HAL open science

Modelling of agent oriented interaction using Petri Nets : application to HMI design for transport system supervision

Houcine Ezzedine, Abdelwaheb Trabelsi, Christophe Kolski

▶ **To cite this version:**

Houcine Ezzedine, Abdelwaheb Trabelsi, Christophe Kolski. Modelling of agent oriented interaction using Petri Nets : application to HMI design for transport system supervision. CESA2003 IMACS Multiconference Computational Engineering in Systems Applications, Jul 2003, Lille, France. pp.1-8. hal-03326765

HAL Id: hal-03326765

https://uphf.hal.science/hal-03326765v1

Submitted on 14 Feb 2024

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

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

In P. Borne, E. Craye, N. Dangourmeau (Eds.), *CESA2003 IMACS Multiconference Computational Engineering in Systems Applications (Lille, France, July 9-11, 2003)*, Ecole Centrale Lille, Villeneuve D'Ascq, pp. 1-8, ISBN 2-9512309-5-8.

Modelling of Agent oriented Interaction using Petri Nets, application to HMI design for transport system supervision

H. Ezzedine, A. Trabelsi, C. Kolski
L.A.M.I.H., UMR CNRS 8530 - Le Mont Houy
University of Valenciennes, F-59313 Valenciennes cedex 09, France
Phone: +33-(0)3-27-51-14-66, Fax: +33-(0)3-27-51-13-76, E-mail: {Houcine.Ezzedine, Abdelwaheb.Trabelsi, Christophe.Kolski}@univ-valenciennes.fr

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 will be validated via the specification and design of a human-machine interface used in the supervision of a land-based transport system (Bus/Tramway).

I. Introduction

Many applications currently offer Human-Machine interactions which are increasingly flexible, and even adaptive and intelligent [KOL, 98; HÖÖ, 00; CAL, 01]. 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 [KEE, 00; KLU, 01; GRI, 01].

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 [MIL, 88; SHE, 88]. 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 are 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 (such as real time aspects).

The second part describes the basic principles linked to multi-agent systems. 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. Basic Principles Linked to Multi-Agent Systems

The general principle of Multi-Agent Systems (MAS) 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.

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 [FER, 95; CHA, 92; CHA, 96]:

- 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. Architectures of Agent Oriented Interactive Systems

Several architecture models have been put forward by researchers over the past twenty years [FOL, 82; GOL, 84; PFA, 85; COU, 90; BAS, 91; NIG, 94; COU, 01]. 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…) [EZZ, 02].

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 [COU, 90], MVC (Model-View-Controller de 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). 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 *interactive agents* (or *interface 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.

- 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.

IV. Use of Petri Nets for Multi-agent Specification and Design

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. 1):

- 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,

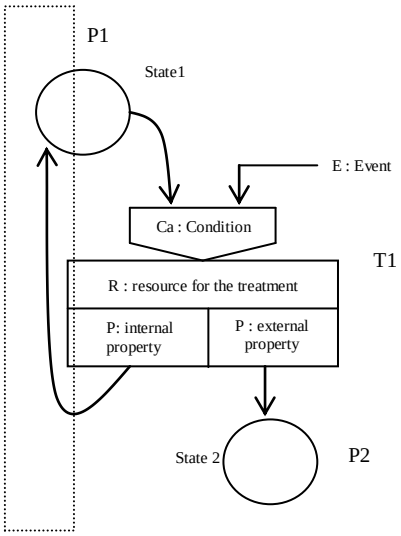

Fig. 1. Definition of an agent's service

- 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 …).

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

V. Application to a Supervision Interface for a 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 (LAIL, I3D, LAMIH and INRETS) [MES, 98; CHI, 00; EZZ, 01]. 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.

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, overground trains) want to be better 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.

