A model of Environment, active support of the communication

Flavien BALBO[1 2]

[1] Lamsade, Paris Dauphine University
Place Maréchal de Lattre de Tassigny
75775 Paris Cedex 16, France
balbo@lamsade.dauphine.fr

[2] Inrets-Gretia, French National Institute for Transport and Safety Research
2, avenue du Général Malleret-Joinville
F-94114 Arcueil cedex
balbo@inrets.fr

Abstract

The aim of our work is to propose a hybrid model allowing the representation of shared contexts. The originality of our approach is to propose a model of the environment where the agents evolve. We consider the environment as an active support of communication, which searches interested agents for each deposited message. We consider this environment as a shared context of work, which means that each transmitted message is available for all agents and each agent specifies the messages it is interested in, by writing a rule in its environment, thus allowing their reception.

We present in this article the model of the proposed environment. We describe the characteristics of the agents and an adapted communication protocol. This model will be illustrated by an application coming from the field of public transport.

1 Introduction

Each multi-agent system is designed with agents evolving in an environment. While the concept of agent has been the subject of many works, there are not many models of the notion of environment. This can be explained by the lack of control on the activity of the environment in some applications. For instance, Internet is only a passive support for communications and agents (Decker, Sycara and Williamson 1997) do not have any control on it. In the same way, in simulated environments, we find this distribution of control between the agents and the environment. Thus Le Strugeon (Le Strugeon 1995) defines the environment as:

> A static global entity, which does not develop by itself but according to the actions, carried out on it.

On the contrary, other systems choose to assign a significant part of control to the environment. Then Magnin in (Magnin 1996) represents the environment as being made of "environmental rules" representing the laws of the world (here a football game). In the same way, Wenger in (Wenger and Probst 1998) uses a black board, which allows the agents described as autonomous to become active according to the coupling between its content and their skills. Despite of these significant differences in the assignment of control, we can point out a common characteristic for all of these more or less implicit representations of the environment: it allows to connect the agents in the broad sense. The form range from a more or less direct communication support (Decker, Sycara and Williamson 1997) (Le Strugeon 1995), to the management of interactions (Magnin 1996) (Wenger and Probst 1998).

From this analyze, we have chosen to study the possible contribution of the environment as a communication support. Decker, in (Decker, Sycara and Williamson 1997), where the environment is only a physical support of communication, describes the need to use intermediate agents in order to allow the agents to better communicate. Connection, a traditional problem in the multi-agent paradigm, consists in connecting a caller for information or services with the most suitable provider. The various technologies used by this intermediate agent are based on the same principle: to make the best matching between the skills of the providers and the needs of the callers. What justifies the choice of an agent to fulfill this function is that it is not possible to assign any part of control to the environment. Indeed, we think that this task of connection between agents (through communication) can be assigned to the environment, which we then will define as *an active support of communication*. This option of development allows to widen the concept of message addressing.

In the example of application we chose, we will see that an agent may wish to communicate with another one according criteria, which do not depend on particular skills and evolve in a dynamic way (like the position of a bus on a network). Thus, for a message, the problem consists in finding back the right recipient according to the characteristics required by the transmitter. The current approach consists in putting the characteristics of

reception in the message. Then each agent decides on how to process the message. In the case of a traditional diffusion, a badly addressed message will be removed. In the case of a diffusion with stages (Decker, Sycara and Williamson 1997) (Malville and Bourdon 1998), the message will be sent again to agents likely to be interested in it. If we separate the message from the characteristics of reception, a knowledge base, which we will define as the environment of our multi-agent system, will find out the right recipient of a message.

The environment is then an active support of communication looking for the potentially interested agents for each of the messages sent. In that way, we can consider this environment as a shared context of work, where each message sent is available for all of the agents. Each agent specifies the messages he is interested in by writing a rule in its environment, making them available for reception.

In the following section we will present an example to illustrate the problem we are dealing with. Chapter 3 will describe the structure of our environment, chapter 4 will identify the modules required by the agents in order to use this environment. Chapter 5 presents examples of interactions from our application and we then will conclude our presentation.

2 Example

We will illustrate our model by an actual application in the field of public transport. Our aim is to define a system representing the activity of a public transportation network. After an analysis of the problem, we choose to define two types of agents: STOP agents and BUS agents. Each stop calculates a schedule for a Bus to come on its line at its position and the BUS agents account for the activity of the vehicle by announcing when they will arrive at the stop concerned. A STOP agent can trigger an alarm if an expected BUS has not been announced while its schedule is over.

The system runs with actual data from a public transport company. The positions of the vehicles come from sensors located on the network, and the data provided can be uncompleted. A vehicle may not be announced on stops he has served thus generating a false alarm, i.e. declared as late while it passed over the stop. We will see how our architecture allows to manage these inconsistencies without using broadcasting.

