An Agent oriented approach to Transportation Regulation Support Systems *

Flavien Balbo
Université Paris-Dauphine
LAMSADE/CNRS - GRETIA/INRETS
Place du Maréchal de Lattre de Tassigny
75775 PARIS Cedex 16 Paris, France
balbo@lamsade.dauphine.fr

Suzanne Pinson
Université Paris-Dauphine
LAMSADE/CNRS
Place du Maréchal de Lattre de Tassigny
75775 PARIS Cedex 16 Paris, France
pinson@lamsade.dauphine.fr

ABSTRACT

This paper presents an agent-based approach to design a Transportation Regulation Support System (TRSS). Based on a multi-agent modeling of a urban transportation network, the objective of our approach is to integrate the functionalities of the existing information system with the functionalities of a decision support system. The TRSS monitors the network activity and adjusts itself to the environment changes, that is to say it automatically detects incoherent data (regulation under normal conditions) and traffic disturbances and then it automatically proposes solutions to optimize the traffic flow (regulation under disturbed conditions). To demonstrate our approach, a transportation regulation support system called SATIR (Système Automatique de Traitement des Incidents en Réseau - Automatic System for Network Incident Processing) is presented. SATIR has been tested on the Brussel transportation network (STIB). Lastly, we show how using the multi-agent paradigm opens perspectives regarding the development of new functionalities to improve the management of a bus network.

General Terms

public transportation network management

Keywords

agent-based applications, environment, Decision Support System

1. INTRODUCTION

The development of the surface public transportation networks (SPTN) is a main issue from the ecologic, economic and societal viewpoints. But, this means of transportation competes against the comfort of the personal vehicles and contrary to the guided transportation (train and subway) it has to support the traffic disturbances. The result is that SPTN are often considered as not reliable. If the projects like the Bus Rapid Transit highlight the benefits of an improvement of the quality of the infrastructures, a better management of the available resources is less costly. For Intelligent Transportation System, the objective of the network management is to improve the attraction for public transportation with a decrease of the waiting time and an increase of the commercial speed. This improvement has to be done without an increase of the management costs. Efficient planning algorithms and a more precise knowledge of the network state improve the management of the resources. Nevertheless this improvement of the theoretical supply has to go with an improvement of the management of the network in real time. Indeed, the optimum that is the theoretical supply may become obsolete according to the evolution of the urban traffic. Regulators (the staff in charge of monitoring the bus networks) have to ensure the success of the transportation plan, in the sense of adapting the theoretical supply to the real evolution of the demand. In order to achieve their complex task, regulators use systems known as Automatic Vehicle Monitoring systems (AVM). In this paper, we show that if the use of an AVM is the first step to the computerization of the transportation network activity, this system is limited to coping with disturbances linked to unanticipated demands and to traffic conditions. The collecting and shaping of data are insufficient to help regulators and this system has to be completed with a Decision Support System (DSS) able to analyze this information and to give in real time a dynamic and contextual assessment of the problems. Our proposition takes place in the Multi-Agent Decision Support System (MADSS) domain. The DSS should not be based on black boxes ([13]) but on paradigms that are more collaborative and active. By reification of actors and of their use of the information and by distribution of control, a Multi-Agent System (MAS) makes explicit the process it has to manage.

Section 2 describes the real time management of an urban transportation network, the advantages and difficulties of the use of the current information system are underlined. Section 3 presents the alternatives for the integration of a DSS to the current information system. Section 4 describes the SATIR (Système Automatique de Traitement des Incidents en Réseau) project that is our proposition to integrate a DSS to an AVM. Section 5 details our experimentation. Section 6 proposes a conclusion.

*(Produces the copyright information for AAMAS 2008). For use with aamas2008.CLS and aamas2008-short.cls

2. URBAN TRANSPORTATION SYSTEMS: STATE OF THE ART

In this section, the processing of information for the management of an urban bus network in real time is presented. The first part presents the model of data for the urban transportation domain and the information system (AVM) based on it. The second part is related to the regulation task.