A. 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 [PAL, 97; ABE, 01]. 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 (see Fig. 2). 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,

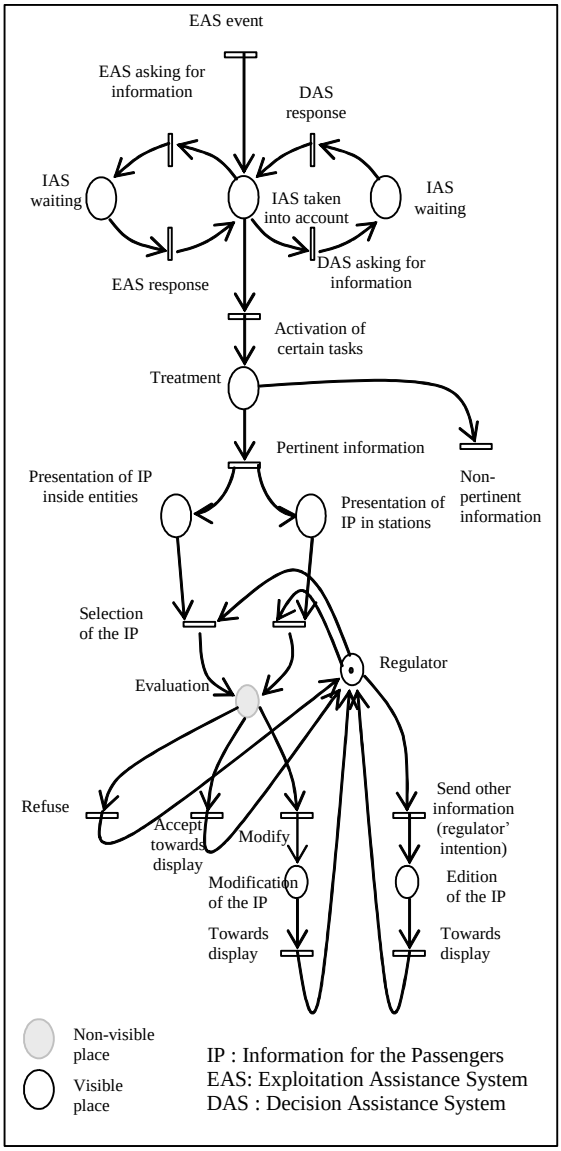

Fig. 2. Modelling of interaction

- 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.

B. Events linked to the IAS

As shown in Figure 3, 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.

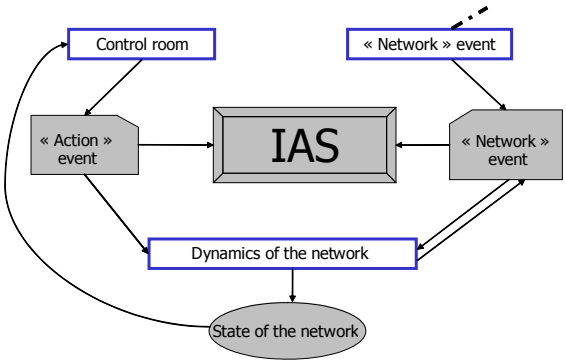

Fig. 3. Events linked to the IAS

C. 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).

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.

Specification of dialogue control agents: the dialogue control agents have a dual role. They make it possible to: (1) Diffuse the modifications observed at the application level on the interactive agents, (2) 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.

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.

D. Examples of the use of Petri nets for the definition of an agent's service

Example 1:

Table 1 and Figure 4 illustrate an internal operation within an interface agent.

TABLE 1
AGENT'S INTERNAL OPERATION

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

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.

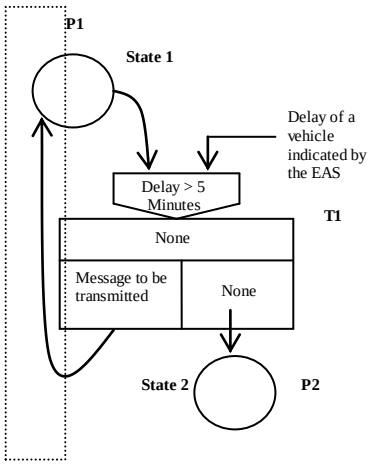

Fig. 4. Agent's internal operation

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. 5, 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.