3 The environment as a support of interactions

We have defined the environment as the active support of communication, the result being a knowledge base, the construction and activity of which come from the communication needs of the agents.

In the paragraph about the model of agent we have defined, we will see how this environment is built on the basis of the rules specifying the communication needs of each of the participating agents. In this chapter we will describe the formal characteristics of the exchanged messages and their consequences on the organization of the agents.

Our goal is to allow the exploitation of the work already achieved to define high level languages such as KQML (Patil et al. 1992). Each of the languages supposes that the recipient is known and provides only a semantic allowing a high level of interactions. Our aim is complementary and consists in adding additional information to the structure of these messages in order to allow their processing by the environment, i.e. under the BNF form.

<message>:: =<environment><reception> <content>

In this article, we will describe the characteristics corresponding first to the environment, second to the reception.

<environment>::=<priority> <limit of existence>

The first layer of information describes the data allowing the environment to process the message, i.e. the priority of the message and the limit of existence of the message. The management of conflicts in the environment (in the sense KBS) is ensured by a priority level contained in the message (attribute *?significance* of rule). Each agent gives a priority level to the message he sends, corresponding to the significance of the rule which allows its reception, e.g. an agent can put a priority on its messages whenever it has detected an alarm. The control of the environment (the choice of the rule) thus results from the communication of the SMA, the priority of the message processing depending on the needs specified by the agents.

A message sent to only one recipient will be destroyed by the latter, whereas the environment will destroy a message with several potential recipients. Each agent decides on the life duration of the message it sends attributing a value to the characteristic of *limit of existence*. This limit may correspond to a date or to a state of the system, the choice being made according to the needs of the application.

<reception>::= <test> '('<condition>')'

The second layer of information corresponds to the data, which will make it possible to find the recipient of a message. The first characteristic, called *test*, makes it possible to know the order and nature of the characteristics sought at the recipient's. The second attribute puts together all of the values, which will be used in the test. In order to allow comparisons, we will use all the mathematical comparators, the equality being understood by default. We also use all the predicates. The use of a predicate means that the condition of reception of a message requires a comparison between the potential

recipients. For instance, we have defined the predicate *>First Position*, which enables to find back the first (the most remote in the sense of the position on the line) of an agent's predecessors.

In that way we obtain "a logic of communication" enabling every agent to find its interlocutor according to the characteristics it seeks at this agent's.

For instance, a message with a name of test equal to "*<FirstPosition_NextB*us", means that the transmitter seeks the first of the STOP agents located before the position referred to in the message. The receiver must also wait for the BUS agent, the identifier of which corresponds to the attribute *NextBus*.

The recipient receives only the content of the message. The message is processed in the same way as a message exchanged by two agents in a traditional SMA.

After the study of the structure of the messages we can deal with the definition of the agents enabling their exploitation.

4 The Agents.

In this section, we will describe the agent's modules required for our architecture. The base of knowledge, which constitutes the common environment, must find the right recipient for each message. Therefore, an agent must provide the rules enabling him to receive the messages he is interested in.

In order to be identifiable by the environment, the agent must also provide all the information characterizing him. In this objective as in (Jaber, Guarnieri, and Wybo 1998) each agent has a public layer containing this information. These data are not directly accessible to the agents, but only to the base of knowledge controlling the communication. This information can be fixed or changing with time. An agent wishing to send a message must have the description of the public layer of the agent's category he wants to contact. By specifying the values of the characteristics he seeks, it is the transmitter who chooses the agent he wishes to communicate with.

For example, in our application, the public layer of the STOP agent will contain the following static information: (1) an identifier for a direct sending of a message to the agent ; (2) the number of the line allowing to identify the STOP agent as belonging to a sub-group of the network (authorizing a limited broadcast to this unit) ; (3) a sequence number on the line which allows a relative addressing.

The public layer will also contain information changing with time, such as the identifier (information *NextBus*) and the schedule (information *NextSchedule*) of the next BUS to come. Contrary to the whole planning containing the schedule of the BUS it has to control, this information is made available for the network.

Thus, an agent can be identified in the environment (through the definition of the *Public Layer*), but he still has to be able to receive messages. To make this possible, each agent has a module representing his knowledge of the environment. This module is made of two parts. The first one contains all the rules enabling him to participate to the environment, the second contains the information about his relatives

The module containing the rules should not be considered as an expert system of communication, but as a module containing all of the sensors enabling the agent to participate to the environment. The reception rules allow an agent to receive messages directly addressed to him, and they generally refer to a single static characteristic of the agent. In our application, they are rules receiving messages with an addressing criteria based upon the identifier.

```
(defrule rule_1 { ?significance + 1}
    ?m: (Message test = « identifier »
        condition1 = ?np transmitter= ?nb
        significance = ? significance)
       (STOP identifier = ?np position= ?pos)
    ?a: (STOP position < ?pos nextBus = ?nb)
    ->
       { ?a->recevoir( ?m) ;}
    )
```

Rule 1: Interception Rule.

