identifying and organising the relevant information concerning the interactions of the human operator with the system (static descriptions), as well as the processing of this information by the operator (dynamic descriptions); this unit represents the operational point of view.
- The second unit is called the *Cognitive Processing Unit (CPU).* For each situation (normal or abnormal), it describes the knowledge necessary (static description) in order to achieve the goals (dynamic description). This unit constitutes the operator task control point of view.

Please note that we consider that the modelling process proposed in figure 2 can be seen as a part of a global design and evaluation methodology. A description of global methodologies can be found for instance in (Piccini 2002) or (Lepreux et al. 2003).

This description is similar to the horizontal and vertical axes described in (Bainbridge 1981) which respectively define the cognitive tasks and processing tasks. The description of the two models, static and dynamic, is done using object oriented analysis techniques and object Petri nets (OPNs) (phase #3 in figure 2). We will concentrate on these OPNs in this article, beginning with a brief presentation in the next part.

This description corresponds to a model (specification) of the HCI. It goes on to deal with the design and production of the human-computer interface, which then has to be evaluated. The design and production stages are not dealt with in this article; readers can refer to (Spolsky 2001; Van Harmelen 2001; Jacko et al. 2002; Shneiderman and Plaisant, 2004) for example, for more information.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

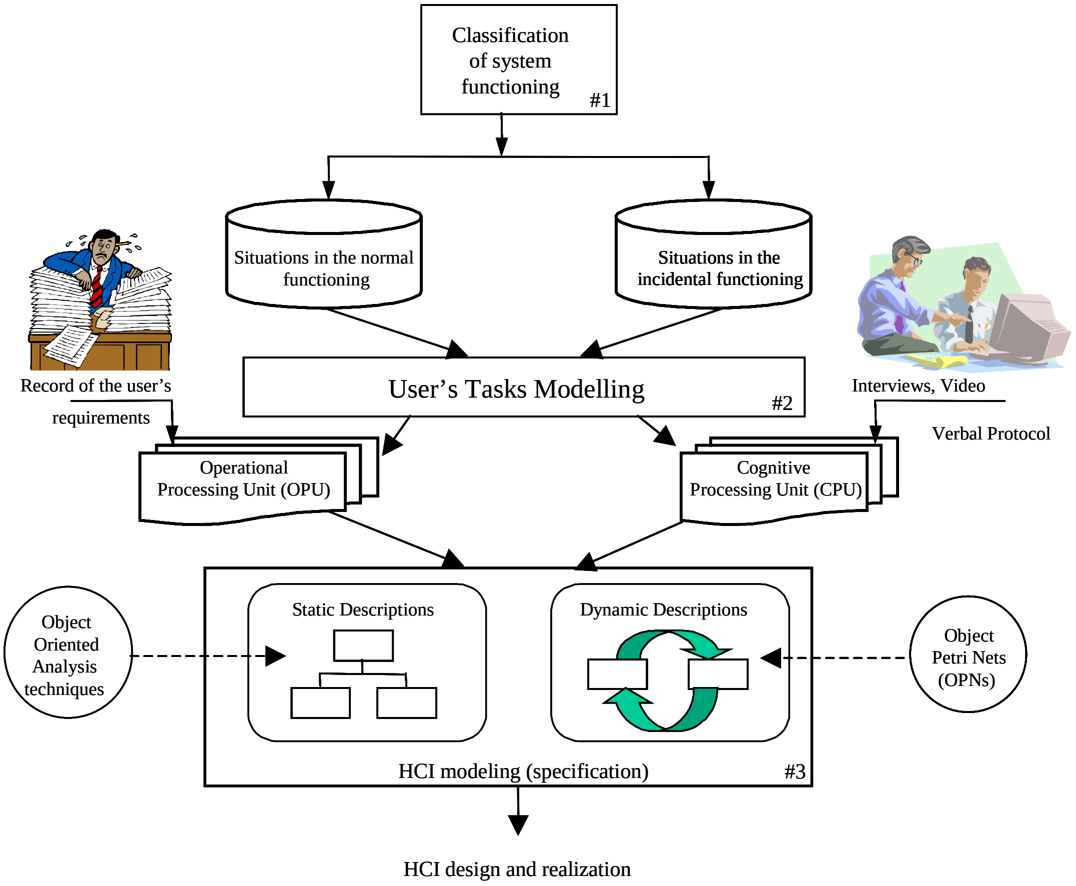

Fig. 2. Modelling process

3 OPNs

OPNs make it possible to model a system according to a set of objects which are associated to a place on the net and which, through their internal state, characterise an overall system state (Sibertin-Blanc 1985; Agha et al. 2001). An analogy can be drawn with a coloured Petri net (David et al. 1994; Jensen 1996) in which the colours of the tokens represent the identifiers of the objects contained in each of the places on the net (Fig. 3).

Objects O1 and O2 can initially be in the "stop" and change to the running state by going through the "Start" transition. Object O1 goes through this transition by calling upon its own start up mechanism (as does object O2).

Before moving on to a formal definition of OPNs, the following definitions concerning the notions of object and class should be kept in mind.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

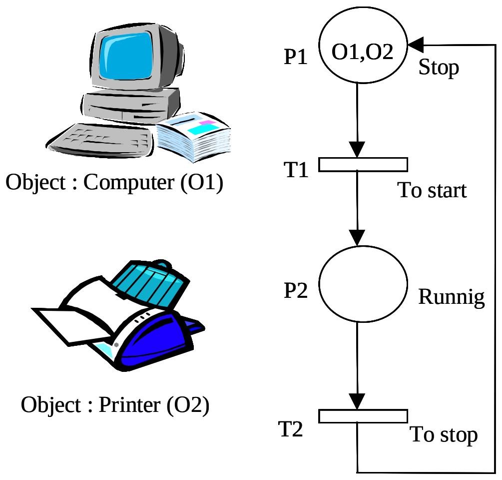

Fig 3. Simple example of Object Petri Nets (OPNs)

3.1 Brief reminder concerning the notions of object and class as regards OPNs

In object oriented design (Delatte et al. 1993; Meyer 1997; Booch et al. 1998), an object incorporates both declarative knowledge, called **attributes** and procedural knowledge, called **methods**. An object has a **state**, defined by the values of its attributes, and the methods are the operators of a change of state.

An object's methods have an effect on the value of its attributes. In this way, they make it possible for the state to evolve according to the actions the object performs.

An object can be seen as being a reusable component providing an **interface** for the outside world (the information and processes it makes available to the other objects) and an internal part, hidden from the environment, which specifies the way in which the object implements the functionalities provided by its interface.

The formalisation of the object concept is most often based on two notions, the first textual and the second graphic, which are used in a complementary manner and are intended to qualify both the static and the dynamic properties of the objects.

As regards the dynamic properties, an operation can be performed sequentially by transferring control of the user object to the used object, following a plan defined by the Operation Control Structure (OpCS). Control is given back to the user object at the end of the operation. An operation can also be performed in a concurrent manner. In this case, control is transferred according to a protocol which depends on both the state of the object and the flux going into the object. This protocol is described in the Object Control Structure (ObCS: see concerning this (Palanque et al. 1997) or (Tabary et al. 2002)) using a notation and semantics equivalent to those of the Ada meeting.

Given that C is an object class, C is defined as the triplet <Id, Attr, Serv>:

Id: is a unique identifier of the class,

Attr: is a set of attributes which characterise this class of object,

Serv: is a set of services or methods {m1,m2,…, mk} associated to the class which make it possible to modify the state of the various objects which come from the class.

3.2 Formal definition of OPNs

Object Petri Nets (OPNs) are a formalism which belongs to the class of high level Petri nets. They were introduced in 1985 (Sibertin-Blanc 1985), and incorporate the manipulation of data in a Petri net: instead of manipulating non differentiated tokens, OPNs manipulate objects.

An Object Petri Net is a septuplet:

R = <P,T,Pre,Post,M0,C, Activation>

With

P: set of places,
T: set of transitions,
Pre: set of pre conditions,
Post: set of post conditions,
M0: original marking,
C: set of object classes,
Activation: set of services belonging to a class which corresponds to a transition.

