



**HAL**  
open science

# Modelling of an interactive system with an agent-based architecture using Petri nets, application of the method to the supervision of a transport system

Houcine Ezzedine, Abdelwaheb Trabelsi, Christophe Kolski

## ► To cite this version:

Houcine Ezzedine, Abdelwaheb Trabelsi, Christophe Kolski. Modelling of an interactive system with an agent-based architecture using Petri nets, application of the method to the supervision of a transport system. *Mathematics and Computers in Simulation*, 2006, 70 (5-6), pp.358-376. 10.1016/j.matcom.2005.11.006 . hal-03279243

**HAL Id: hal-03279243**

**<https://uphf.hal.science/hal-03279243v1>**

Submitted on 21 Dec 2023

**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.

# Modelling of an interactive system with an agent-based architecture using Petri nets, application of the method to the supervision of a transport system

Houcine EZZEDINE, Abdelwaheb TRABELSI and Christophe KOLSKI

L.A.M.I.H., UMR CNRS 8530 - Le Mont Houy  
University of Valenciennes, F-59313 Valenciennes cedex 09, France

**Abstract** – This article presents an original method using high level Petri nets for the specification and design of interactive systems. We suggest an agent oriented architecture based on the classic components of an interactive application (application, dialogue control, interface with the application). Our approach is validated via the specification and design of a human-machine interface used in the supervision of a land-based transport system (Bus/Tramway).

**Keywords** – Interactive system, Petri Nets, supervision, transport system, simulation.

## I. INTRODUCTION

Many applications currently offer Human-Machine interactions which are increasingly flexible, and even adaptive and intelligent [5,25,29]. Indeed, interactive system designers place a far greater degree of importance on the reactivity, the maintainability, the re-use and the adaptation of their system with regard to the context in which it is used. These various properties are facilitated by a modular structuring of the interactive system.

In this context, multi-agent systems provide a promising perspective because of their capacities for structuring and organising the different software components in relation to Human-Machine interaction [24,27,28]. These approaches offer the possibility of potentially interesting links between agents [23,41]: coordination links, cooperation links, and communication links. We therefore suggest using these rich links in order to put forward interactive system architectures which are adapted to the context targeted.

However, the majority of models available are mainly concerned with user-controlled applications and do not take into account the specificities found in supervision applications in which the user plays the role of controller and commander of an independent process [20,35,36,42,44]. We therefore put forward an agent-oriented specification method for the design of Human-Machine interfaces intended for the control/command of industrial systems. The specialised agents suggested by the method must be able to guarantee communication between the process and the user via input/output devices. In the supervision context, each one of these two functions has its own characteristics which are linked to the specificities of the supervision role (real time, nature and quality of information exchanged ...).

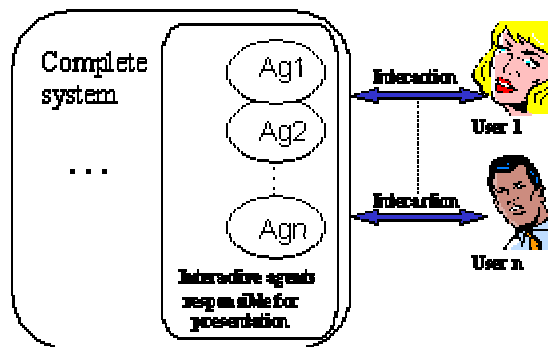
The second part of this article (which expands on [17]) describes the basic principles linked to agents for human-computer interaction. The third part concentrates on aspects of interactive system architecture. In the fourth part, we will explain how Petri nets are used for the specification of interactions. The fifth part describes the case study.

## II. PRINCIPLES OF THE AGENTS FOR HUMAN-MACHINE INTERACTION

The general principle of Multi-Agent Systems (MAS) [19,26,31,47] is the distribution of the problem, which in our case is Human-Machine interaction, into a certain number of co-operative entities which can communicate and co-ordinate their behaviour in order to reach a common objective. The particularity of each MAS is to be found in the criteria selected for the distribution of capabilities and knowledge (data) between the agents.

One of the advantages of the multi-agent approach to an interactive system concerns its ability to class the user as an agent belonging to the modelled system. The user interactions are therefore specified in the cooperating knowledge (in the sense given by [22]) of each of the system's software agents. The importance is therefore to be found in a user-adaptive presentation facet (See fig. 1) [41], which must be specified by capacities of negotiation and communication possessed by the presentation agent (or several presentation

agents). The agent will thus make it possible to deal with the intra-individual and inter-individual differences between users by providing adapted levels of interaction.



**Fig. 1** : Adaptation of the agent-based presentation facet according to the user.

The agents have two facets : one facet tends towards the collective aspect, which leads us to define the notion of agent grouping (mechanisms concerning group activities), and another individual facet with the mechanisms and the knowledge regarding the internal functioning rules at the level of the agent, concerning its own individual goals. Two types of agent can be distinguished in multi-agent systems [6,7,19] :

- lower level reactive agents (with a low degree of granularity) ; these agents only have a limited communication language and protocol, their capacities merely obey the law of stimulus/action.
- systems made up of cognitive agents are founded upon the principle of co-operation between agents which are capable of performing complex operations alone. A cognitive system generally includes a small number of agents capable of reasoning using a data base and of processing the various pieces of information linked to the application field.

It should also be noted that hybrid agents are also to be found ; these combine the two types of agent mentioned above.

### III. TOWARDS ARCHITECTURES OF AGENT ORIENTED INTERACTIVE SYSTEMS

Several architecture models have been put forward by researchers over the past twenty years, and some of them deal with the problems involved with Computer Supported Cooperative Work (CSCW) [2,4,11,12,13,18,21,38,40,46]. Two main types of architecture can be distinguished: architectures with functional components (Langage, Seeheim and ARCH) and architectures with structural components (PAC, PAC-Amodeus, MVC...). The classic models of interactive systems distinguish three essential functions (presentation, control and application). Some models (such as the Seeheim and ARCH models) consider these three functions as being three distinct functional units. Other approaches using structural components, and in particular those said to be distributed or agent approaches, suggest grouping the three functions together into one unit, the agent. The agents are then organised in a hierarchical manner according to principles of composition or communication (for example PAC [11], MVC (Model-View-Controller of Smalltalk).

Our approach could be considered as being intermediate as it borrows elements for its principles from both types of model given above.

In our architecture, we suggest using a division into three functional components which we have called respectively: interface with the application (connected to the application), dialogue controller, and presentation (this component is directly linked to the user) (Fig. 2), according to the basic principles given in [24]. These three components group together agents:

- the *application agents* which handle the field concepts and cannot be directly accessed by the user. One of their roles is to ensure the correct functioning of the application and the real time dispatch of the information necessary for the other agents to perform their task;

- the *dialogue control agents* which are also called mixed agents; these provide services for both the application and the user. They are intended to guarantee coherency in the exchanges emanating from the application towards the user, and vice versa;
- the *interactive agents* (or *interface agents*, or presentation agents), unlike the application agents, are in direct contact with the user (they can be seen by the user). These agents co-ordinate between themselves in order to intercept the user commands and to form a presentation which allows the user to gain an overall understanding of the current state of the application. In this way, a window may be considered as being an interactive agent in its own right; its specification describes its presentation and the services it is to perform.