2.1 An Automatic Vehicle Monitoring (AVM) System

2.1.1 Data Model

In Europe, the first project for a data model has been CASSIOPE (1989-1992) the result being the Transmodel[1]. This model is the European reference of the conceptual data modeling for public transportation domain. Transmodel has been improved with the European projects Eurobus (1992-94) and Harpist (1995). Finally, the Titan project (1996-98) (Transmodel based Integration of Transport Application and Normalisation) has completed and validated this work. The result is a modeling based on the Entity/Association formalism. The objective is to represent in the same modeling the physical and timetable configuration of a network. A hierarchical decomposition of the information has been adopted. The physical configuration is based on the decomposition of the network in lines and for each of them in routes. A route is decomposed in sections and for each of them there are a stop and an inter-stop distance. The timetable configuration is based on the decomposition of the timetable in missions and for each of them in runs. This modeling has been done to answer to a specific need: to access in the easier way to a specific component of the timetable, like the description of the route of a run or for a section the list of the schedules for a period of time.

2.1.2 Functionalities

In order to use this information modeling, the transportation networks use a specialized information system called Automatic Vehicle Monitoring (AVM). With the theoretical information (network description and timetable), the AVM computes in real time the theoretical state of the network. In order to compute the real state of the network, an AVM uses the data coming from vehicles located by sensors. AVM compares the actual positions of vehicles (captured by the sensors) with their theoretical positions in order to provide the regulator with an overview of the routes. In this way, the regulator can see whether vehicles are running ahead of timetables or are running late. By comparing theoretical information with real one, the AVM system tries to detect delays and advances of buses on the network. Some AVM systems propose detections depending on a geographical condition like delay/advance alarm in a town-centre or depending on a timetable condition like the detection of the delay on the next departure. In real time, an AVM organizes the collecting and shaping of data; it facilitates the access to this data and computes some alarms. The main objective of the AVM is to give a basic information computation. An AVM works according to a classical way for the diagnostic domain: model-based approach relying on normal and faulty behavioral models. This approach is based on the comparison between a theoretical modeling of the diagnosed system and an artefact of the real system. Since the theory of the regulation domain is not complete, the AVM is only able to support this approach and the result is mainly dependant of the experience of the regulators.

2.2 The regulation task

This section presents the new functionalities of the AVM to support the work of regulators. We first describe the regulation process and highlights the limits of the AVM system in real-time management of bus network.

2.2.1 The regulation process

Analysis of the work station of network regulators from the Brussels Intercity Transport Company (Société de Transport Intercommunale de Bruxelles - STIB) enabled us to identify four phases required in the regulation process. First the regulator begins by monitoring the network.

The diagnostic phase begins with the detection of a problem and ends with the assessment of its consequences on the network activity. The interface of the AVM facilitates this phase. The current alarms are visually accessible and relieve regulators of computation. As soon as a disturbance is chosen by the regulator, he has to complete his knowledge of the problem. This process is complex because disturbances evolve independently along three axes. The Time axis measures the seriousness of a disturbance according to the timetable. The Space axis measures the seriousness of a disturbance according to its position on the network. The Shape axis measures the consequences a disturbance may have on the network activity. To determine its importance, a disturbance must be evaluated according to these three axes. For example, a vehicle having off-peak hour difficulties in a suburb (a disturbance that is not critical a priori) may cause a real problem if bus frequency is low.

When a disturbance has been detected and assessed, the planning phase begins. The type of the risk gives the primary objective to the regulator: to increase/decrease the supply at one part of the network. He computes the feasible procedures according to the current state of the network. This complex process involves various information sources like the real-time information, the theoretical timetables or the information coming from drivers and from other regulators, etc.