As in any Petri net, the OPN graph includes two types of node ; the places and the transitions are linked by oriented arcs.

The places: a place is represented by a circle. It can contain objects; several objects in the same class can be found in the same place.

The transitions: a transition is represented by a rectangle. A set of object classes is associated to each transition, each of the object classes indicates a separate crossing possibility.

The arcs: an oriented arc links a place to a transition or a transition to a place. The weight of an arc is a Pre or Post function which establishes a correspondence between each object class (Ck) associated to the transition and the objects (<Oi>) contained in the place (before and after) following the direction of the arc. Compared to generalised PNs, the Pre and Post functions will have a further argument, that is to say the object class identifier.

Thus Pre (Pi, Tj/Ck) and Post (Pi, Ti/Ck) correspond to a formal sum of the couples <Oi>.
Pre (Pi, Tj/Ck) = K1.<O1>+…+Kj.<Oj>
Post (Pi, Tj/Ck) = K'1.<O1>+…+K'm<Om>
With K1(K'1),…,Kj(K'm) being constants and O1,…, Om,…,Oj being the objects contained in the places.

Crossing a transition**:** a transition Tj which has been validated in relation to a class Ck can be crossed. The crossing process which will be referred to as Tj/Ck consists in performing the following three operations simultaneously:

- a quantity of marks (objects) equal to Pre(Pi,Tj/Ck) is removed from each place Pi before Tj
- a quantity of marks equal to Post(Pi,Tj/Ck) is added to each place Pi after Tj
- the service of the selected object is performed.

The OPN tool thus defined will be the basis for the description of the previously mentioned components (arcs, places and transitions).

4 Components of the model obtained

Figure 2 was made up of the following components:

- A description of the system according to various functioning situations,
- A description of the tasks in two units, that is an operational processing unit (OPU) and a cognitive processing unit (CPU),
- Each of these units includes a dynamic description based on OPNs and a static description obtained using an object oriented analysis.

Each of these three components of the model will now be described and illustrated using an industrial control/command application in French railway traffic (the ASTREE project).

4.1 Presentation of the ASTREE project

The ASTREE project deals with a new on-line exploitation of the railway network using the general centralisation of the control-command system. The project must meet many requirements including traffic flow, itinerary management, control of the speed of the mobile elements, regulation, etc. The main principles selected by the SNCF (Société Nationale des Chemins de Fer Français, the French national railway network) are the following (Fig. 4):

- To equip the mobile units with means of location and transmission,
- To install computers linked to the shunting points in order to know what itinerary has been traced in front of the train. These calculators are also capable of questioning the mobile units concerning their position,
- To develop software programs for transport decision making. In this way, according to the algorithmic solutions found, commands will be sent to the immobile installations, while traffic authorisations, safety instructions, and economic driving advice will be transmitted to the mobile units.

These principles must always meet the requirements in terms of safety, running flexibility and savings both in investments and in running costs.

In the project, our role consisted in designing, modelling and evaluating new interactive systems to be made available to the operators in the railway traffic control room (Ezzedine et al. 1994; Kaddouri et al. 1995; Abed et al. 1998; Ezzedine et al. 2004). The project provided us with a field in which to experiment with modelling methods in a rich and complex context.

An analysis carried out of the personnel involved in the project revealed that those who are directly concerned are **the regulators** (in the railway traffic regulation room). Their main task is to monitor the progress of the mobile units in order to ensure their regularity and to take measures to deal rapidly with the consequences of any disruptions (delays of mobile units, incidents, etc.).

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

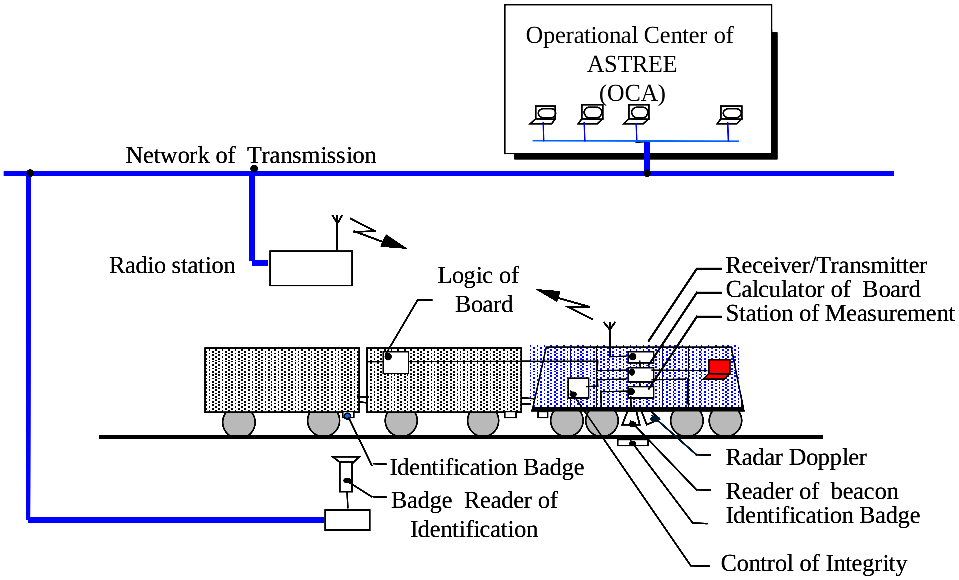

Fig. 4. Architecture of the ASTREE project

4.2 Description of functioning situations

The situations likely to be faced by the operators during their control/command tasks are characterised by two possible states:

- *Normal functioning*: this class of situation covers the system's known and stabilised configurations. In the case of the ASTREE system, a normal situation can be characterised by the respect of the anticipated traffic flow and missions.
- *Incidental functioning*: this class of situation covers the states of system dysfunction (Fadier 1990; Villemeur 1992; Smith 2001). In the case of ASTREE, these situations correspond to delayed trains or trains ahead of time, as well as to human or material conflicts or incidents.

Thus, varied abnormal situations can occur in the ASTREE system. More specifically, these abnormal situations can be related to defects in the ground infrastructure (switch at a junction which does not obey an instruction, broken rail section, creation and validation of maintenance work projects on the lines, suppression of finished maintenance work projects, faulty level crossings,…). For example, in the event of a problem with a switch at a junction, the regulator must notice it and be informed of it (in particular using the information available on the human-computer interface); the regulator must forbid train traffic on the sections involved, and inform the maintenance personnel of the problem. He / she must also seek an intermediate solution so that the trains concerned are able to perform their mission (importance of maintaining service quality). Thus various human activities are performed using the human-machine interface.

The abnormal situations can also be directly related to the management and control/command software programmes in the ASTREE system, in particular those which deal with the traffic plan on the railway network. Several incidents relating to the trains' missions can occur before the running phase (risk of theoretical clashes) or in real time. These types of incident include the following:

- Clashes between missions (simultaneous occupation of the network lines, simultaneous arrival of trains at a station, allocation of one train to two different missions at the same time, etc.)

- Need to make a real time modification to the journey of a train (by slowing it or modifying its itinerary following an incident on the line for example; cf. the abnormal situations relating to the infrastructure mentioned above)

- Creation of a new mission for a train which has not been provided for in the theoretical schedule (prepared six months beforehand) for a one-off mission (for example, a train organised for a VIP: the president, a minister, …).

A generic class characterises incident situations. This class is identified as follows:

- the event at the root of the incident,
- the origin of the incident (human or material),
- the degree of instability of the situation, which is given by the number of trains delayed,
- the location of the incident,
- the constraints linked to the incident.

4.3 Description of the operator's tasks

In a normal situation the on-site analyses performed show that the operator monitors the functioning of the system (train movement).

In an incident situation, the evolution of the situation as far as the operator is concerned goes through the following phases (Cacciabue et al. 1992) :

- Identification of the origin of the disruption, sending people to the spot if necessary (example: there is an object on the track, placed voluntarily or otherwise),
- Management of the disruption after estimating its evolution (example: traffic movements are forbidden temporarily on the track which has been blocked by an object),
- Return to a normal state after dealing with the incident (example: the object is removed from the track),
- Management of the consequences of the disruption (example: certain tracks are closed; the trains must be slowed down or redirected, etc.).