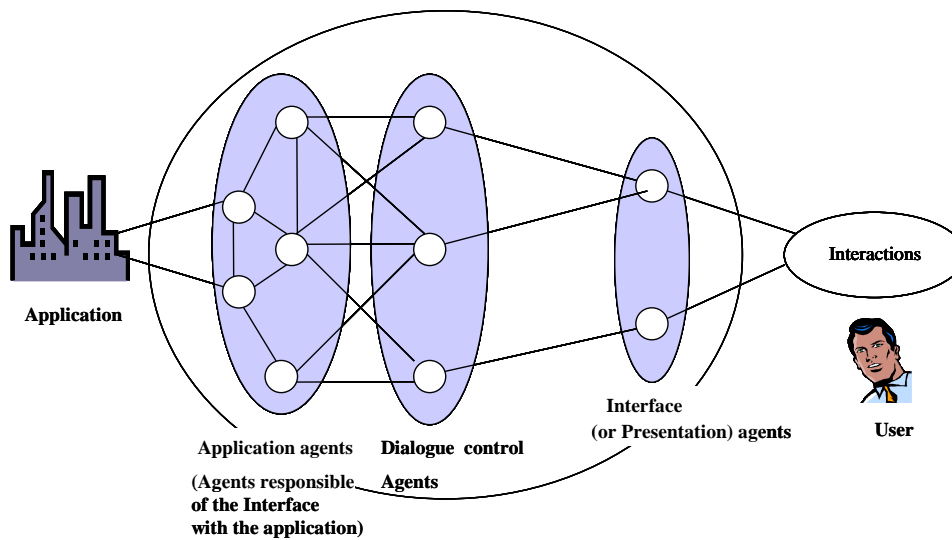


Fig.2 : Agent-oriented model of an HMI [24]

#### IV. USE OF PETRI NETS FOR SPECIFICATION AND DESIGN OF AGENT-BASED INTERACTIVE SYSTEMS

Each agent therefore plays a role within its group. This role can be expressed in the form of the services it offers in the interactive system. A service is defined as being a quadruplet  $s = \{E, C_a, R, P\}$  with (see Fig. 3) :

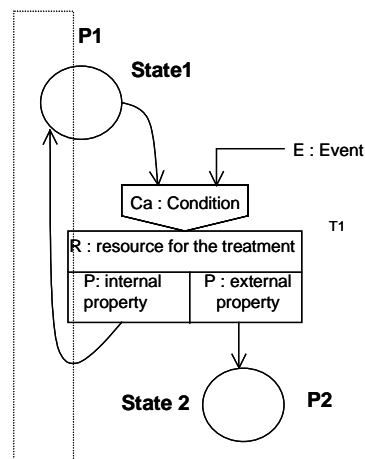


Fig. 3 : Definition of an agent's service

- E : the event which triggers the service,
- $C_a$  : the conditions to be met in order to perform this service,
- R : the resources necessary for the service to be performed,
- P : the property of this service, which can be either an operation concerning the agent alone (with or without a change of state for the interactive agents), or a call for the service of another agent (with a

change of state for the reactive agents). The succession of various calls for services gives rise to the succession of windows in the human-machine interface.

The behaviour of the agents is modelled using high level Petri nets; this makes it possible to validate certain properties of the application before it is implemented (vivacity, re-initialisation, ...). The modelling of the multi-agent system which corresponds to the interface with the application facet (cf. Fig. 2) using high level Petri nets, will make it possible to provide the specification of services between agents on the one hand, and on the other hand the dynamic simulation of the application with a view to assessing the efficiency of the overall system.

We illustrate the notion of service using two significant examples in section V.4.

## V. APPLICATION TO A SUPERVISION INTERFACE FOR A SIMULATED BUS AND TRAMWAY NETWORK

The validation of our approach occurred within the framework of a project involving an industrial partner, the SEMURVAL company which will run the future urban transport network (tramway and bus) in the town of Valenciennes, as well as several research laboratories (LAGIS, LAMIH and INRETS) [8,15,16,33]. Urban transport networks, and in particular the systems intended to provide information for the passengers will act as an example of the application of our method.

This project is sponsored by the Nord-Pas de Calais regional authorities and by the FEDER (Fonds Européen de Développement Régional – European fund for regional development). Our research work consists in the specification, design and evaluation of a human-computer supervision interface intended for human regulators working on passenger information on the transport system in Valenciennes (referred to later as the Information Assistance System or IAS)

As the tram lines are still being built, the project team has to simulate the existence of a Tramway/Bus network. The transport network will include information screens (or display units) intended for the passengers. These screens will be found both in the stations and in the vehicles; they will show information on the schedules and on the connections. This information will be calculated automatically by an exploitation assistance system (EAS) which knows the position and state of each vehicle in real time

We need to create an assistance system for the regulation of land traffic on the public transport network in Valenciennes. This regulation system is made up of three sub-systems: an Exploitation Assistance System (EAS), a Decision Assistance System (DAS) and an Information Assistance System (IAS) (see Fig. 4).

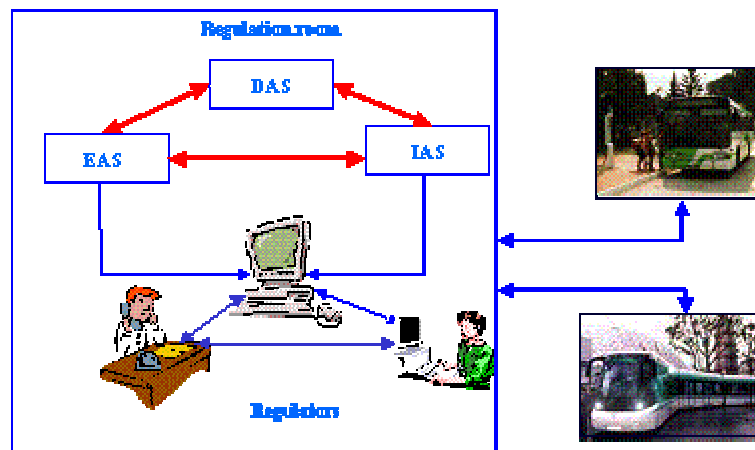


Fig. 4 : Description of the traffic regulation assistance system

Each of the three sub-systems has a particular role to play :

- The EAS : centralises the information regarding exploitation (schedules, delays, vehicles ahead of time, messages, alerts, etc.), and makes it possible to manage these elements [32] ;
- The DAS : creates, evaluates and suggests regulation strategies to the human regulator using the information provided by the EAS. In this way, the regulator has less work to do, which should help to

improve the quality of regulation and thus the quality of the overall service. The DAS is not intended to replace the human regulator, but it must provide assistance in decision-making [9,30,34] ;

- The IAS is the interactive system which is the main subject of our research work; it has to present a set of information to the regulators in the control room and make it possible to send relevant information to the passengers [10,15,17].

As in the vast majority of supervision problems, it is a question of constantly monitoring the state of a process and sending orders for the regulation of that process. The process has its own particular behaviour, linked to variables which the human regulator can often neither completely understand nor modify. In our case, the traffic conditions have an influence on the times at which buses go through stations, but the regulator cannot influence the traffic conditions. On the other hand, the regulator can modify the state of the transport system and regulate the passenger information accordingly. Our approach must provide the information needed by the regulator in order to perform this regulation. The regulator must be able to :

- follow the evolution of the passenger information generated and transmitted automatically by the IAS over the network (that is in the stations and in the vehicles),
- regulate this information if necessary according to the management aims (reaction in the event of considerable delays for example) or according to one-off situations (stops being moved temporarily, manoeuvres in the event of vehicle breakdown, etc.).

Information in public transport networks has become not only a need but also a requirement. The users of a multi-mode public transport system (trams, buses, underground trains, over-ground trains) want to be better and constantly informed before and during their journeys and to receive reliable information. This information must therefore be regulated (and not simply automatic) and a dynamic interface must be designed to enable the human regulator to perform his/her role of information regulation (control/command of the information displayed). Each element of information transmitted to the passengers must be truly pertinent and relevant.

The system targeted by our research project will be referred to as the IAS (Information Assistance System) in the rest of this article.

## **V.1 MODELLING OF THE INTERACTIONS BETWEEN THE IAS AND THE HUMAN REGULATOR WITH INTERPRETED PETRI NETS**

In any interactive application, the use of a formalism describing the system's dynamics is of capital importance. Indeed, modelling is all the more necessary since it provides formal analysis techniques which make it possible to prove properties of the design before its implementation [1,39]. We therefore decided to use Petri nets, and more precisely interpreted Petri nets, in order to model the interactions between the system and the human regulator (Fig. 5). The Petri net is only concerned with the interactive part and therefore only with the relevant information which has to go via the regulator. All other information is dealt with using an automatic mode. Four main functions performed by the IAS can be distinguished:

- automatic management of information (with no intervention by the regulator). This encompasses times of delayed stops, destinations, following stops, and also information on connections (informing the passengers on the connections possible),
- the visualisation function : this makes it possible to inform the regulator, at any time, of the state of the network from the point of view of information intended for the passengers. One of the roles of this function is to control the state of the display devices (breakdown of display panels, divergence between the information actually displayed and the information sent to the regulation post, ...), information being processed (consultation of the information displayed on the mobile units and the network stations at any time) as well as the position of the mobile units and their state,
- the decision support function, which consists in suggesting messages to be displayed to the regulator in the event of considerable disruptions. These messages can be shortened, modified or sent directly by the regulator as he/she decides.
- the message function which allows the regulator to send a message to the network unit of his/her choice at any time.