The decision phase is the last phase of the regulation process. There are two parts, firstly the regulator chooses the regulation procedure and secondly he monitors its execution. If the result of the planning phase is a set of feasible procedures, the regulator has to find a compromise between contradictory constraints. For example, the empty runs or the failures of the regulation's procedures have to be limited. After this choice, the regulator has to monitor that the resources related to the procedure and the state of the network evolve according to the forecasting of the regulator, otherwise, he may have to change his choice.

Figure 1 sums up the tasks of the regulator and the links between them. The regulator is involved in all of them. This multiplicity of tasks makes his work very complex. Note that the decision task that is based on the result of the diagnosis and planning tasks should be his main activity.

2.2.2 Regulation process issues

The main advantage of the AVM is to facilitate the access to numerous and heterogeneous information sources, but the

[1] http://www.transmodel.org

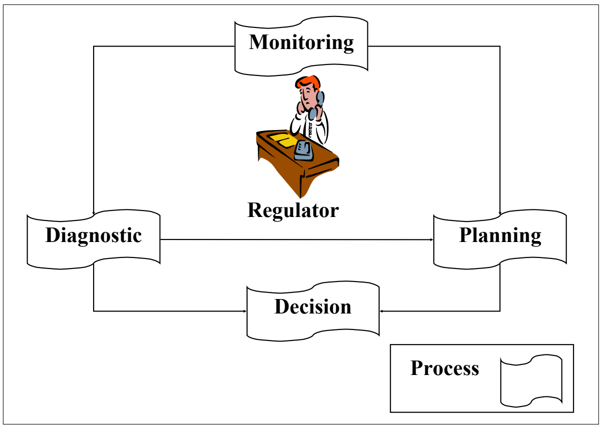


Figure 1: The regulation process in a urban transportation system

regulator has to use his experience to extract the right information at the right time. This information organization implies that the regulation process is strongly dependent on regulator experience.

The first issue is related to the information incompleteness. An AVM gathers information but part of them may be missing according to the success of the collecting process. Moreover, the regulator work is based on completing information as timetables miss the necessary knowledge to compute them. This knowledge is essential for a better management of the transportation plan. The theory of the domain proposes different rules (called logics) for the regulation of a network and they underlie the computation of the timetable. If the description of these logics is not in the scope of this paper, remember that each of them is related to a specific objective. For instance, the logic of regularity aims at minimizing the average waiting time of the travelers and to divide the supply on all the buses in the line. On the contrary, the logic of taking away aims at respond to a local important demand. In this case, the timetable is computed in order to concentrate the resources on the most critical points in the line in order to keep all the passengers. Consequently, in case of a disturbance, the regulation process should not be based on the theoretical state of timetable but on its objectives. This change in the regulation process is only possible for experimented regulators, who find the missing information with their knowledge of the network.

The second issue is related to the distribution of the regulator's attention. Because of the urban traffic, numerous vehicles in the transportation network are late especially at the peak hours. A regulator can not take into account all these delays. An experimented regulator uses his knowledge of the line structure (the position in the city, the presence of difficult areas) and of the demand structure to determine the most critical lines according to the schedule. For instance, information indicating that a vehicle is too late to do its next departure may be useless if this vehicle can make up for the lost time on the last part of the run. The regulator uses the computed alarms as primary indicators and analyzes its real importance according to its context.

The third issue is related to the assessment of a disturbance by the regulator. This task implies that the problem evolution on the network is taken into account. Primary alarms on the advance/delay of each bus provide instantaneous picture of line conditions. Monitoring all these alarms in their space-time development is almost impossible and leads to extra work for regulators. The regulator has to forecast in a time-window the evolution of the situation. This forecasting takes into account the future buses positions in order to estimate the real importance of the problem. The regulator has to choose the information according to its own expertise and the estimate state of the network.

The fourth and last issue is related to the lack of a global vision. The splitting up of monitoring by line and the high number of the lines to be monitored (each regulator tracks 13 lines with 5-20 buses running during the day in the STIB network) prevent global management of the network. Regulators use their expertise to relate disturbances without visible links, in order to propose global solutions.