In order to share the tasks to be performed between the human and the machine, a detailed study of all the elementary functions of the HCI was performed. The study gave rise to a modelisation using SADT method boxes (Structured Analysis – Design Technique) (Ross 1977). After the study, the interventions of the human operator and the machine (the calculator) were registered as "mechanisms" or "means" for each of the boxes (in the bottom part of the boxes). The result found corresponds to obtaining a functional analysis of the human-machine system (Fig. 5 and 6).

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

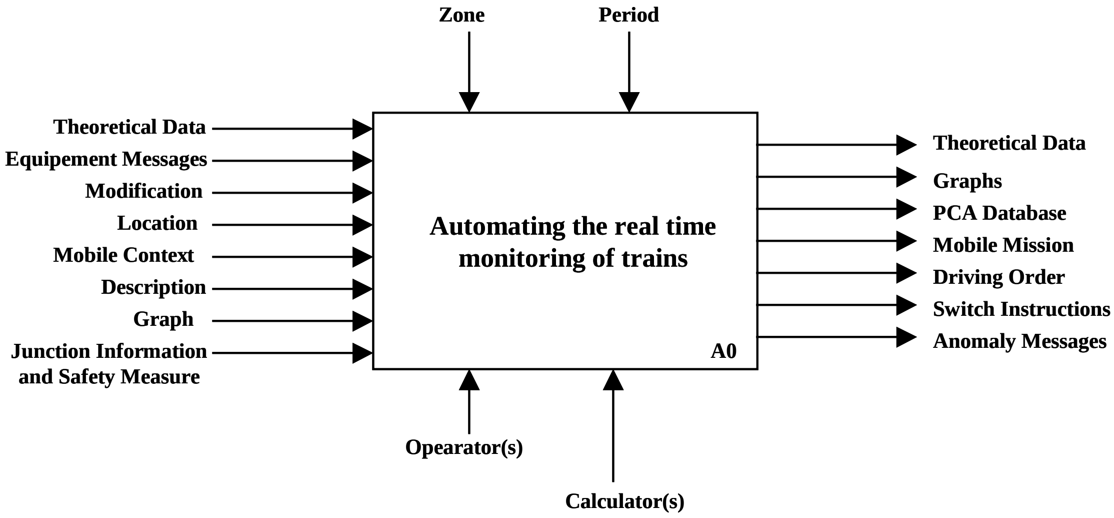

Fig. 5. Functional Analysis of the ASTREE project (mother box : general view)

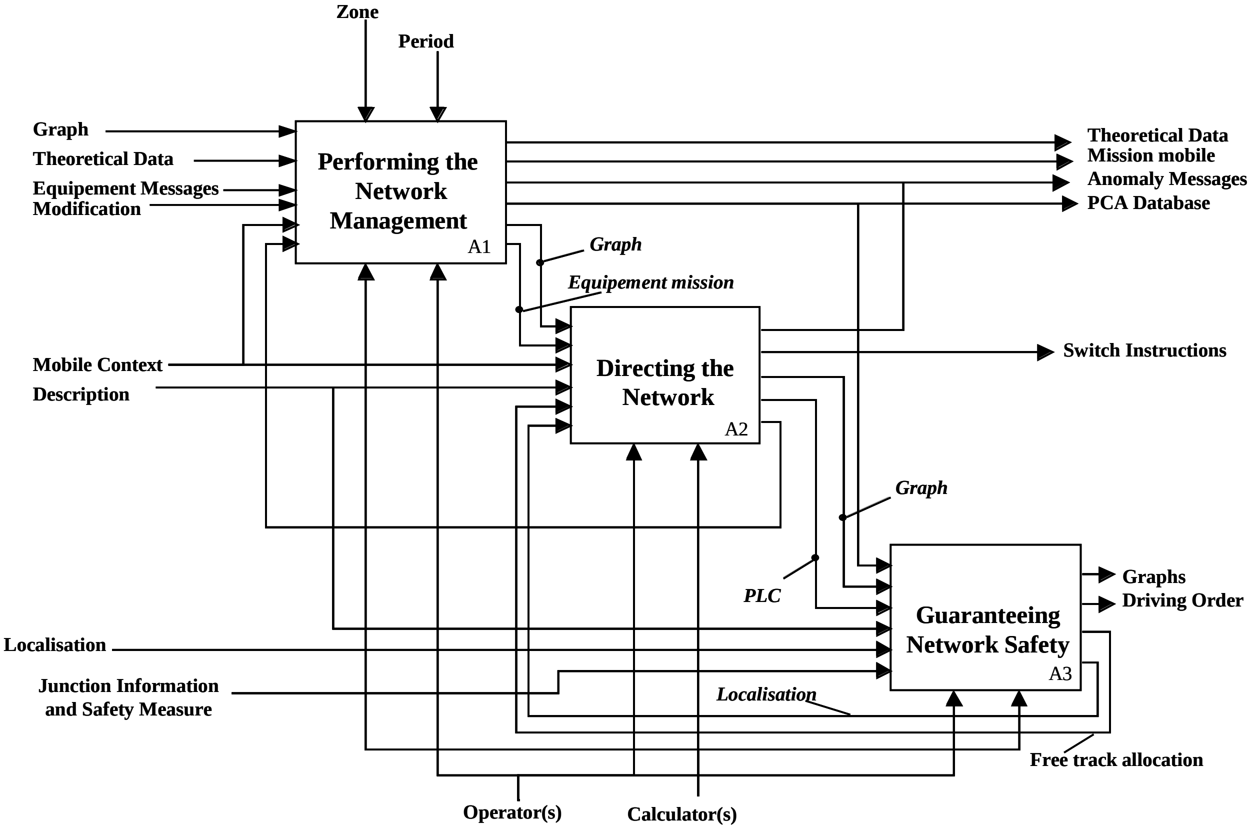

Fig. 6. Functional analysis of the ASTREE project (Daughter boxes)

It appears in general that the role given to the operators in the posts is that of directing and supervising the running of the trains in order to guarantee their regularity. This includes:

- the role of programming and supervision, which corresponds to integrating all movements arising out of current conditions and any short term traffic needs which may occur,

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

- the role of programming various maintenance work projects involving the railway ground infrastructure,
- the role of solving problems in the event of dysfunctions (incidents, accidents) in the process and blockages in the driving calculators.

Therefore two types of intervention can be performed : the *a priori* type (for example establishing the theoretical schedules of railway traffic and the timetable of the on-board personnel) and the real time type (for example limiting the consequences of delays).

Within the framework of the ASTREE project, the functions of the human-machine system that the human operator has to perform are those which are not easy to automate. So, when there are "anomalies" at the exit of a functional SADT box used for the driving calculators, they correspond to the entries to other SADT boxes which are used for the human operators.

The tasks thus noted especially concern the unblocking of the automated process when there are no solutions according to the algorithms being used (for example, the automated system in normal running mode is unable to integrate and validate a convoy in the transport plan according to the characteristics which have been given to it), the resolution of problems (during the initialisation of convoys or if there are delays, disruptions or incidents, etc.), the surveillance of traffic progress in order to anticipate certain decisions, the dialogue to integrate convoys and/or last minute maintenance work projects, etc.

The tasks given to the operator were analysed using the SADT tool. The means which must be made available to the operators were also defined (telephone, computer, radio). The first levels of analysis are shown in figures 7 and 8.