TABLE 2
CALL FOR ANOTHER AGENT'S SERVICE

Event	Click on a vehicle
Condition	Communication established with the vehicle agent
Resource	None
Internal property	None
External property	Call for the service of the vehicle agent

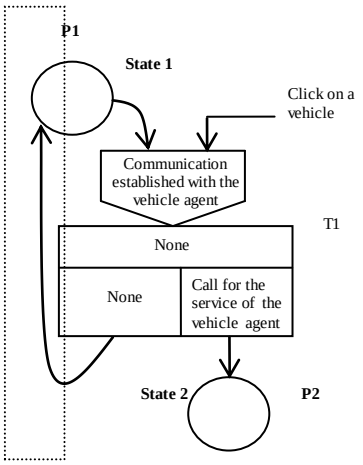

Fig. 5. Call for another agent's service

E. 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.

The *State of the traffic* 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.

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. 6). 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.

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 (See Fig. 7.b). It shows the regulator the information contained in the running plans in the

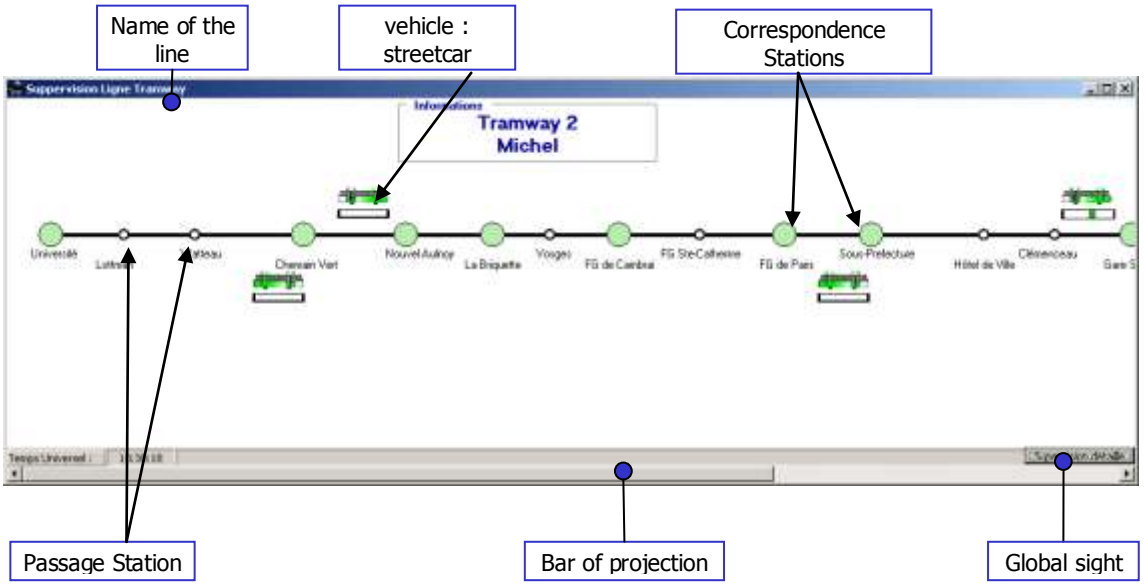

Fig. 6. View of the State of the *line* interface agent

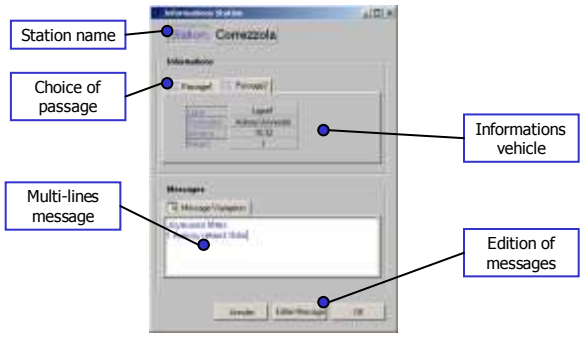

Fig. 7.a. View of the *Station* interface agent

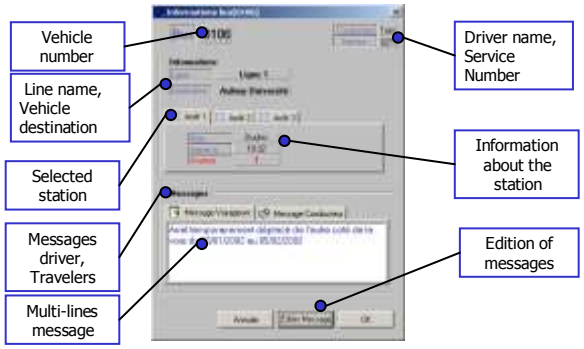

Fig. 7.b. View of the *Vehicle* interface agent