These issues are based on the lack of information modeling to help a novice regulator. Our proposition is to integrate a Decision Support System into the information processing in order to support regulators task. This evolution from an information system to a DSS corresponds to the evolution of applying computer science in the transportation domain. The first step has been to formalize the domain knowledge in order to create an information system for an efficient data management. The objective of the next step is to automate the use of this data through the design of a DSS.

3. INTEGRATING A DSS AND AN AVM SYSTEMS

The cooperation between a man and a system during a decisional process can be done in two different ways: 1) vertical cooperation; 2) horizontal cooperation. The difference between them depends of the distribution of the control. The first choice is to use the DSS as an information source and the operator is responsible of the decision. The second choice is to have a DSS that computes in parallel to the operator in order to reduce its work. This part discusses the choice of the cooperation type that is the best suitable for the urban transportation regulation domain.

3.1 Choice of the system architecture

The issue is to find the right place of the DSS during the current information process. In a horizontal architecture, the DSS and the AVM compute in loop and the operator is the censor of this information process. From our point of view, the choice of an "autonomous" architecture is premature in the urban network management domain. The information quality and the low formalization of the domain's knowledge are difficulties that are hard to solve. Indeed, in urban environment the information is not sure because their collect is not an easy process. That is the reason why, a regulator spends an important part of his time-work in order to confirm the information coming from the AVM as soon as he has doubts on the correctness of the data. The generalization of the localization technologies like Global Positioning System (GPS) will solve this issue. Nevertheless this technical improvement can not easily replace qualitative information coming from drivers. For example, the management of a disturbance related to a misplaced car is not the same that if the reason of the disturbance is a person accident. An automatic system will detect that the vehicle is late but it will not be easy to know the reason of this delay.

In a vertical architecture, the DSS is a guide or a server of solutions. The AVM that has been created to facilitate the access to theoretical and real data is a basic guide but it has been shown that this role is not enough to help the regulator. A DSS as a solution server may be based on a simulation tool or it may have access to the information in the AVM. The regulator uses it as a simulation tool according to the *what-if* model. For example, Brezillon ([3]) proposes a DSS to the management of the parisian subway which is activated by the regulator in the case of a disturbance. This DSS proposes solutions according to information provided by the regulator. That means that the regulator filters the data and solves the quality data problem. With this organization of information processing the problem is the difficulty of managing disturbances in real time in a dynamic environment because the regulator has a new task, the information filtering.

In order to solve the problems related to the information management, a solution would be to link the DSS with the data.

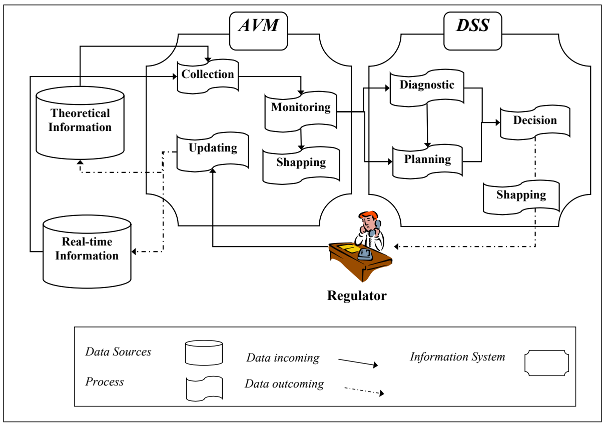


Figure 2: Integration of a DSS and AVM systems

Figure 2 shows an architecture that is a response to the data management problem. The AVM preserves its initial function: the collecting and shaping of data and the DSS access to useful data for its reasoning. The DSS proposes solutions to regulators which evaluate their validity because they access to the two systems. This architecture is based on a vertical organization of the cooperation between operators and systems. The main advantage is to reuse the existing system but it implies implicit information duplication. For example the position of the buses is used by the AVM for the interface and the alarm computation but it is also used by the DSS in order to compute a solution. It will be difficult to propose a modeling of the data qualification. For example, how could the DSS detect that the delay of a bus is due to a problem or to an error in the position collect (a captor is breakdown) without a comparison with the precedent position of the bus or the precedent records of the captor. The last issue is organizational, some AVM already have some of the functionalities that are related to the DSS like the computation of alarms.