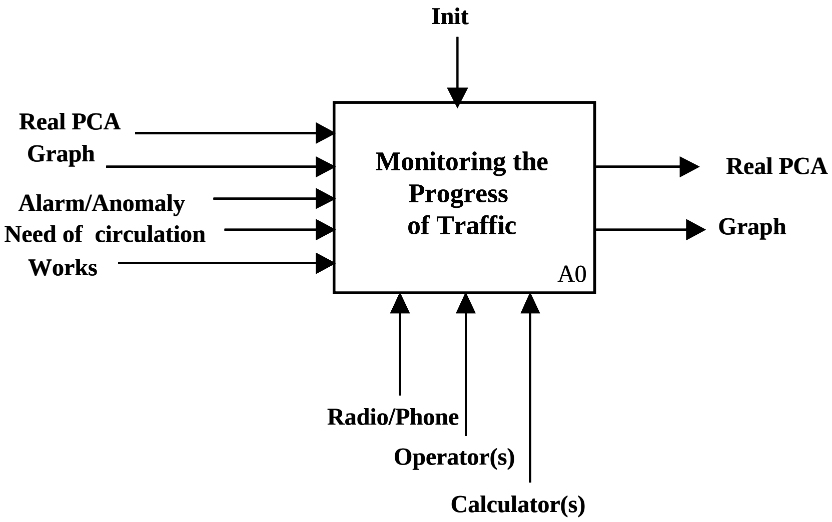

Fig 7. Global analysis of the operator's tasks (mother box – general view)

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

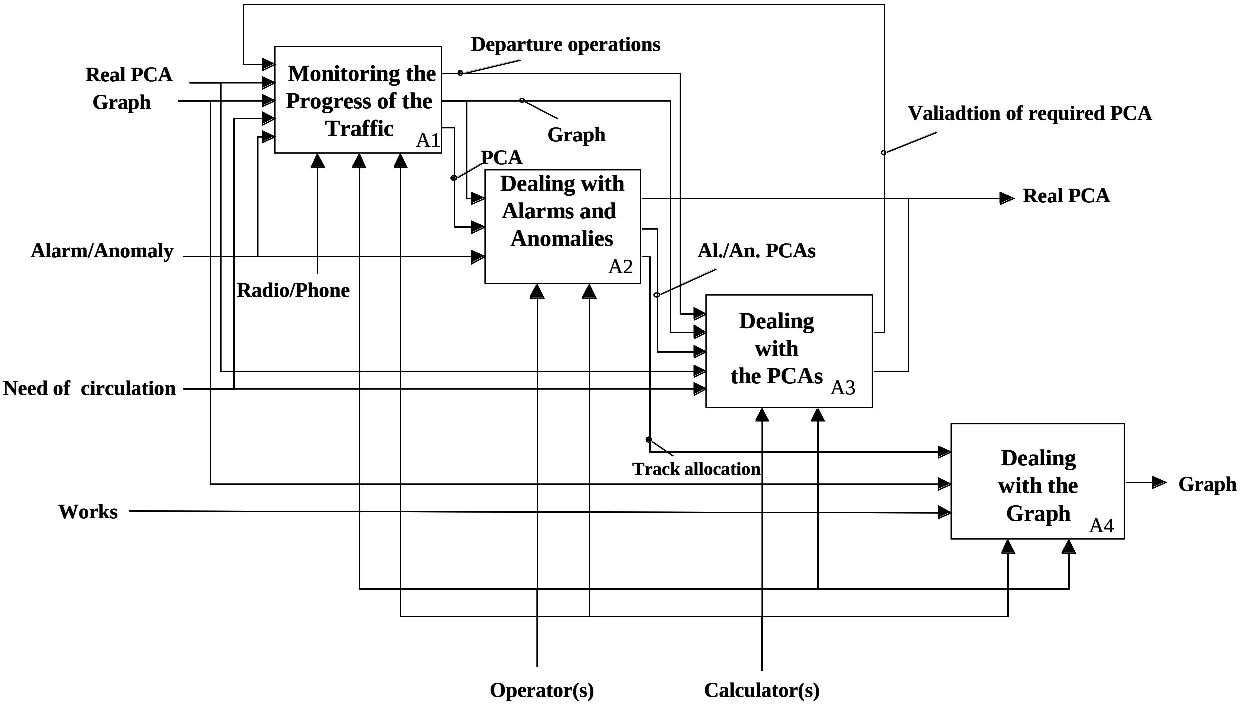

Fig. 8. Global analysis of the operator's tasks (daughter boxes)

Thus, from this classification of phases, we identify the different situations faced by the operator, and then describe his/her activity for each one. This activity is described using task models. The task is modelled according to two components: the first describes the operator's actions on the interface's conceptual objects, the second specifies the goals and sub-goals the operator wishes to achieve.

These two units are described following two representations: the first one static and the other one dynamic. These representations will now be presented.

4.3.1 The static model

The object classes and situation classes are structured in an arborescent manner according to the user's point of view.

In the case of ASTREE for example, the object *train mission* (also called PCA for Plan de Circulation ASTREE in french, translated by: ASTREE Traffic Plan) can be broken down into two objects: (1) the journey route (which corresponds to a set of places serviced) and (2) the set of technical characteristics of the train (train number, engine number, etc.) (Figure 9).

This implies that an action such as the modification of a mission must be broken down into a route modification and/or a modification of the technical characteristics of the train. This modification is performed according to the point of view and the experience of the network regulator. Concerning this aspect, it appeared that the regulators' reactions can be very different to those anticipated (imagined) at the start by the HCI designers (who can suggest other alternative actions which seem to be logical given their functional analysis of the human-computer system).

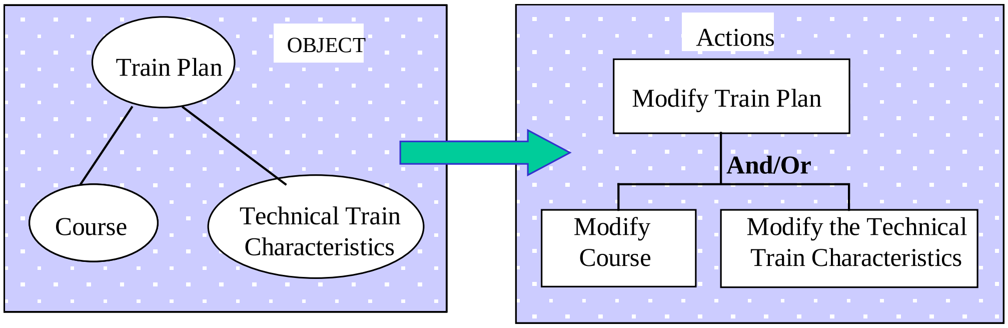

Fig. 9. Break-down of a "train mission" object following the user's point of view.

The conceptual objects of the interface (PCA, mission, maintenance work, alarms, etc.) are deduced from the information given in the file of user needs provided by the system designers. The classes of situation are identified using observations of operators in work situations, interviews and questionnaires (according to methods taken from ergonomics (Wilson et al. 1996)).

The two types of object defined in the static model represent the task context. They can be described by models which come from object oriented analysis (see the Unified Modelling Technique, UML (http://www.omg.org/uml)). This classic aspect in object modelling is not dealt with in detail in this article. Numerous modelling examples are to be found in (Booch et al. 1998) for instance.

4.3.2 The dynamic model

The dynamic model is a complement to the static model (Fig. 2). The dynamic aspect of the two units, the OPU and the CPU, is described by OPNs. The context is defined as the set of objects manipulated by the two networks (Fig. 10). The initial state of the task is defined as being the set of initial states of the context.