The regulator can create, edit and send messages to the passengers at the stations and/or on board the vehicles. The regulator therefore interacts with the IAS in order to perform his/her regulation task correctly. The top part of figure 5 (part 1) shows the interactions possible between the EAS, the DAS and the IAS, along with the processing of the events received by the IAS. The lower part (part 2) shows the various interactions possible between the regulator and the IAS.

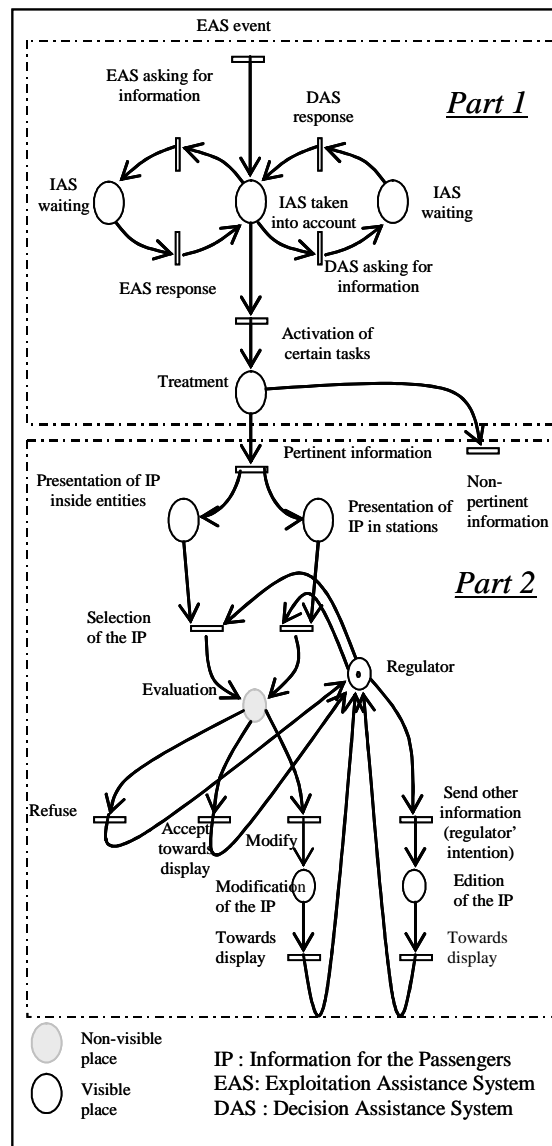


Fig. 5 : Modelling of interaction

This first global level of modelling does not allow a distinction between the IAS's two functioning modes, that is normal running mode and disrupted running mode. This is why we will now give two more detailed examples of models, the first in normal running mode, and the second in disrupted running mode (both using interpreted Petri nets).

### V.1.1 NORMAL RUNNING MODE

The normal running mode is represented on the right hand side of figure 6. The place P1 represents the information assistance system on standby. In the normal running mode (right side of figure 6), the IAS can react to two possible actions (crossing of T1 or T2):

- The crossing of T1 represents a request for information by the regulator. This request is dealt with directly by the IAS with no particular involvement of the DAS or the EAS. An example of a request for information would be the visualisation of a line.

- The crossing of T2 represents a request for information by the regulator via the EAS. In this case, the place P2 models the period during which the IAS waits for the response from the EAS. The crossing of T3, which symbolises the response of the EAS, brings about the display of the information modelled by the place P3.

In normal running mode, the regulator can send messages to the vehicles and to the stations upon his/her own initiative (crossing T12).

### V.1.1 DISRUPTED RUNNING MODE

The disrupted running mode is shown in figure 6. Following the detection of a disruption (crossing of transition T18), the EAS prepares the information necessary for the IAS (P14) and the DAS (P15). The information received by the IAS is processed (P4) in order to sort out which information is relevant. The information which comes from a non serious disruption will be used to update the IAS (P5). In the event of a serious disruption, the IAS can react in two ways:

- Directly present the necessary information to the regulator
  - Wait for the decision from the DAS to display the information for the regulator.
- Place P11 is a virtual place (a non visible place). It represents the regulator's decision as regards the messages suggested :
- The crossing of T14 represents the complete refusal of the message suggested.
  - The crossing of T15 happens once the suggested message has been accepted.
  - The crossing of T16 occurs when the regulator decides to modify the message which has been suggested.

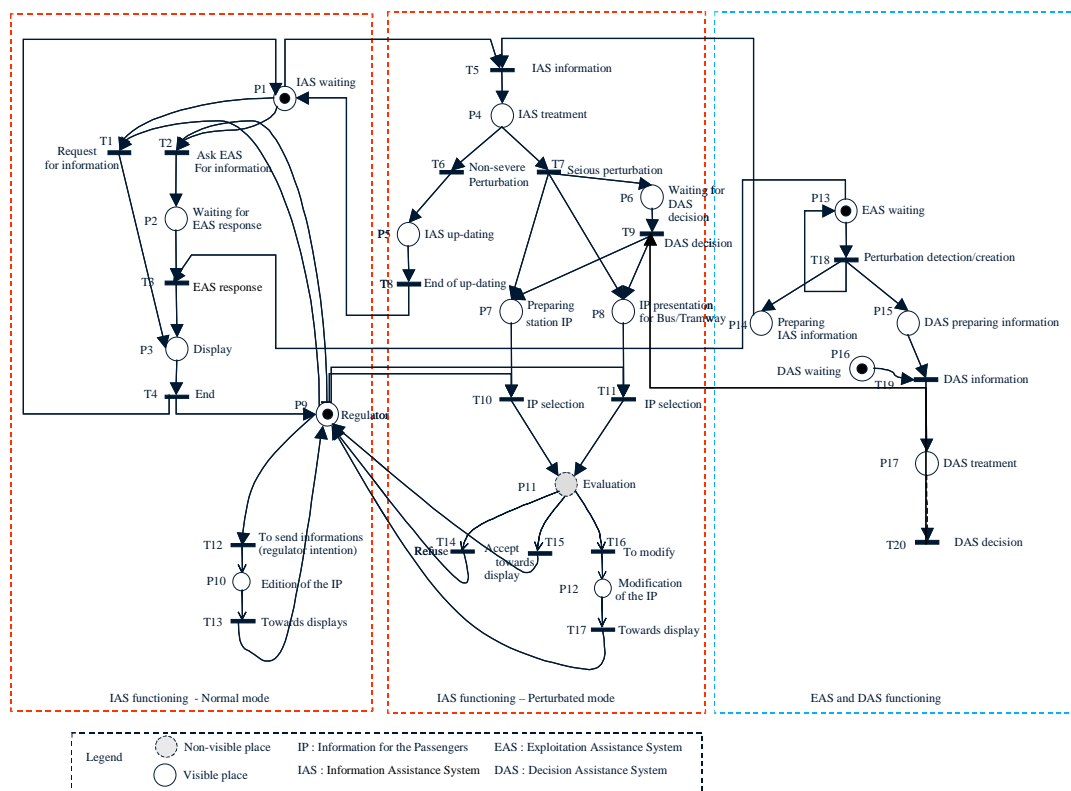


Fig. 6 : Model of the running of the IAS with interpreted Petri nets

### V.1.3 GLOBAL MODELLING OF THE VARIOUS INTERACTIONS INVOLVING THE REGULATOR

In this section we present an example illustrating the interaction between the human regulator, the IAS, the EAS, the DAS and the vehicle agents (Fig.7).