An agent can add to the environment interception rules of messages, which are not directly addressed to him (rule 1). Thus, a STOP agent can intercept all the messages from the BUS from which he awaits a transit advice note (see the example on the management of the inconsistencies).

With the last type of rules, an agent can contact other agents (using a conjunction of the characteristics of the public layer of the potential recipient) even if the latter has no rule to receive the message. Thus, if the recipient has the initiative to create the former rules (he indicates the characteristics of the messages he is interested in), it is the transmitter who has the initiative to create the latter type of sensors. The purpose is to allow the transmitter to send a message to an other agent who didn't contemplate to receive this message. Thus an agent may require information at the first stop waiting for a particular bus (rule 2). This information consists in a conjunction of characteristics from the *Public Layer* module of the STOP agent. It would not be efficient to make this agent contemplate all of the combinations from the *Public Layer* model. Thus they have to be chosen by the transmitter.

```
(defrule  Rule_2  ?significance
?m:(Message
    test=«<=FirstPosition_NextBus»
    condition1 =?position condition2 =?bus
    significance = ?significance)
?a:(STOP
    position<= ?position&= ?positionCourante
    nextBus = ?bus)
(not (STOP position < ?positionCourante
    nextBUS = ?bus))
-> { ?a->receive( ?m) ;})
```

Rule 2: Seek of a particular agent

For each rule the agent can determine whether it is a persistent or a specific rule. The environment will be built on the basis of persistent rules belonging to a category of agent. It means that the environment is the result of the union of the rules of communication of the agents wishing to take part in it. Thus, in our application, the environment is made of reception rules for the messages of the BUS and STOP agents. Rules with a more limited life span can be added to this base. The rules being interpreted, it is possible to add some other rules to the system in a dynamic way, according to the needs of the agent. This functionality has not yet been developed, but it would of course be of interest. It makes it possible for an agent to punctually recover information, which interest it only under certain conditions. Thus, an agent having located a delay may wish to receive information about traffic conditions in order to make a better estimation of the consequences (or of the causes) of this delay. If this information is available (for example from a road traffic management system) then he can receive it by adding a rule. Once the delay is removed he will withdraw his rule and "unsubscribe" this information source.

Each agent has a module of contacts providing two types of information: the contacts by category and the particular contacts. The contacts by category correspond to the description of the *public layer* of a type of agent. A BUS agent has the description of the *public layer* of the STOP agent and vice versa. This format of contacts is little used, but one of our objectives is to introduce a module of rule generation allowing to add personalized rules to the system from this knowledge, according to the needs of an agent. The second type of contacts is more traditional, since it contains information making it possible for an agent to communicate with agents with whom he has privileged relations. For example, a STOP will only know the precise references of the stop situated just before and after it on the line. Thus it will communicate only with the STOP at any time using their identifier.

After the presentation of the agents and their interactions modes, we will show in the following chapter how these architectures allow the agents to organize.

5 Organization

We will rely on our application in order to highlight the potentialities of organization. To do that, we will examine how with two rules of reception, the STOP agents ensure the coherence of information. In this section, anomalies are to be understood as a failure in information transmission and thus a violation of the minimum interaction constraint. We will not deal with the anomalies linked with the transport network activity.

The anomaly is a defect of a transit advice of a bus to come at the stops provided. The consequences are twofold: (1) STOPS remain waiting for a message whereas the bus physically has passed, (2) STOPS may receive messages of transit advice for buses they are not waiting for immediately but which have been announced.

The structure of the message we have used (the third level of communication) provides additional information. These data correspond to the semantics of the language of interaction we use. Since the present article does not deal with the description of the language, we will not specify its syntax. However we must specify that each message contains the identifier of the transmitter, of the receiver (which can be null if the information is unknown of the transmitter), as well as the nature of the message. With these data a STOP agent can build a rule of interception which enables him to process the first consequence of the defect of data transmission. In order to satisfy our objective of optimization of the communications, this rule will be built using the characteristics of the message concerning the transmitter, like previously, but also the identifier of the receiver and the nature of the message. The knowledge of the receiver allows, through the comparison of the *position* information contained in the public layer of the STOP agents, to attribute the message only to the agents located on the line before the recipient, who are waiting for news from the transmitter. The knowledge of the nature of the message still allows to limit the intervention of the rule.