The OPNs which model the OPUs and the CPUs, describe the goals to reach following the appearance of an incident situation represented in the form of an object whose services form the strategy which must be followed by the operator in order to return to a normal state of the system. A sub-goal of the CPU model is achieved via the triggering of an action on an object from the OPU model. As long as the operator's action on the conceptual object continues, the OPN of the CPU remains in a "standby" state; at the end of the action on the object, the OPN of the CPU goes on to the next sub-goal. This description is in fact a reflection of the operator's cognitive process following abnormal situations.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

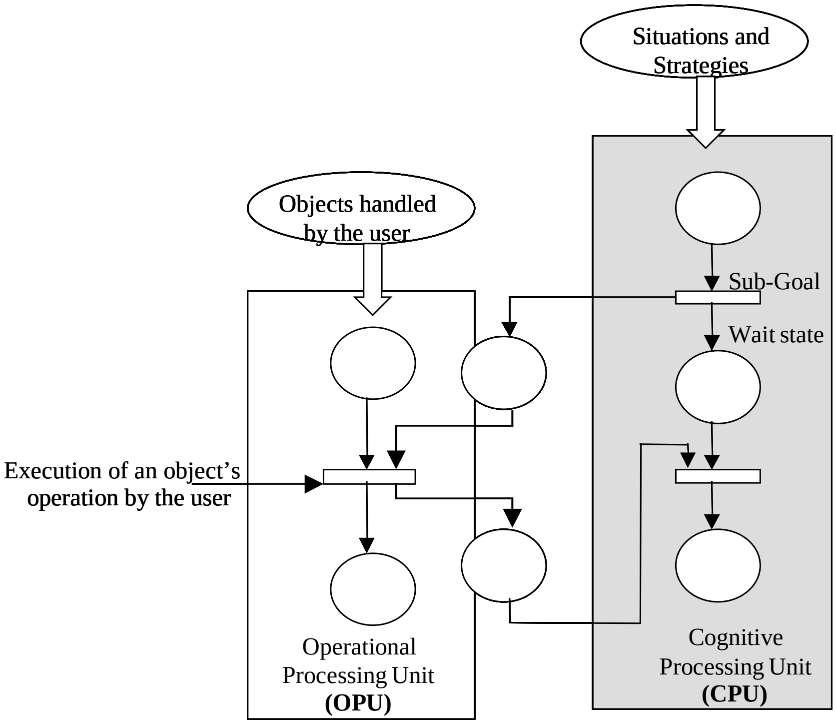

Fig. 10. Dynamic description of the OPUs and CPUs (Basic principles)

In the case of the ASTREE project, the analysis of the data resulting from the observation of the regulators' work revealed three main elements which guided the strategy adopted by each regulator for a return to a normal system state. These elements are:

- The reduction of the impact of the incident on the delays caused to other trains,
- The reduction of the uncertainty of data concerning the true situation,
- The necessity of maintaining control of the traffic.

Following the example of these elements, the management of the incident situation, in relation to the evolution of the information available to the regulator, is defined as followed:

- Taking of preventive measures (speed limits, diversion of a train, etc.),
- Progressive restriction of the uncertainty as regards the authenticity of the event at the root of the incident by setting up an information network made up of several actors (correspondants in stations, use of a reconnaissance train, listening to demonstrators' messages, etc.) ,
- End of the representation of what is at stake in the situation.

An example of an incident situation is the loss of the natural order of priority of trains (in other words, a deviation from the forecast scheduling of train arrivals at a junction). An example of modelling using OPNs is given in Fig. 11. This example is taken from a conflict situation between two trains arriving at a junction (point). One of the trains is late compared to the scheduled plan which leads it to lose its priority at the shunting point. The model of the regulator's activity as regards this situation can be summed up as follows:

- Firstly, following the display of a conflict message (input "object conflict" in the CPU box), the regulator checks the state of the network (and more precisely the state of the junction point in

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

question). This is described by the action of the regulator represented by transition T1. This sets off the unit OPU1 which has "point object" as its context.

- The transition T13 contained in OPU1, activates the display service responsible for the characteristics of the point attached to the "point object" and its start-up time. Indeed, this time is vital to the decision taken by the regulator as it enables him/her to have an accurate idea of the time available for action.
- The regulator’s next aim is the projection of the rail traffic at a given moment in order to perfect his/her mental representation of the state of traffic at that moment and to evaluate the situation along with the consequences of actions he/she might take on the network. This phase is triggered by transition T3 which in turn sets off the OPU2. Using transition T14, the OPU2 activates the "visualize traffic" service attached to the ATP object.
- Then the next stage consists in evaluating the delay following this projection of train traffic. This is described by transition T5 which sets off the OPU3. The OPU3 triggers the "calculate the delay at this moment" service attached to the ATP, and this calculation is performed by the ASTREE operating system.
- As from transition T6, the CPU is found in state E7 which characterizes a situation of choice between two alternatives: either "limit the train’s speed" defined by transition T8 which will set off OPU5, or modify the running of the train, characterized by transition T7 which sets off OPU4. In both cases, the train no longer has priority at the point in question which is the conflict object.
- Transitions T11 and T12 finish the regulator’s task and make it possible to validate the modifications applied to the ATP of the delayed train and thus its priority at the level of the junction point.

The conflict has therefore been resolved using interactions between the cognitive and operational units which represent the regulator’s task model.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

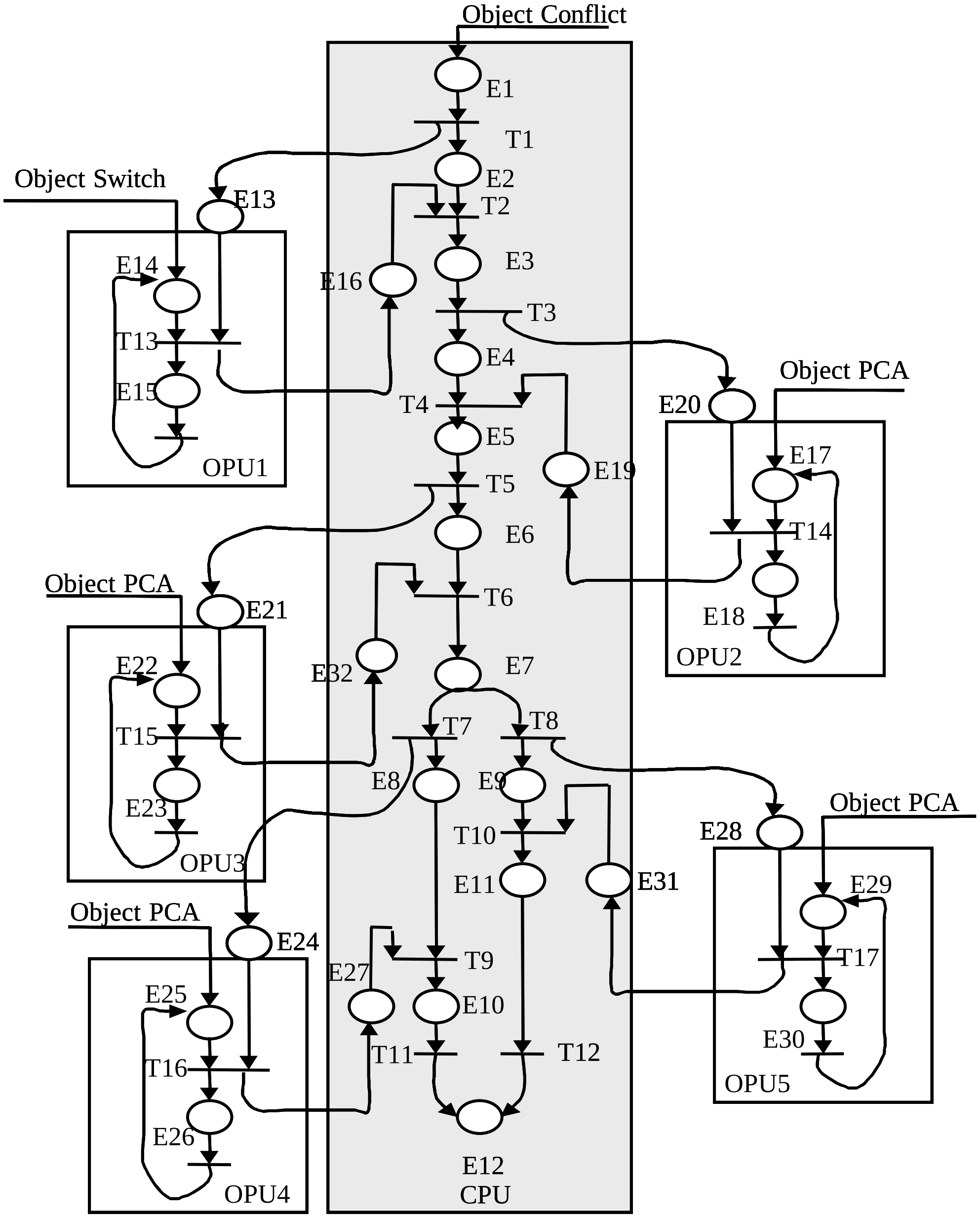

Fig. 11. Dynamic model of a situation of priority loss
(Example taken from the ASTREE project (Kaddouri et al. 1995))

4.3.3. Example of the resolution of a problem modelled using Petri Nets

Figure 12 shows the modelling of human operator activity using Petri nets and according to a representation by OPU and CPU. This model concerns the resolution of a traffic problem related to the creation of train traffic plans (originally shown up by Benaissa et al 1993). On the same figure, we show the theoretical task (imagined by the designers) which is the result of a prior analysis (interviews, questionnaires, observations…) and the real activity following what is actually done by the interactive system.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

For the modelling of the real activity, objective data is automatically gathered (catching of user events and HCI, video); subjective data is gathered by questioning those involved.