Modelling of the behaviour between IAS/Regulator, EAS, DAS and vehicles with Petri net

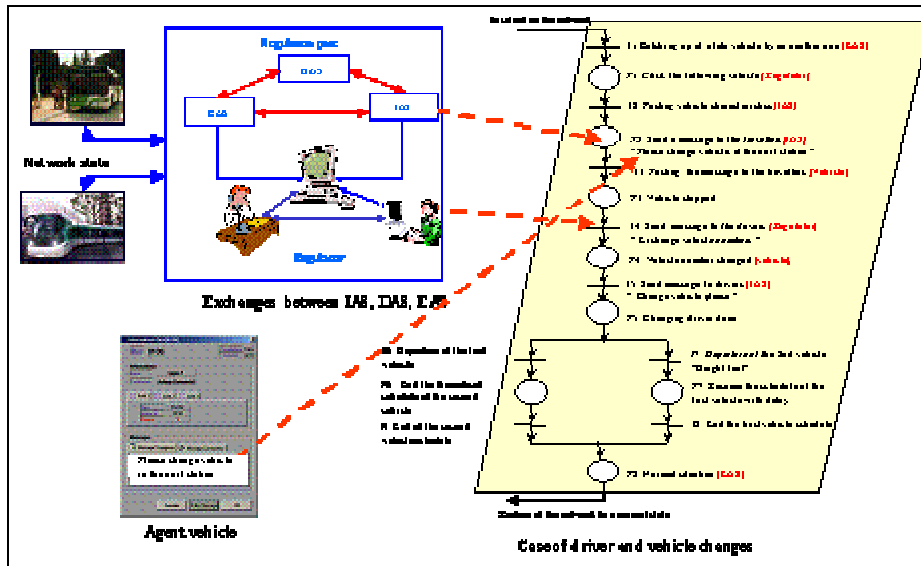


Fig. 7 : Modelling of the interactive behaviour between IAS/Regulator, EAS, DAS and vehicle agents with Petri nets

As shown in Figure 7, events coming from the EAS "action" or "state of the network ", which are connected to the time information and/or to the incidental information, are taken into account by the IAS with the crossing of respective transitions t1.

After obtaining further information from the IAS/Regulator system (Place P1), several tasks are activated then sequentially treated by the regulator, the IAS, the vehicle or the drivers (crossing of transitions t2, in t7). This information indicates both the information for travellers inside vehicles and the information intended for the drivers. This information is selected by the regulator (Place P2, transitions t4 and t5).

This formal representation is very useful for us in the specification of the HMI. It allows us to list several general classes of components :

- The *state* component which is always defined by the marking  $M_i$  on the Petri net (for example, the marking for the places P1 to P8) illustrates the posting of traveller information and the message given to the drivers in the vehicles as well as the vehicle state.

- The *behaviour* component describes the evolution of the marking. In other words, it describes the possible evolutions from a given state,

- The *service* component gives the conditions of passage between two states. It is represented by the set of transitions or places, where we can distinguish :

- The regulator actions which are to be triggered (it is the case of the place P1 and the transition t4);
- The service transitions activated by the EAS or the IAS (t1, P2, t5, P8).

## V.2 EVENTS LINKED TO THE IAS

As shown in Figure 8, the IAS can be affected by two types of event which can considerably change its functioning: "action" events and "network" events.

- The "action" events correspond to orders given by the regulators from the control/command post. They can correspond to:
  - a modification to the topology of the network (supervision of a stopping point,...),
  - messages sent by the regulator,

- a modification to the running plan.
- The "network" events which come directly from the field:
  - Disruptions (maintenance work, vehicle breakdown, demonstrations...),
  - Messages sent by the driver.

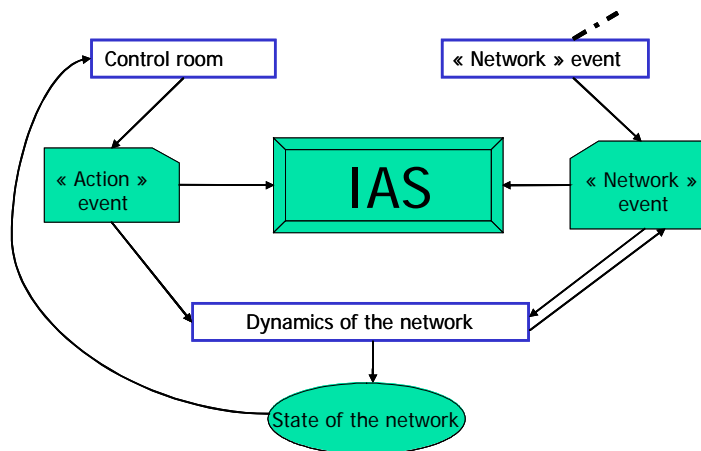


Fig. 8 : Events linked to the IAS

### V.3 AGENT ORIENTED SPECIFICATION OF THE HUMAN-MACHINE INTERFACE

We suggest the use of an agent oriented architecture which divides the agents into three classes. As a reminder, these classes are called respectively: application agents, dialogue control agents and interface agents (or presentation agents).

#### V.3.1 SPECIFICATION OF APPLICATION AGENTS

The application agents are intended to manage the passenger information in the vehicles and stations and to calculate the information to be displayed (delays, timetable and route modifications, etc.). According to the traffic context, each agent possesses rules enabling it to act correctly in its environment.

#### V.3.2 SPECIFICATION OF DIALOGUE CONTROL AGENTS

The dialogue control agents have a dual role. They make it possible to:

- Diffuse the modifications observed at the application level on the interactive agents,
- Diffuse the regulator's command actions towards the application agents.

These agents act as an intermediary for interaction between the interface agents and the application agents.

#### V.3.3 SPECIFICATION OF THE INTERFACE AGENTS

We have identified six types of interface agent responsible for direct interaction with the human regulator. These agents are represented in the form of interactive windows. The regulator can interact with these agents via the various functions possible in the windows, for example: the buttons, the edition zones, the pictures, etc. These agents are:

- The *State of traffic agent*,
- The *State of the line agent*,
- The *Station agent*,
- The *Vehicle agent*,
- The *Message agent*,
- The *Overall view agent*.

The modelling of the multi-agent system interface with the application using high level Petri nets makes it possible firstly to specify the services between agents and secondly to provide a dynamic simulation of the application with a view to assessing the performance of the whole system.

#### V.4 EXAMPLES OF THE USE OF PETRI NETS FOR THE DEFINITION OF AN AGENT'S SERVICE

Two examples are given.

##### Example 1 :

Table 1 and Figure 9 illustrate an internal operation within an interface agent. Following the event "vehicle delay" indicated by the network support system, the *State of the traffic* interface agent is notified by an application agent, via a dialogue control agent, of the delay of the vehicle in question. The interface agent analyses this event and checks if the condition "Delay greater than 5 minutes" is fulfilled. The validation of this condition brings about the display of a dialogue box, used as a resource by the interface agent, which will inform the regulator of a considerable disruption in the traffic. The interface agent will use its internal properties to determine the nature and category of the message to be transmitted.

| Event             | Delay of vehicle indicated by the Network Support System |
|-------------------|--|
| Condition         | Delay greater than 5 Minutes                             |
| Resource          | Dialogue box   |
| Internal property | Message to be transmitted (nature, category, content...) |
| External property | None   |

Table 1 : Agent's internal operation (literal description)

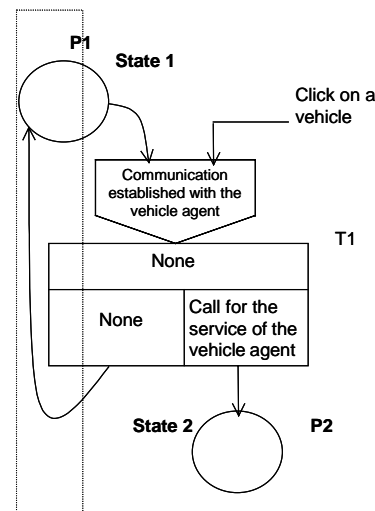


Fig. 9 : Agent's internal operation (Petri Net modelling)

##### Example 2 :

As we have stated in the previous section, the interface agents are interactive. Therefore, if the *State of the traffic* interface agent perceives the event "Click on a vehicle" performed by the regulator (see Fig. 10, Table 2), it calls upon the service of the *Vehicle* interface agent. In this case, the interface agent will need to use neither resource nor internal properties.

|                   |  |
|-------------------|--|
| <b>Event</b>      | <b>Click on a vehicle</b>                        |
| Condition         | Communication established with the vehicle agent |
| Resource          | <b>NONE</b>                                      |
| Internal property | None   |
| External property | Call for the service of the vehicle agent        |

Tableau 2 : Call for another agent's service (literal description)

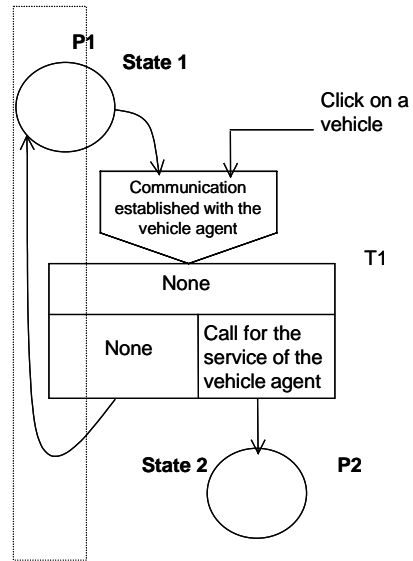


Fig. 10 : Call for another agent's service (Petri Net modelling)

## V.5 GRAPHIC INTERFACE

It should be remembered that the aim of the IAS is to enable the regulator to visualise, edit, create and transmit information intended for the passengers in the stations and/or vehicles.