We have described the structure of a more complex rule, and we now must precise how this rule will be activated. In the message he sends, each agent sets its priority, this value becoming the level of priority of the rule allowing the addressing of the message. In our case, a transit advice note is directly addressed and thus will be destroyed by the recipient. An agent wishing to intercept this message thus must be sure that its rule will be triggered before the rule of the recipient, whatever priority level is given to the message. The intercepting agent must therefore use the same principle of dynamic priority by attributing to the rule a priority equal to the priority of the message plus one (at least). In that case, whichever the priority of the message to be intercepted, the interception rule will be activated before the rule of the initial recipient.

The second problem corresponds to the case where a bus would not be announced on a set of stops, whereas its

successor is announced on a subset of these stops. Figure 1 represents this case: BUS b1 was announced at STOP 2 and, then will only be announced at STOP 10. At the same time, before the announcement of the first bus at STOP 10, b2 is announced at STOP 3 and then at STOP 5. According to the schedule, the STOPS are waiting for b1, which has physically passed them. Thus at the arrival of the message from BUS b2 agent, it is not waiting for that bus (as STOP 4 agent). However it takes this passage into account at two levels:

1) it updates its schedule, considering this vehicle as having passed and remains in standby for BUS b1
2) it gives this information to its predecessor, which also updates its schedule. Since this event is new to it, it transmits this information to his own predecessor (STOP 3) which does not take the message into account since this BUS already has passed for it.

All of the STOPS waiting for BUS b1 will stay on standby until the announcement of the bus at STOP 10. The announcement will then be intercepted according to the previous procedure allowing all of the STOPS to update.

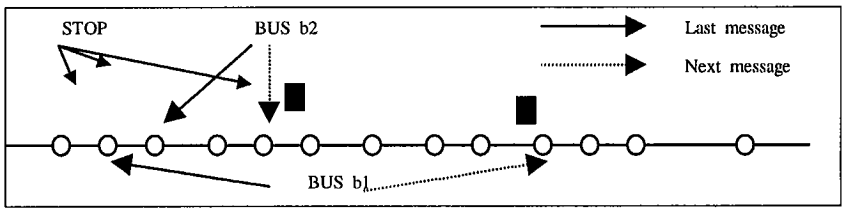

Figure 1: case of incoherence

6 Conclusion

In this document, we presented an architecture allowing to represent shared contexts. We described an environment of communication enabling co-operative work between agents, which we implemented using ILOG RULES. This architecture was used in order to represent the activity of a transport network. We also have demonstrated that this architecture enriches the use of communication in a SMA, by widening the possibilities to address a message. In a traditional environment an agent can, if it needs to communicate, either address its message directly, or broadcast it more or less widely (if it does not know the receiver directly). In our architecture, this same agent uses the support of communication that represents the environment in order to find the recipient.

Our architecture also makes it possible to use several environments simultaneously. In that case, communication is not only intra environment but also inter environment.

One of our immediate objectives is to improve the dynamics of the environment, allowing the agents to actually add rules during the execution. Initially these rules will have to be already known by the agent to be added according to pre-defined conditions. In a second step these rules should be built according to the needs of the agent and to the characteristics of the potential receivers.

References

Decker, K. Sycara, K.Williamson, M. 1997. Middle-Agent for the Internet », In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 578-583, Morgane Kaufmann.

Le Strugeon, E 1995. Une Méthodologie d'Auto-Adaptation d'un Système Multi-Agent. Ph.D. diss., Dept. Of Computer Science, Valenciennes Univ.

Magnin, L 1996. Modélisation et Simulation de l'Environnement dans les Système Multi-Agents, Application aux Robots Footballeurs . Ph.D. diss., Dept. Of Computer Science, Paris VI JUSSIEU Univ.

Wenger, D., Probst, A.R. 1998. Adding Value with Intelligent Agents in Financial Services. In Jennings,N.R., Wooldridge, M.J. eds. *Agent Technology Foundation Applications and Markets.:* Springer press

Malville, E., Bourdon, F. 1998. Task Allocation: A Group Self-Design Approach. In Proceedings of the Third International Conference on Multi_Agent Systems, Paris, France: IEEE Press.

Patil, R., Fikes, R., Patel-Schneider, P., McKAy, D., Finin, T., Gruber, T. and Neches, R. 1992. The DARPA Knowledge Sharing Effort:Progress Report. In *Proceeding of the Third International Conference (KR'92)*, San Mateo, Calif.:Morgan Kaufmann Press

Jaber, A., Guarnieri, F., and Wybo, JL., 1998. Un Système d'Agents Logiciels Intelligents Pour Favoriser la Coopération entre des Systèmes d'Aide à la Décision Dédiés à la Gestion de Crise. In Proceedings Journées Francophone Intelligence Artificielle Distribués et Système Multi-Aagent.Hermès Press.