of a set of thumbnails depending on a direction which can be selected on a drop-down scroll list (See Fig. 7.a).

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 (see Fig. 8): 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.

The *Overall view* interface agent: in order to make the task of supervising the traffic easier for the regulator, we created an *overall view* interface agent. As its name implies, the view given by this agent provides the regulator with a global view of the traffic on the network. This view encompasses all the lines to be supervised and facilitates access to line, stations and vehicles.

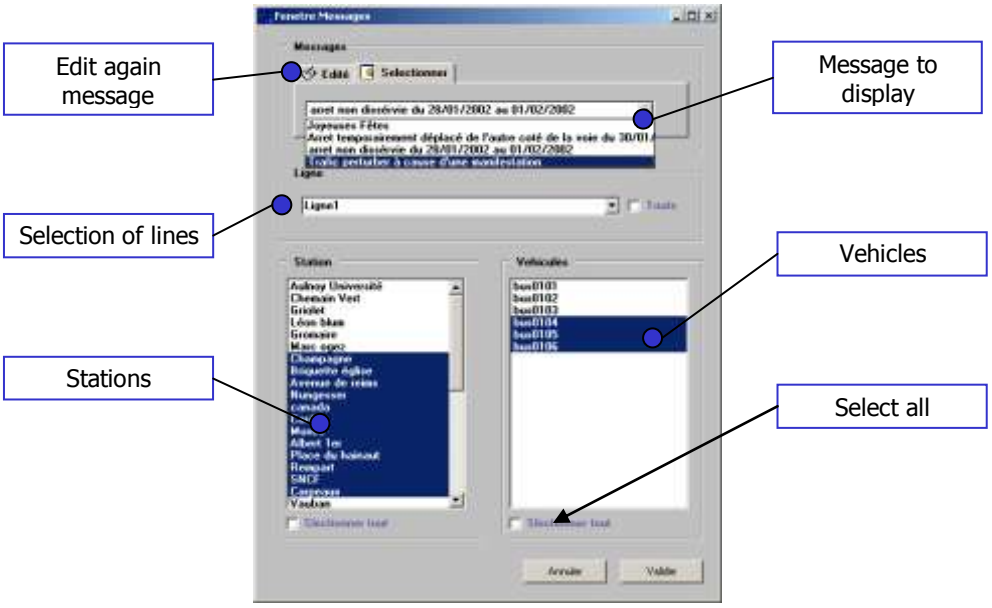

Fig. 8. View of the *Message* interface agent

VI. Conclusions and Perspectives

In this article, we have put forward an agent orient 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 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 and NIPO projects).

References

[ABE, 01] M. Abed, H. Ezzedine, and 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, pp. 145-174. Paris : Éditions Hermes, 2001.