In order to perform his/her task of regulation correctly, the regulator interacts with the six interface agents: the *State of the traffic* agent, the *State of the line* agent, the *Station* agent, the *Vehicle* agent, the *Message* agent and the *Overall view* agent. All of these agents are developed using the C++ language. Several examples of such agents are given below.

### V.5.1 THE "STATE OF THE TRAFIC" INTERFACE AGENT

The "*State of the traffic*" interface agent gives a synthetic representation of all the delays concerning mobile units travelling on the network. Thus, with the help of the network support system, it ensures the real time surveillance of vehicle delays on the network supervised (Fig. 11).

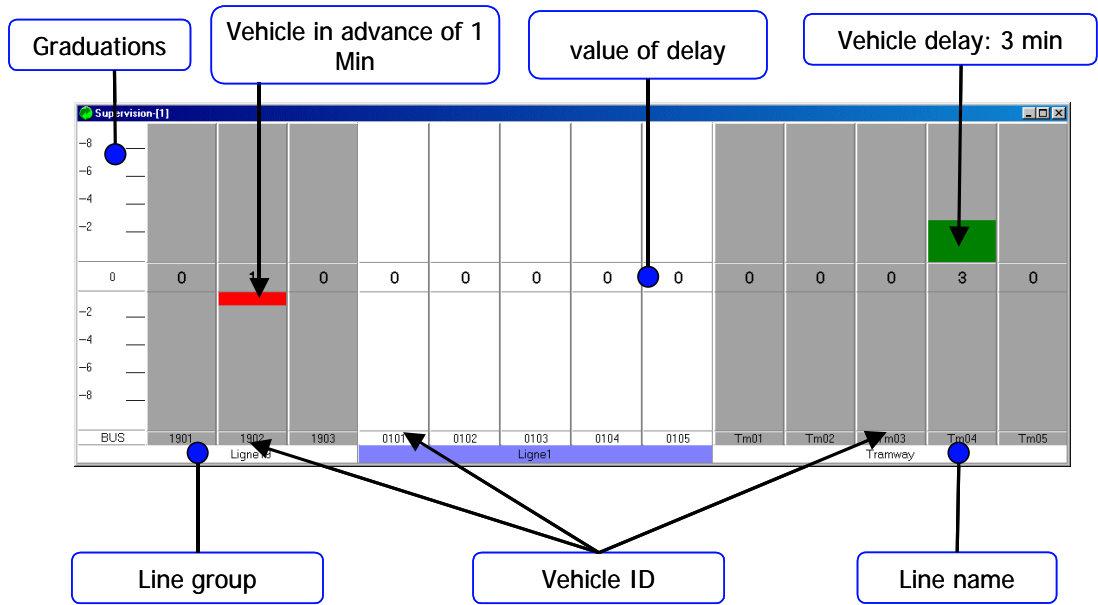


Fig. 11 : View of the “State of the traffic” interface agent

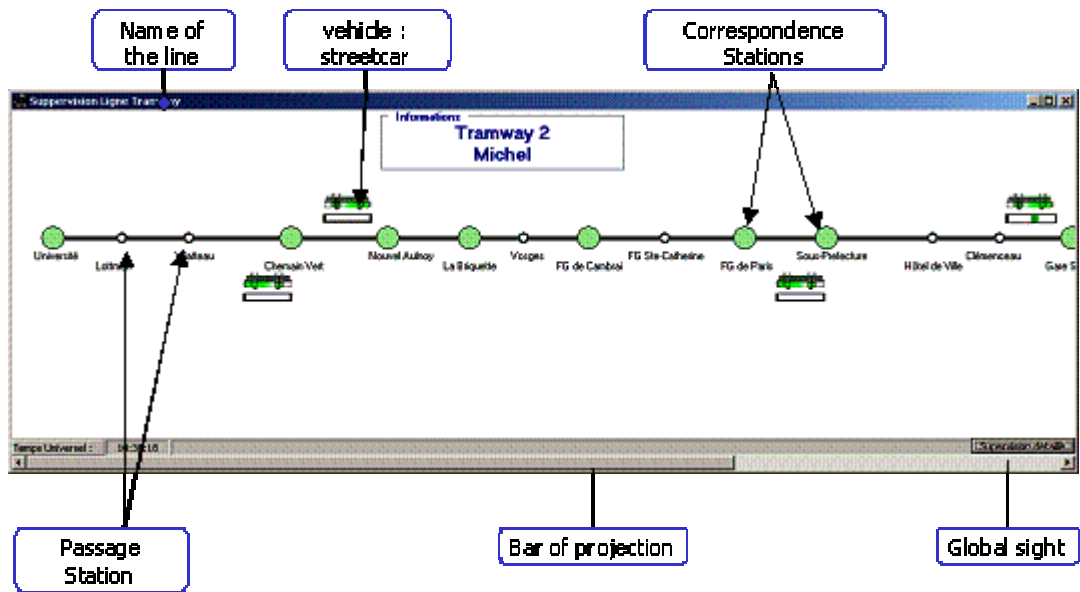


Fig. 12 : View of the “*State of the line*” interface agent

### V.5.2 THE “STATE OF THE LINE” INTERFACE AGENT

The view of the *State of the line* interface agent is made up of graphic elements such as stations, route sections, vehicles, etc. (see Fig. 12). A click on a vehicle directly displays the view (window) of the *vehicle* interface agent which will deal with any further interaction with the regulator.

### V.5.3 STATION AND VEHICLE INTERFACE AGENTS

The *Station* and *Vehicle* interface agents: the view of these two interface agents is accessible by acting on their associated representations in the *State of the line* interface agent view (vehicle and station). The view of the *Station* interface is made up of two independent panels (Message and information from the running plan). The view of the *Vehicle* interface agent is more or less made up of the same elements and functions as the view of the *Station* interface agent. (Fig. 13.a). It shows the regulator the information contained in the running plans in the form of a set of thumbnails depending on a direction which can be selected on a drop-down scroll list (Fig. 13.b).