Errors can be located by using such a comparison. Figure 12 shows that some subjects, after having noticed that the data they have input is incomplete, recall the input procedure but do not repeat the whole of the input task because they suppose that all the data previously entered has been saved (an idea which is incompatible with the principle of default values). In this case they merely complete the input by following it with a validating action.

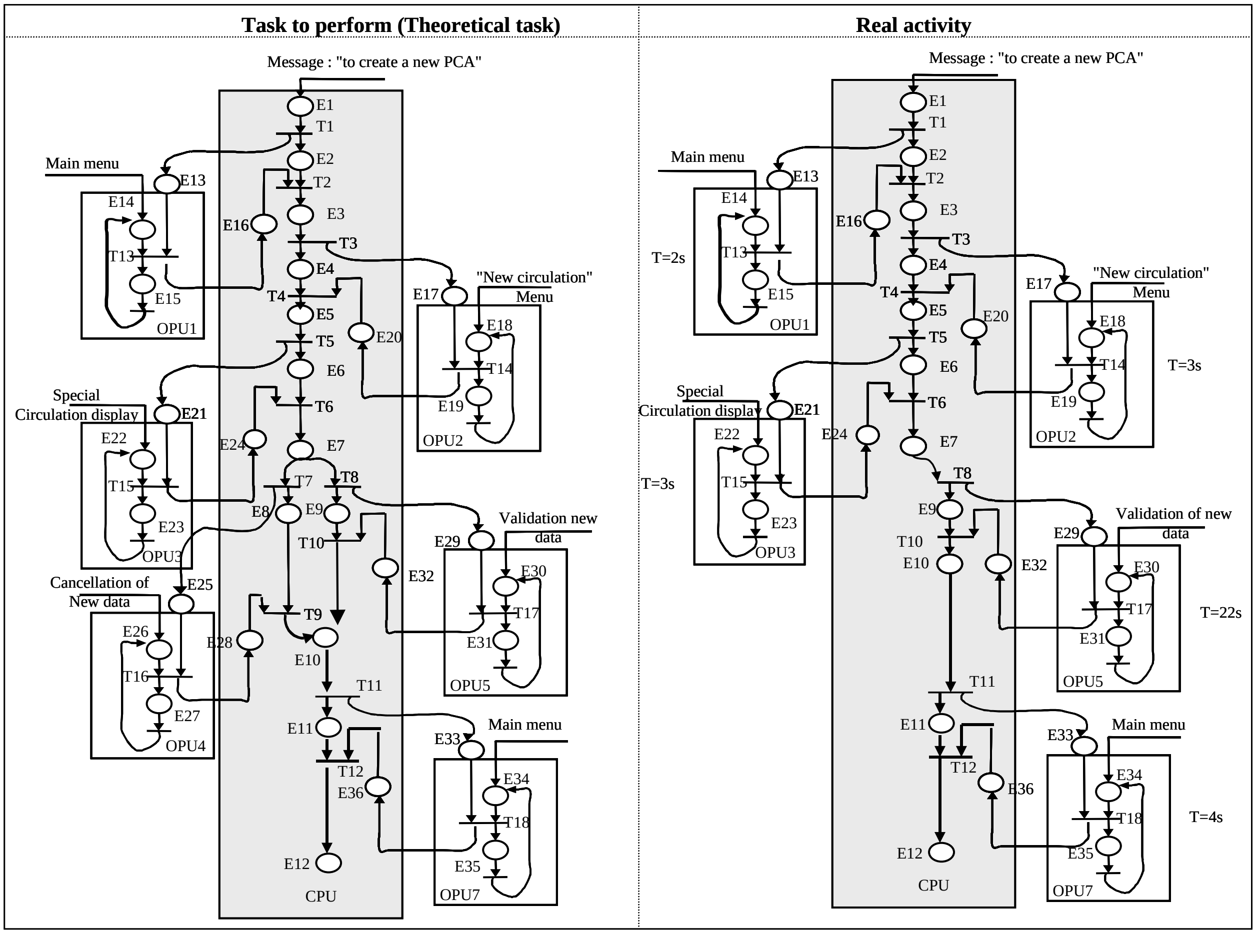

Fig 12. Modelling of the theoretical task and the real activity of the human operator by OPU and CPU.

4.3.4. Limits of this model

Petri nets are a formal tool mainly used to model discrete systems with parallel development (Peterson 1981), which is important for supervision interfaces. Indeed, the state graphs and the event diagrams (which are well known in computer science) have no mathematical formalisms which are adapted to the complex activities we are interested in; we should also underline their limitation as regards their graphic representation for parallel situations and interruptions. These drawbacks quickly added to the attraction of Petri nets which increasingly became a strict tool in the field of the dynamic description of user tasks (Abed 1990 ; Palanque and Bastide 1997 ; Kontogianis 2003).

Indeed, Petri nets make it possible to model various behaviours, parallelism, synchronization, and they provide a clear and legible graphic representation (Alla and David 1997). The association of notions of colour (coloured Petri nets (Jansen 1996)) and especially of the object approach (Object Petri nets (Sibertin-Blanc 1985) to the same formalism considerably added to their power of expression; they now constitute the most widely used tool in the field of HCI dynamic behaviour modeling. In addition, the use of Petri nets is preferred for the resolution of problems of asynchronism of events but the formalism has a drawback: it can result in a bulky and therefore illegible model for complex applications. This is also the case for our proposition based on OPUs and CPUs corresponding to PNs and themselves forming the global PN; however, we feel that this way of structuring the global PN in OPUs and CPU makes the model more legible.

If we refer to the three types of behaviour identified by Rasmussen (mentioned in §2.2), the model proposed is highly adapted to situations, either normal or abnormal, which require skill-based or rule-based behaviours. However, for unusual situations which require the human operator to adopt a knowledge-based type of behaviour, the proposed model is only useful if the situations have been identified *a priori* by the analysts, modeled and then associated to a specification for an interactive tool which is truly useful and usable, and likely to help and back up the human reasoning process. On the other hand, if the situations have gone (completely or partially) unnoticed by the analysts, the human operator is likely to be unprepared when faced by them. It is a classical problem in complex industrial systems; it is compensated in the best of cases by the experience and ability to adapt of the human operators or by regular training sessions organized for them.

5 Related work and discussion

As in the work on the TOOD method (Task Object Oriented Design ; (Abed et al. 1998; Tabary 2001)) and the ICO (Interactive Cooperative Objects ; (Palanque 1997; Palanque 1992; Palanque et al. 1995)), Petri nets are used to model human tasks in a HCI context ; this modelling method then proves to be useful for the specification of the interactive system. However, in the method dealt with in this article, unlike the methods mentioned above, the emphasis is placed explicitly on the notion of situation, leading to the identification of normal and abnormal (incident) situations.

Gomes et al. (2001) propose an interesting approach based on reactive Petri nets (inherited from coloured Petri nets) for interface specification but do not mention cognitive aspects modeling.