[BAS, 91] L. Bass, R. Little, R. Pellegrino, S. Reed, et al. The Arch model : Seeheim revisited, Proceedings of User Interface Developers' Workshop, Seeheim, 1991.

[CAL, 2001] G. Calvary, J. Coutaz, and D. Thévenin, "Supporting context changes for plastic user interfaces: a process and a mechanism", In Blanford A., Vanderdonckt J., Gray P. (Eds), People and Computer XV – Interaction without Frontiers, Springer, pp. 349-363, 2001.

[CHA, 96] B. Chaib-Draa, "Interaction between Agents in Routine, Familiar and Unfamiliar Situations", International Journal of Intelligent et Cooperative Information Systems, vol. 5, n°1, 1996.

[CHA, 92] B. Chaib-Draa, B. Moulin, R. Mandiau, and P. Millot, "Trends in Distributed Artificial Intelligence", Artificial Intelligence Review, Kluwer Acadimic Publishers, Great-Britain, vol.6, pp.35-66, 1992.

[CHI, 00] F. Chihaib, S. Hammadi, and P. Borne, "The Contribution of Linear Programming by Constraints Propagation In the Regulation of Traffic of Urban Transport Network", Proceedings ITS 2000 7th World Congress On Intelligent Transport Systems, Torino November 6-9, 2000.

[COU, 90] J. Coutaz, Interface homme-ordinateur, conception et réalisation, Dunod, Paris, 1990.

[COU, 01] J. Coutaz and 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, pp. 207-246. Paris : Éditions Hermes, 2001.

[EZZ, 01] H. Ezzedine, H. Maoudji, F. Cartegnie, and A. Péninou, "Towards agent oriented specification of Human-Machine Interface : Application to the transport systems", 8th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, pp421-426, Kassel, Germany, 2001.

[EZZ, 02] H. Ezzedine, Méthode et modèles de spécification d'évaluation des interfaces homme-machine dans les systèmes industriels complexes, Habilitation à diriger des recherches, Université de valenciennes et du hainaut-cambrésis, 2002.

[FOL, 82] J.D. Foley and A. Van Dam, Fundamentals of interactive computer graphics, Addison-Wesley, 1982.

[FER, 95] J. Ferber, Les systèmes multi-agents, vers une intelligence collective, Paris : InerEditions,1995.

[GOL, 84] A. Goldberg, Smalltalk 80 : The Interactive Programming Environment, Addison-Wesley Publ., 1984.

[GRI, 01] E. Grislin-Le Strugeon, E. Adam, and 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, pp. 207-248. Paris: Éditions Hermes, 2001.

[HÖÖ, 00] K. Höök, "Steps to take before intelligent user interfaces become real", Interacting with computers, pp. 409-426, 12, 2000.

[KEE, 00] R.D. Keeble and R.D. Macredie, "Assistant agents for the world wide web intelligent interface design challenges", Interacting With Computers, 12, pp. 357-381, 2000.

[KLU, 01] M. Klusch "Information agent technology for the internet: a survey", Data and Knowledge Engineering, 36 (3), pp. 337-372, 2001.

[KOL, 98] C. Kolski and 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), pp. 193-231, 1998.

[MES, 98] K. Mesghouni, S. Hayat, S. Hammadi, and 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.

[MIL, 88] P. Millot, Supervision des procédés automatisés et ergonomie, Hermes. Paris, 1988.

[NIG, 94] L. Nigay, Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales, Thèse en informatique, Université Joseph Fourier, Grenoble, 1994.

[PAL, 97] P. Palanque and R. Bastide, "Synergistic modelling of tasks, users and systems using formal specification techniques", Interacting With Computers, 9 (2), 1997.

[PFA, 85] G.E. Pfaff, User interface management system, Springer-Verlag, 1985.

[SHE, 88] T.B. Sheridan, "Task allocation and supervisory control", In Helander (ed.), Handbook of Human-Computer Interaction, M. Elsevier Science Publishers B.V., North-Holland, 1988.