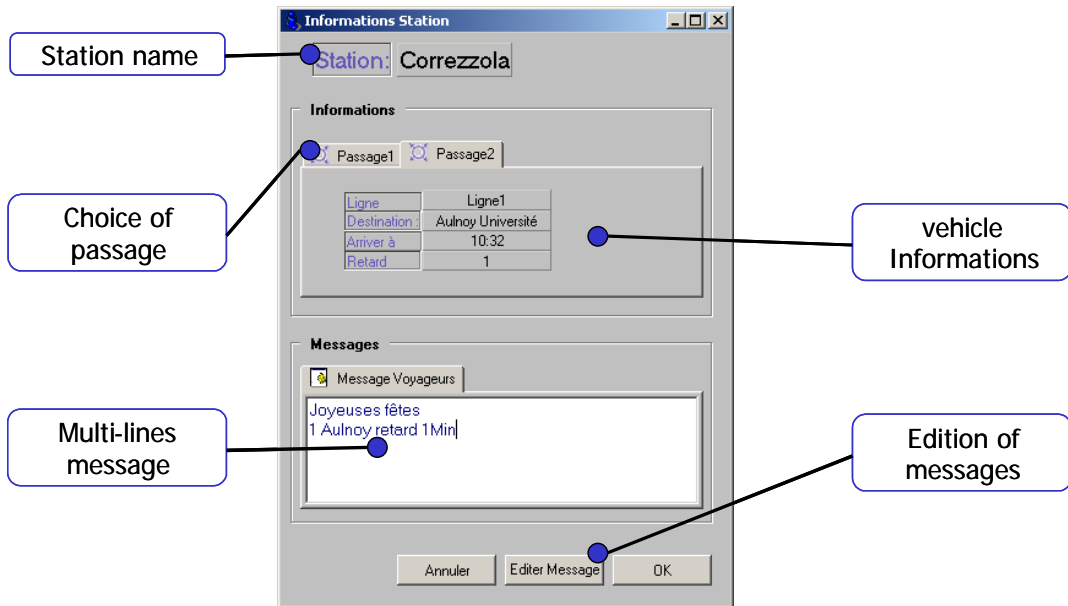


Fig. 13.a : View of the *Station* interface agent

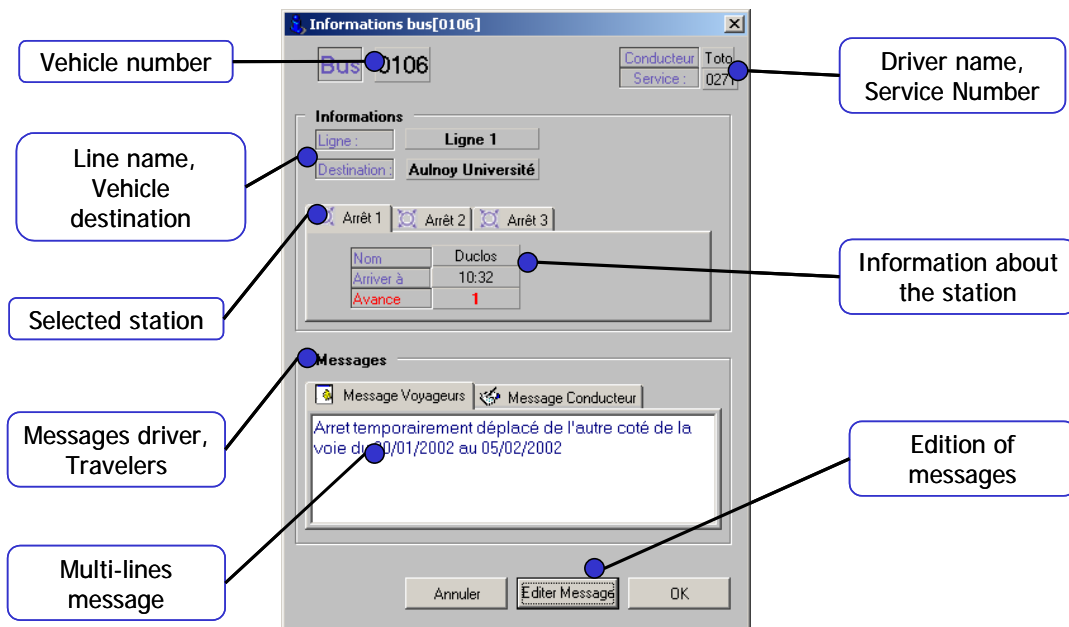


Fig. 13.b : View of the *Vehicle* interface agent

#### V.5.4 THE MESSAGE INTERFACE AGENT

The view of the *Message* interface agent enables the regulator to obtain a synthetic view of all the messages being sent to vehicles and stations. It is made up of four types of panel (Fig. 14): one panel which contains a list of all the messages which can be sent, one panel making it possible to choose the destination line, with an option to make a multiple selection, the two other panels enable the regulator to select the station and vehicles which are to receive the message selected.

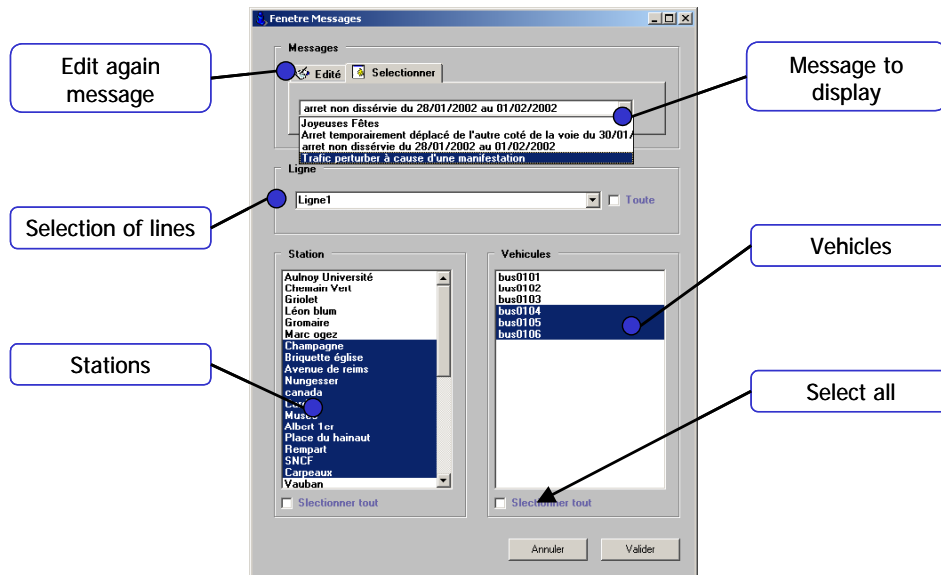


Fig. 14 : View of the *Message* interface agent

#### V.5.5 INTERACTION BETWEEN THE VEHICLE AGENT AND THE STATE OF THE TRAFFIC AGENT

Each *Vehicle* agent can be affected by the human regulator's actions. Indeed, whenever the mouse goes over the representation of a vehicle (a picture : bus or tram), the name of the driver and the number of the vehicle are displayed in a help zone on the supervision window (*State of the traffic* agent) of the line being supervised (Fig. 15). With this function, the regulator can rapidly recognise the identification of the vehicle and the name of the driver, as well as seeing whether the vehicle is late or ahead of schedule.



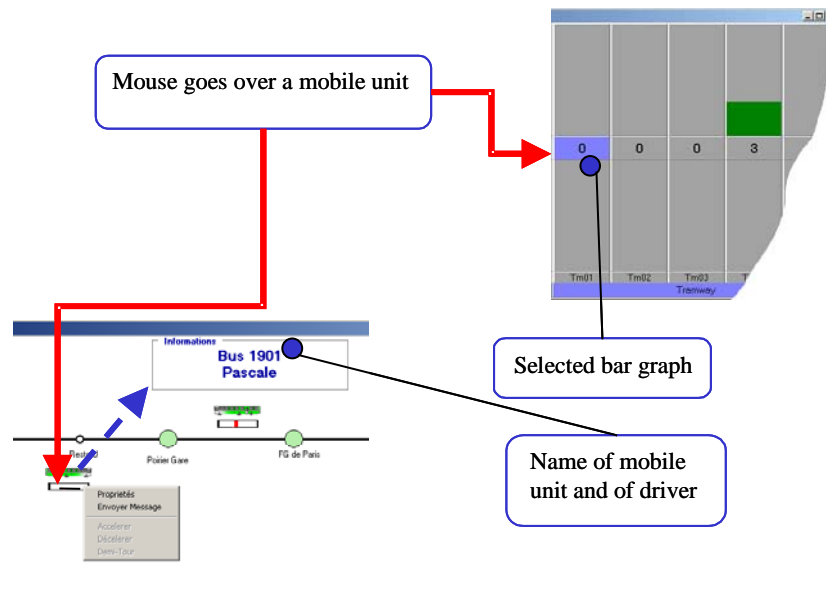


Fig. 15 : Interaction between a *vehicle* agent and a *state of the traffic* agent.

## VI. EVALUATION AND CRITICAL ANALYSIS OF THE APPROACH

In order to determine the performances of our approach of modeling, an evaluation of the developed interactive systems is necessary.

The evaluation of interactive systems consists in ensuring that the users are able to carry out their task by using the interface. The interface must therefore meet their needs. The methods for evaluating user interfaces are as varied as they are numerous. In [3,37,43,45], the emphasis is placed on two global evaluation criteria: utility and usability: (1) the utility criteria determines if the interface meets the user needs; in other words, if application allows the user to reach the work objectives. Utility considers the global system performance, functional capacity and quality of proposed technical assistance. (2) The usability criteria ensures an easy and intuitive manipulation of the interactive application ; the ease of learning as well as the quality of the aids are main arguments which define usability.

Indeed, evaluation consists in verifying and confirming the interactive system, whatever the domain of application (control/command, information systems, etc.). If the HMI prototype meets advanced criteria, it is then accepted and validated. Otherwise it has to be reorganized for a new evaluation.

In order to realize this evaluation we developed an agent oriented electronic informer. An "electronic informer" is a software tool which ensures the automatic collection, in a real situation, of users' actions and their repercussions on the system. The collection of information is done in a discreet and transparent way for the user, who must not at any time feel hampered by the presence of the informer. This is an advantage of such a tool. Objective data collected through human-machine interactions can be analyzed and shown in a synthetic shape to the evaluator. This facilitates the analysis of the interfaces use problems (Fig. 16).