Kontogianis (2003) chooses to use coloured Petri nets for ergonomic task analysis and modeling with emphasis on adaptation to system changes. Even though the examples given by the author do not aim especially at the specification of human-computer interfaces, and the type of Petri net is not the same, the principles used show similarities with our method as regards the modelling of different types of activity (synchronization, choice, parallelism, …).

According to the same principle as in the research work done by Abed (Abed et al. 1992; Abed 2001), this modelling method using OPNs of the OPUs and CPUs contributes towards obtaining a model of the regulator’s activity which can be compared with the task model obtained *a priori* during the object oriented analysis of the human-machine system. This leads to comparing two Petri nets : it thus becomes possible to identify the differences (and therefore the errors, missing elements, etc.), and then to deduce possible improvements to be made to the interactive system (since it has been possible to compare the real activities with those originally planned by the designers and which may prove to be mistaken).

Along the same line of thought as that presented in several research projects on predictive task models (such as (Kieras 2003) or (Gray et al. 1992)), the method provides a support which has the potential of facilitating the implementation of predictive evaluations (before realization) of the HCI by the end user.

The method presented has the advantage of clearly separating the actions represented by the Operational Processing Units (OPUs) and the decisions made by the operator following a given situation represented by the Cognitive Processing Units (CPUs). This separation shows the effect of the aims set for the operator on the technological process defined by these variables. Moreover, from this description, the complexity of the operator's task can be measured using the CPUs and their effects on the OPUs. The task description with the OPNs gives a formal aspect to the model defined in this way and can lead to its simulation in order to have a prior evaluation of the HCI and thus save time in the development cycle of an interactive system.

Finally, it could be interesting to underline the fact that the traditional methods for the modeling of human operator activity using PN tools but which represent cognitive and operational activity on the same net, are thus able to mask the parallelism between these two types of activity; our contribution consists in dissociating the two and representing them separately. Our modeling method could be compared and form an analogy with a model of the activities of robots cooperating in an automated industrial process.

6 Conclusion

In this article, we have suggested a method for the modelling of human activity, based on the prior description of normal and abnormal situations during which human operators will have to interact with the interactive system. This method is based on object Petri nets.

The method was used during an industrial project concerning the integration of a supervision system in a railway traffic regulation room. The method is part of an active international research trend leading to the exploitation of Petri nets in the field of design and evaluation of interactive systems.

The first results have shown the advantages of the method for the formal aspect given to task modeling, which is then useful during the specification phase especially because of the separation between the OPUs and the CPUs.

There are many research perspectives which can be highlighted. In-depth thought is needed concerning the way in which help can be provided to the development teams in order to: (1) deduce and then build the OPUs and CPUs as quickly as possible, (2) forecast the difficulties users may meet when performing tasks. It is also necessary to study the way of connecting knowledge of software ergonomics to the model in order to generate automatically or semi-automatically all or part of the interactive system (along the same line of thought as the current research trend: « Tools for Working With Guidelines », which aims at systems based on knowledge in the design and evaluation of interactive systems (Vanderdonckt, 1999; Vanderdonckt et al. 2000; Moussa et al. 2000; Riahi 2004).

Acknowledgements

The authors would like to thank the SNCF, and also the FEDER, the GRRT and the Nord-Pas-de Calais regional authorities which have provided support for this research (SART and NIPO projects).

References

Abed M (1990) Contribution à la modélisation de la tâche par outils de spécification exploitant les mouvements oculaires : application à la conception et l'évaluation des interfaces homme-machine. Ph.D. Thesis, University of Valenciennes and Hainaut-Cambrésis, France, September.

Abed M. (2001) Méthodes et modèles formels et semi-formels pour la conception et l'évaluation des systèmes homme-machine. HDR Thesis, University of Valenciennes and Hainaut-Cambrésis, France, May.

Abed M, Bernard J.M, Angué J.C (1992) Method for comparing task model and activity model. Proceedings 11th European annual conference Human Decision Making and Manual Control, Valenciennes, France.

Abed M, Ezzedine H (1998) Vers une démarche intégrée de conception-évaluation des systèmes Homme-Machine. Journal of decision systems, Vol. 7, pp. 147-175.

Agha G.A, De Cindio F, Roeznberg G (2001) Concurrent object oriented programming and Petri Nets. Advances in Petri Nets. LNCS, Springer.

Bainbridge L (1981) Mathematical equations or processing routine?, In Human Detection and Diagnostics of System Failures, J. Rasmussen and W.B. Rouse (Eds.), Plenum Press, London, pp. 259-286.

Bass L, Little R, Pellegrino R, Reed S, Seacord S, Sheppard S, Szesur M (1991) The Arch model: Seeheim revisited. Proceedings of User Interface Developpers'Workshop, Seeheim.

Benaïssa M.L, Ezzedine H., Angué J.C (1993) An interface Specification Method for industrial processes. XII.European annual conference on human decision making and manual control. 22-24 juin 1993 - Kassel – Germany.

Booch G (1993) Object oriented analysis and design with application, Benjamin/Cummings (Eds), Redwood City, CA.

Booch G, Rumbaugh J, Jacobson I (1998) The Unified Modeling Language User Guide. Addison Wesley.

Boy G (1986) An expert system for fault diagnosis in orbital refueling operations. Proceedings 24th Aerospace Science Meeting, AIAA'86, Reno, Nevada, January.

Cacciabue P.C, Decortis F, Drozdowicz B, Masson M, Nordvik J.P (1992) COSIMO: a cognitive simulation model of human decision making and behavior in accident management of complex plants. IEEE Transactions on Systems, Man and Cybernetics, 22 (5): pp. 1058-1074.

Cellier JM, Eyrolle H, Mariné C (1992) Expertise in dynamic environments. In J. M. Hoc et E. Hollnagel (Eds.), Cognitive engineering in dynamic environments, Academic Press, Londres.

Coutaz J, Nigay L (2001) Architecture logicielle des systèmes interactifs, In Kolski C. (Ed.). Analyse et Conception de l'IHM. Interaction Homme Machine pour les SI, Volume 1, pp. 207-246. Paris: Éditions Hermes.

David R, Alla H (1994) Petri nets for modeling dynamic systems, a survey. Automatica, vol. 30, pp. 175-202.

Delatte B, Heitz M, Muller J.F (1993) HOOD Technical Group, *HOOD Reference Manual 3.1*. Masson and Prentice Hall.

Diaper D, Stanton N (2003) Handbook of Task Analysis for Human-Computer Interaction. London: Lawrence Erlbaum Associates.

Dorner D. (1997) The Logic of Failure: Recognizing and Avoiding Error in Complex Situations. HarperCollins Publishers.

Duncan K.D (1981) Training for fault diagnosis in industrial process plant. In J. Rasmussen and B. Rouse (Eds.), Human Detection and Diagnosis of system Failures, Plenum Press, New York.

Ezzedine H, Benaissa M.L, Angué J.C (1994) An interface design method for railway traffic processes. Proceedings WCRR'94: World Congress on Railway Research, Paris, pp.1233-1239.

Ezzedine H, Kolski C (2004) Démarche d'évaluation d'IHM dans les systèmes complexes, application à un poste de supervision du trafic ferroviaire. Revue d'Interaction Homme-Machine (Journal of Human-Computer Interaction), 5, pp. 91-122.

Fadier E (1990) Fiabilité humaine: Méthodes d'analyse et domaines d'application. In J. Leplat et G. De Terssac (Eds), Les Facteurs humains de la fiabilité dans les systèmes complexes, Ed. Octares.

Fekete JD, Girard P (2001) Environnements de développement de systèmes interactifs, In Kolski C. (dir.), Environnements évolués et évaluation de l'IHM, Interaction homme-machine pour les SI 2, pp. 23-52, Hermes, Paris.

Goldberg A (1984) Smaltalk-80, the interactive programming environment, Addison-Wesley.

Gomes L, Barros JP, Coasta A (2001) Man-machine interface for real-time telecontrol based on Petri nets specification. In T Bahill, FY Wand (Eds.), IEEE SMC 2001 Conference Proceedings (e-Systems, e-Man and e-Cybernetics), Arizona, USA: IEEE Press, pp. 1565-1570.