The criticism on the positioning of the DSS, AVM and regulator leads to the proposition of a new generation of information system that we call: Transportation Regulation Support System (TRSS). Our proposition includes the functionalities of the AVM system and is based on the multi-agent approach in order to organize data and processing (Figure 3).

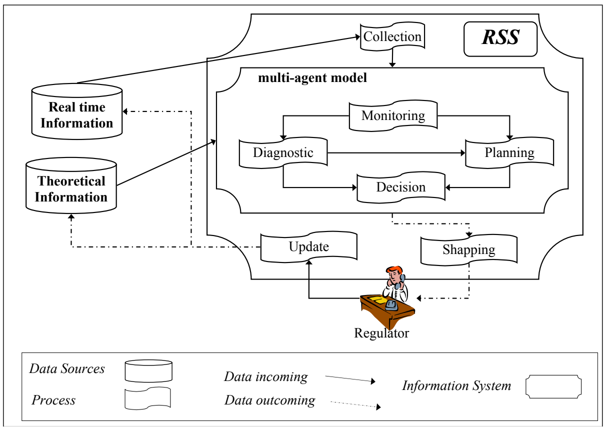


Figure 3: A Transportation Regulation Support System Architecture

The gathering of a DSS and an AVM system solves the problem of the duplication of the data. The same data are used to manage the transportation network under normal operating conditions (monitoring function) and also under disturbed conditions (diagnostic and planning functions). In order to avoid a superposition of complex functions, our proposition is based on the modeling of the dynamic of the data processing by a multi-agent reification of the static hierarchical data model. A multi-agent system has been defined to manage the transportation network under normal conditions (network monitoring, dynamic schedule management, data inconsistencies management) as well as under disturbed conditions ([1]). Next section justifies our choice of the multi-agent paradigm.

3.2 Choice of a multi-agent modeling approach

Several researches in the multi-agent community ([12, 5]) have been done in the transportation domain. In ([5]), the authors underline that 63% of the research are related to the conception of DSS. Mathematical system and Interactive Decision Support System have been important in the modeling of the decision process in the transportation domain but they are often black boxes that hide the decision process. Moreover these systems give synthetic results that have to be analyzed by the regulator in order to become a final diagnostic. These models compute with numerical data and it is not easy for them to take into account qualitative data, like the relative importance of a delay according to the position of the bus in the network. Moreover these models are not suitable to compute with uncertain and incomplete data. Most of all, these models presume that the data is available and reliable but as written before this is a strong hypothesis in the urban transportation network management.

A MAS has a different approach for the conception of a DSS. The first difference is related to the objective of these systems that is the comprehension of the process that they manage. A Multi-agent system makes easier the comprehension of a complex reality, by the reification of the components of the system to manage. This underlining of the components and of their links facilitates the comprehension of the regulation process that is at the beginning of its formalization. The multi-agent approach paradigm is well adapted to the transportation domain since it facilitates an approach by analogy in a domain where the objective is the management of distributed entities. The second difference is related to the management of quantitative, qualitative and symbolic description that the multi-agent paradigm facilitates. This point is useful to put into perspective the importance of the alarms.

In the domain of public transportation management, the works of ([10, 4]) are based on the use of a simulator. Ossowski *et al.* introduce an organizational and communicative model of decision support environments applied to transportation management. These systems are not integrated and are not directly fed with real-time data coming from vehicle sensors and in that sense are closer to the architecture shown in Figure 2. At last [14] proposes a multi-agent architecture for system related to the management of traffic. This work is mainly related to the issue of the management of entities that are physically distributed rather than to the transportation problem.