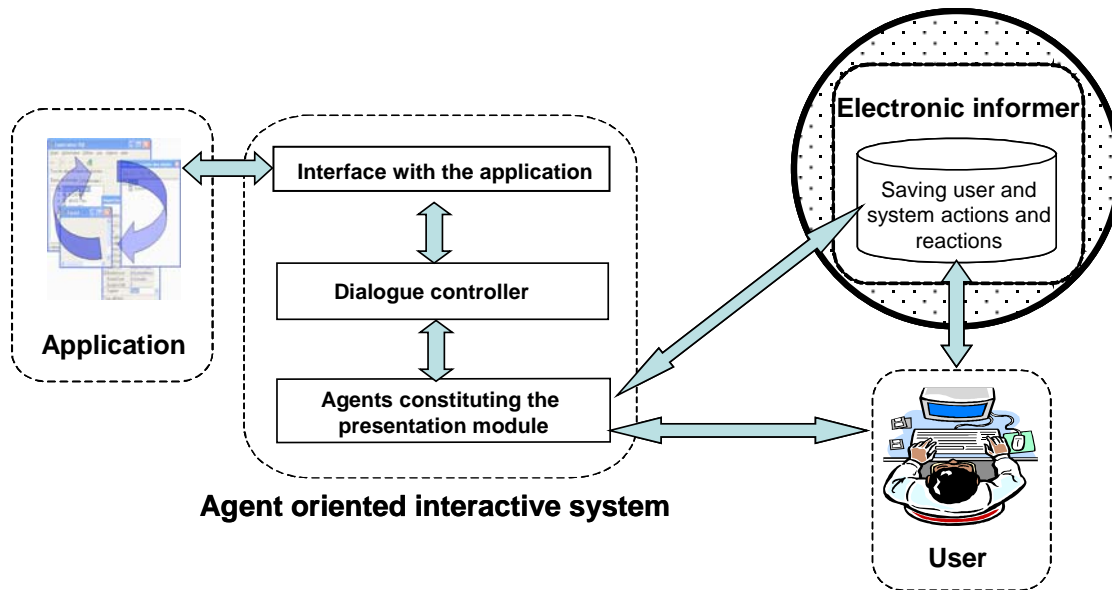


Fig.16. Using the electronic informer for the evaluation of agent oriented interactive system [16]

Our proposition of an electronic informer, studied and developed as a tool for the evaluation of agent oriented interactive systems, is strictly related to the architecture of the evaluated system. Indeed, the electronic informer will be based essentially on the acquisition of information and data coming from the evaluated global system. This will make it possible to build tasks really carried out by the user (*a posteriori* mode) and to compare them with the model of tasks to be carried out (*a priori* mode), according to principles of confrontation described in [1,14]. We are interested in our evaluation particularly with the interface agents (constituting the presentation module). Figure 16 shows the use of the electronic informer as an evaluation tool for agent oriented interactive systems.

The evaluation of the interactive systems proceeds in two phases: the first consists in collecting the maximum of data related to the HMI and to the user; the second phase is dedicated to the analysis of the recovered data. The experiment shows that a simple observation of the human-machine interaction, even provided with points of measurement, does not make it easily possible to return account of the inconsistencies or maladjustment of the interaction and even less of the correct actions to carry out. Indeed the manual deployment of the recorded data proves to be long and tiresome [4]. In order to facilitate the work of the evaluator, we equipped our electronic informer with an analysis data module. The evaluator can thus consult the various data according to five categories : (1) mouse movement, (2) data seized by the keyboard, (3) data seized by the parole, (4) visualization time of each sight, (5) use of the help. Category 4 (duration of visualization of each sight) allows providing data relating to the frequency and the duration of displaying the interface agents (presentation component) following to:

- an interns event system (conflict of allocation of resources) or external relating to the process (incidental),
- a request from the user in order to consult information on the state of the application to achieve an action or a well defined task.

Figure 17 presents an example of result provided by the electronic informer concerning the rate of consultation of the interface agents (presentation component).

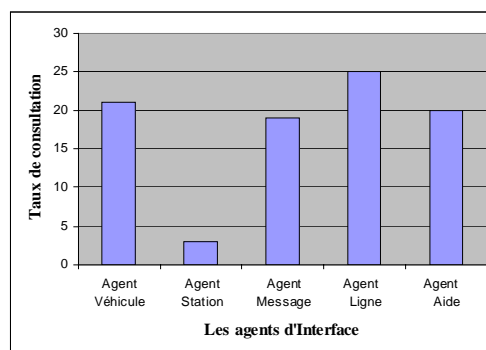


Fig. 17. Rate of consultation of the interface agents

These rates of consultation inform the evaluator about the degree of utility of each sight. In this example, the presentation agent "Station" is rarely requested, the evaluator can examine the utility and/or the utilisability of such a sight. In addition, the excessive call of the interface agent "Help" let the evaluator believe that the user is, either beginner compared to the tasks to be achieved, or he or she is confronted with new situations. In this case, the evaluator can consult the nature of use of the help in order to better understand the concerns of the user.

The main criticism which one can bring to our approach of modeling architecture of agent oriented interactive systems lies in the complexity of the interaction model between the agents which are presented by the Petri nets. Indeed, the more the architecture of the interactive system is complex in term of number of agents and their interactions, the more the obtained model risk to be illegible for the designer and the evaluator. The solution consists to break up the complete interactive system into several subsystems made up of agents then to model each one of them. Some of our researches are oriented in this way. Our modelling approach may be applied in other fields and not only in the human-machine interaction domain; for example for the modelling of co-operative robots in a flexible manufacturing system where each robot can behave like an agent of a multiagents system or subsystem.

## VII. CONCLUSIONS AND PERSPECTIVES

In this article, we have put forward an agent-oriented specification approach for the design of human-machine interfaces. This approach was applied during a project involving a bus and tram network, and was intended to provide significant support in the field of passenger information.

The use of an agent-oriented interactive system specification approach for the Information Assistance System (IAS) led to a modular breakdown of the interface. These modules were the subject of a specification using high level Petri nets.

Following the specification and design phases, an assessment phase will be performed as regards both the technical and the ergonomic aspects of the interface. This is currently being prepared.

## ACKNOWLEDGEMENTS

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

## REFERENCES

- [1] M. Abed, H. Ezzedine, C. Kolski, Modélisation des tâches dans la conception et l'évaluation des systèmes interactifs : la méthode SADT/Petri, in : C. Kolski (ed.), Analyse et Conception de l'IHM, Interaction Homme-Machine pour les SI, Volume 1, (Editions Hermes, Paris, 2001) 145-174.
- [2] L. Bass, R. Little, R. Pellegrino, S. Reed, et al., The Arch model : Seeheim revisited, Proceedings of User Interface Developers' Workshop, Seeheim, 1991.
- [3] C. Bastien, Panorama des outils de recueil d'information, Proceedings ERGO-IA'2002 Conference, october, Biarritz, France.
- [4] G. Calvary, J. Coutaz, L. Nigay, From Single User Architectural Design to PAC\*: a Generic Software Architecture Model for CSCW, in: Proceedings of CHI'97 (Atlanta, March 1997) (ACM Press, New-York, 1997) 242-249.
- [5] G. Calvary, J. Coutaz, D. Thévenin, Supporting context changes for plastic user interfaces: a process and a mechanism, in: Blanford A., Vanderdonck J., Gray P., eds., People and Computer XV – Interaction without Frontiers (Springer, 2001) 349-363.
- [6] B. Chaib-Draa, Interaction between Agents in Routine, Familiar and Unfamiliar Situations, International Journal of Intelligent et Cooperative Information Systems 5 (1) (1996) 1-25.
- [7] B. Chaib-Draa, B. Moulin, R. Mandiau, P. Millot, Trends in Distributed Artificial Intelligence, Artificial Intelligence Review 6 (1992) 35-66.
- [8] F. Chihaib, S. Hammadi, P. Borne, The Contribution of Linear Programming by Constraints Propagation In the Regulation of Traffic of Urban Transport Network, Proceedings ITS 2000 7<sup>th</sup> World Congress On Intelligent Transport Systems, Torino November 6-9, 2000.