Gray W.D, John B.E, Atwood M.E (1992) Project Ernestine: a validation of GOMS for prediction and explanation of real-world task performance. Human-Computer Interaction, 8, pp. 237-259.

Hoc J.M., Amalberti R. (1995) Diagnosis : some theoretical questions raised by applied research. Current Psychology of Cognition, 14 (1), pp. 73-101.

Hollnagel E (Ed.) (2003) Handbook of cognitive task design. Mahwah, NJ: Erlbaum.

Jacko J.A, Sears A (2002) The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications (human factors and ergonomics). Lawrence Erlbaum Associates.

Jensen K (1996) Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use. 2nd edition, vol. n°2, Springer-Verlag.

Johnson H, Johnson P (1991) Task Knowledge Structures: Psychological Basis and Integration into System Design. Acta Psychologica, 78, pp. 3-26.

Jones P.M, Mitchell C.M (2002) Model-Based Cognitive Engineering in complex systems, IEEE Transactions on systems, man, and Cybernetics-Part A: Systems and humans , vol. 32, n°1, pp. 2-4.

Kaddouri S.A, Ezzedine H, Angué J.C (1995) Task modelling using object Petri Nets , In Anzaï Y., Ogawa K., Mori H. (Eds.), Symbiosis of Human and Artefact, HCI International'95: 6th International, Tokyo, Japan. (pp. 988-994). Amsterdam: Elsevier.

Kieras D (2003) GOMS models for task analysis. In D. Diaper and N. Stanton (Eds.), Handbook of Task Analysis for Human-Computer Interaction. London: Lawrence Erlbaum Associates, pp. 83-116.

Kolski C (1997) Interfaces homme-Machine, application aux systèmes industriels complexes. Hermes, Paris.

Kontogiannis T (2003) A Petri Net-based approach for ergonomic task analysis and modeling with emphasis on adaptation to system changes. Safety Science, vol 41 n°10, pp. 803-835

Lepreux S, Abed M, Kolski C (2003) A human-centred methodology applied to decision support system design and evaluation in a railway network context. Cognition Technology and Work, 5, pp. 248-271.

Meyer B (1997) Object-Oriented Software Construction, Second Edition. ISE Inc., Santa Barbara, Prentice Hall Professional Technical Reference.

Meyers B (1995) User interface software tools. ACM Transactions Computer Human Interaction, vol.2, n°1, pp 64-103.

Millot P, Debernard S (1993). Men-Machines cooperative organizations : methodological and practical attemps in air traffic control. Proceedings IEEE Conference on systems, Man and Cybernetics, Le Touquet, France, volume 1, pp. 695-700, October 17-20.

Moray N (1997) Human factors in process control. In Handbook of human factors and ergonomics, G. Salvendy (Ed.), John Wiley & Sons, INC., pp. 1944-1971.

Moussa F, Kolski C, Riahi M (2000) A model based approach to semi-automated user interface generation for process control interactive applications. Interacting with Computers, 12, pp. 279-292.

Palanque P (1992) Modélisation par objets coopératifs interactifs d'interfaces homme-machines dirigées par l'utilisateur. Ph.D. Thesis, University of Toulouse 1, France.

Palanque P, Bastide R (1997) Synergistic modelling of tasks, system and users using formal specification techniques. Interacting With Computers, 9 (12), pp. 129-153.

Palanque P, Bastide R, Sengès V (1995) Task model–system model: towards an unifying formalism. In proceedings of the HCI international (EHCI'95), Chapman & Hall, pp. 189-212.

Paternò F (2000) Model-Based Design and Evaluation of Interactive Applications. Springer, Milan.

Peterson J.L (1981) Petri net theory and the modelling of systems, Englewood diffs, NJ: Prentice-Hall, 1981.

Penner R.R (1993) Developing the process control interface, in Engineering for human computer interaction, J.A. Larson and C. Unger Eds.), New York: Elsevier, pp. 317-337.

Penner R.R, Erik S. Steinmetz (2002) Model-Based Automation of the design of user interfaces to digital control systems, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol.32, N°. 1, pp. 41-49.

Piccini M (2002) Human Factors in the Design of Supervisory Control Systems and Human-Machine Interfaces for Highly Automated Complex Systems. Cognition, Technology and Work, 4 (4), pp. 256 – 271.

Rasmussen J (1980) The human as a system component. In H.T. Smith and T.R.G. Green (Eds.), Human Interaction with Computer, London Academic Press, pp. 67-96.

Rasmussen J (1983) Skills, rules and knowledge; signals, signs and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man and Cybernetics, 13 (3), pp. 257-266.

Rasmussen J (1986) Information processing and human-machine interaction, an approach to cognitive engineering, Elsevier Science Publishing.

Rasmussen J (1993) Analysis of tasks, activities and work in the field and in the laboratories. Travail humain, vol n°2-3, pp.133-155.

Reason J (1990) Human error. Cambridge University Press, Cambridge.

Riahi M (2004) Contribution à l'élaboration d'une méthodologie de spécification, de vérification et de génération semi-automatique d'interfaces homme-machine : Application à l'outil Ergo-Conceptor+. Ph.D. Thesis, University of Valenciennes and Hainaut-Cambrésis, France, September.

Ross D.T (1977). Structured analysis (SA): a language for communicating ideas. IEEE Transactions on Software engineering, 3 (1), pp. 16-34.

Scapin D.L, Bastien C (2001) Analyse des tâches et aide ergonomique à la conception: l'approche MAD*. In Kolski C. (Ed.), Analyse et Conception de l'IHM. Interaction Homme Machine pour les SI, Volume 1, pp. 85-116. Paris: Éditions Hermes.

Scapin D.L, Pierret-Golbreich C (1990) Towards a method for task description: MAD. In Work with Display Units 89, L. Berlinguet and D. Berthelette (Eds.), Elsevier Science Publishers B.V. (North-Holland), p. 371-380.

Shepherd A (1993) An approach to information requirements specifications for process control tasks. Ergonomics, vol. 36, n° 11, pp. 1425-1437.

Shneiderman B, Plaisant C (2004) Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th edition. Addison-Wesley.

Sibertin-Blanc C (1985) High-level Petri nets with Data Structure. Proceedings 6th EWPNA, Espoo, Finland.

Smith D (2001) Reliability, Maintainability and Risk. Newnes; 6th edition.

Spolsky J (2001). User interface design for programmers. Springer-Verlag.

Stammers R.B, Carey M.S, Astley J.A (1990) Task analysis. In Evaluation of human work. A Practical Ergonomics Methodology, J.R. Wilson and E.N. Corlett (Eds.), Taylor & Francis, pp. 134-160.

Stanton N (1994) Human factors in alarm design. Taylor & Francis Ltd.

Tabary D (2001) Contribution à TOOD, une méthode à base de modèles pour la spécification et la conception des systèmes interactifs. PhD. Thesis, University of Valenciennes and Hainaut-Cambrésis, France, januar.

A version of this article has been published in: *Cognition Technology and Work*, volume 7, pp. 167-181.
The final version is accessible at : http://www.springer.com/

Tabary D, Abed M (2002) A software Environment Task Object Oriented Design (ETOOD). Journal of Systems and Software, 60, pp.129-141.

Tarby J.C, Barthet M.F (1996) The DIANE+ Method. Proceedings of the 2nd International Conference on Computer-Aided Design of User Interfaces CADUI'96, pp. 95-119, Namur, Belgium.

Vanderdonckt J (1999) Development milestones towards a tool for working with guidelines. Interacting With Computers 12, 81-118.

Vanderdonckt J, Farenc C (2000) Tools for Working With Guidelines TFWWG'2000, Springer, London.

Van Harmelen M (2001) Object modeling and user interface design, Addison-Wesley.

Villemeur A (1992) Reliability, availability, maintainability and safety assessment (vol.1 and 2). John Wiley & Sons, Chichester.

Wilson J.R, Corlett E.N (1996) Evaluation of human works: a practical ergonomics methodology (2nd edition). Taylor & Francis, London.