- [9] F. Chihaib, S. Hammadi, P. Borne, Régulation du trafic inter-stations, Final report, GRRT Cooperative Project, Lille, France, April 2001.
- [10] F. Cartegnie, H. Ezzedine, A. Péninou, C. Kolski. Agent oriented specification of interactive systems: Basic Principales and industrial case study, in: C. kolski, J. Vanderdonckt, eds., *Computer-Aided Design of User Interfaces III*, (Kluwer Academics Publishers, Dordrecht, 2002) 381-388.
- [11] J. Coutaz, PAC, an implementation model for dialog design, in: H.J. Bullinger, B. Shackel, eds., *Proceedings Interact'87 Conference* (North Holland, 1987) 431-436.
- [12] J. Coutaz, L. Nigay, Architecture logicielle des systèmes interactifs, in: Kolski C., ed., *Analyse et Conception de l'IHM. Interaction Homme Machine pour les SI, Volume 1* (Éditions Hermès, Paris, 2001) 207-246.
- [13] P. Dewan, Architectures for Collaborative Applications, in: M. Beaudouin-Lafon, ed., *Computer Supported Co-operative Work* (Wiley & Sons Pub, 1999) 169-193.
- [14] H. Ezzedine, M. Abed, Une Méthode d'évaluation d'Interface Homme-Machine de supervision d'un procédé industriel. *Journal Européen des Systèmes Automatisés*, (JESA), vol. 31, n°7, pp. 1087-1110, 1997.
- [15] H. Ezzedine, H. Maoudji, F. Cartegnie, A. Péninou, Towards agent oriented specification of Human-Machine Interface : Application to the transport systems, *Proceedings 8<sup>th</sup> IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, pp421-426, Kassel, Germany, 2001.
- [16] H. Ezzedine, Méthode et modèles de spécification d'évaluation des interfaces homme-machine dans les systèmes industriels complexes, *Habilitation à diriger des recherches*, University of Valenciennes, France, 2002.
- [17] H. Ezzedine, A. Trabelsi, C. Kolski, Modelling of Agent oriented Interaction using Petri Nets, application to HMI design for transport system supervision, in: P. Borne, E. Craye, N. Dangourmeau, eds., *Proceedings CESA2003 IMACS Multiconference Computational Engineering in Systems Applications* (Lille, France, July 9-11, 2003) (Ecole Centrale Lille, Villeneuve d'Ascq, 2003) 1-8.
- [18] J.D. Foley, A. Van Dam, *Fundamentals of interactive computer graphics* (Addison-Wesley, 1982).
- [19] J. Ferber, *J. Multi-agent Systems : An introduction to distributed artificial intelligence*. Harlow, UK, (Addison-Wesley Longman, 1999).
- [20] W.E. Gilmore, D.I. Gertman, H.S. Blackman, *User-computer interface in process control*, a *Human Factors Engineering Handbook* (Academic Press, 1989).
- [21] A. Goldberg, *Smalltalk 80 : The Interactive Programming Environment* (Addison-Wesley Publ., 1984).
- [22] E. Grislin-Le Strugeon, P. Millot, Specifying artificial cooperative agents through a synthesis of several models of cooperation, in: J.M. Hoc, P. Millot, E. Hollnagel, & P.C. Cacciabue, eds., *Proceedings of CSAPC'99* (Presses Universitaires de Valenciennes, Valenciennes 1999) 73-78.
- [23] E. Grislin-Le Strugeon, A. Péninou. Interaction with agents systems: problematics and classification, in: M.J. Smith, G. Salvendy, D. Harris, R. Koubek, eds., *Usability evaluation and Interface design: Cognitive Engineering, Intelligent Agents and Virtual Reality*, volume 1, (Lawrence Erlbaum Associate Publishers, London, 2001) 479-483.
- [24] E. Grislin-Le Strugeon, E. Adam, C. Kolski, Agents intelligents en interaction homme-machine dans les systèmes d'information, in Kolski C., ed., *Environnements évolués et évaluation de l'IHM, Interaction Homme Machine pour les SI 2* (Éditions Hermès, Paris, 2001) 207-248.
- [25] K. Höök, Steps to take before intelligent user interfaces become real, *Interacting with computers* 12 (2000) 409-426.
- [26] N.H. Huhns, M.S. Larry, *Multi-agent Systems and Societies of Agents*, in: G. Weiss, ed., *Multi-agent Systems – A Modern Approach to Distributed Artificial Intelligence* (The MIT Press, 1999).
- [27] R.D. Keeble, R.D. Macredie, Assistant agents for the world wide web intelligent interface design challenges, *Interacting With Computers* 12 (2000) 357-381.
- [28] M. Klusch, Information agent technology for the internet: a survey, *Data and Knowledge Engineering* 36 (3) (2001) 337-372.
- [29] C. Kolski, E. Le Strugeon, A review of intelligent human-machine interfaces in the light of the ARCH model, *International Journal of Human-Computer Interaction* 10 (3) (1998) 193-231.

- [30] H. Laïchour, S. Maouche, R. Mandiau, Traffic control assistance in connection nodes: multi-agent applications in urban transport systems, *Computing International Scientific Journal* 1 (2002) 37-42.
- [31] R. Mandiau, E. Grislin-Le Strugeon, A. Péninou, *Organisation et applications des SMA* (Hermes Editions, Paris, 2002).
- [32] M. Merad, G. Macaigne, S. Hayat, J. Rodriguez, Validation du module d'aide à la prise de décision des régulateurs, Final report, GRRT Cooperative Project, INRETS-ESTAS, Villeneuve d'Ascq, France, April 2001.
- [33] K. Mesghouni, S. Hayat, S. Hammadi, P. Borne, Modeling of the traffic regulation for a subway line with double loops, *Proceedings IEEE Systems, Man and Cybernetics*, San Diego, California, USA, October 11-14, 1998.
- [34] K. Mesghouni, E. Castelain, J.C. Gentina, S. Hammadi. Introduction du critère flux de passagers pour la régulation du trafic en station Final report, GRRT Cooperative Project, LAIL, Villeneuve d'Ascq, France, April 2001.
- [35] P. Millot, *Supervision des procédés automatisés et ergonomie* (Hermes, Paris, 1988).
- [36] N. Moray, Human factors in process control, in: G. Salvendy, ed., *Handbook of human factors and ergonomics* (John Wiley & Sons, 1997) 1944-1971.
- [37] J. Nielsen, *Usability Engineering* (Academic Press, Boston, 1993).
- [38] L. Nigay, Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales, PhD. Thesis, Université Joseph Fourier, Grenoble, France, 1994.
- [39] P. Palanque, R. Bastide, Synergistic modelling of tasks, users and systems using formal specification techniques, *Interacting With Computers* 9 (2) (1997) 129-153.
- [40] G.E. Pfaff, *User interface management system* (Springer-Verlag, 1985).
- [41] A. Péninou, E. Grislin-Le Strugeon, C. kolski, Multi-agent systems for adaptive multi-user interactive systems design : some issues of research, In: H.J. Bullinger and J. Ziegler, eds., *Human-Computer Interaction, Ergonomics and User Interfaces*, vol. 1 (Lawrence Erlbaum Associates , London, 1999) 326-330.
- [42] J. Rasmussen, *Information processing and human-machine interaction, an approach to cognitive engineering* (Elsevier Science Publishing, 1986).
- [43] B. Senach, Evaluation ergonomique des IHM : Une revue de la littérature, Technical report, INRIA, France, n°1180, 1990.
- [44] T.B. Sheridan, Task allocation and supervisory control, in: M. Helander, ed., *Handbook of Human-Computer Interaction* (Elsevier Science Publishers B.V., North-Holland, 1988).
- [45] B. Shneiderman, *Designing the user interface : strategies for effective human-computer interaction*, third edition (Reading, MA : Addison-Wesley, 1998).
- [46] F. Tarpin-Bernard, B. T. David, P. Prinet: Frameworks and Patterns for Synchronous Groupware: AMf-C Approach, In: S. Chatty, P. Dewan, eds., *Engineering for Human-Computer Interaction, IFIP TC2/TC13 WG2.7/WG13.4 Seventh Working Conference on Engineering for Human-Computer Interaction*, September 14-18, 1998, Heraklion, Crete, Greece, pp. 225-241.
- [47] M. Wooldridge, *An Introduction to Multiagent Systems* (John Wiley and Sons Ltd, 2002).