3.3 Choice of the environment modeling approach

Recently, the Environment for Multi Agent Systems technical forum group (E4MAS) has produced valuable outputs concerning the poorly exploited concept of the environment, notably in terms of roles, responsibilities, architecture, as well as practical applications, thus opening many challenges in terms of modeling, methodology and engineering [16]. In [16], the authors enumerate the responsibilities of the environment and some of them have already be applied in the transportation domain.

Because the environment is a shared space for the agents, resources and services, its first responsibility is the structuring of the MAS. The environment modeling is a solution to give a space-time referential to a transportation application. Moreover, its privileged intermediary role makes the environment a good candidate to support spatially and temporally decoupled coordination models and thus it simplifies the design of solutions taking into account dynamic real environment. In [15], the environment contains fields that are propagated in the environment in a certain range and used by the agents to organize the task assignment for Automatic Guided Vehicles. The MAS environment becomes the common referential that enables agents to adapt their behavior according to the dynamic of the real environment.

Because the environment has its own process, its second responsibility is to maintain its own dynamic. According to its structuring responsibility, the environment can also manage the dynamic of the transportation environment, ensuring the coherence of the MAS. In [15], the environment ensures the propagation of the fields. Moreover, the environment can ensure services that are not at the agent level or in order to simplify the agent design. In a traffic light control system [2] based on evolutionary game theory, the environment, that has a global point of view, gives rewards or penalties to self-interested agents according to their local decision.

Because the environment with its own dynamic can control the shared space, its third responsibility is to define rules for the multi-agent system. The environment can controls the execution of the MAS ensuring for example that the coordination rules will be respected. In a bus network simulation [9], the main role of environment is to constraint perceptions and interactions of agents. Indeed, a Bus agent and a Traveler agent can interact only when they are located at the same bus stop. For transportation applications that have an incomplete knowledge, it simplifies the design of the MAS by a clear separation between the roles of the agents and their organization. In the coordinated monitoring of traffic jams application [7], the environment provides organizations, which it dynamically evolves according to the current context. Organizational evolution uses a set of laws, which define the way organizations and role (played by agents) should evolve given the current context.

Because the agents are "users" of the services of the environment and to really create a common knowledge, the last responsibility of the environment is to let observable and accessible its own structure. From our point of view, this last responsibility simplifies the reification of the MAS components in the environment and their manipulation by the agents.

4. A TRSS APPLICATION: THE SATIR SYSTEM

4.1 Multi-agent modeling of a transportation network

The agent modeling takes into account our initial proposition that is to gather in the same unit the data and the knowledge useful to its process. For instance the agents not only use the position of the buses but they also detect the inconsistencies. That is the reason why cognitive agents have been chosen. The static modeling of the data is based on a hierarchical organization. The stop is the elementary component of the physical and timetable information; the vehicle is the access point to the real-time information. We also define two types of agents: 1) The STOP agents that represent the theoretical structure of the network and compute the theoretical timetable; 2) The BUS agents that represent the dynamic part of the network. Every BUS agent is the abstract model of an actual vehicle running on the transportation network and reports its movements to the STOP agents.

The interaction between agents in the coordination protocols is based on the exchange of messages. In the domain of urban traffic control, the sender does not always know the name of its receivers because the receiver of the message is often identified according to its position. For example, when a bus has to contact its nearest bus, it does not know its identification. Usually, the simplest protocol is a broadcast protocol (more or less limited). The drawback of this solution is the high communication cost, mainly in real-time systems like urban transportation systems since the location of buses is updated very frequently (every 40 seconds in our application). Another simple solution is the use of acquaintances; the interaction problem is solved by an increase in the interaction knowledge of agents. In a TRSS, this solution is inadequate because the problem remains when an agent is not able to link its needs to an agent identifier. A third solution is the use of a middle-agent. This approach called, "capability-based coordination", is a preference/capability matching, used to identify the best provider for a given capability search. In our transportation problem this solution has no sense because all STOP agents have the same capability (idem for BUS agents). Because the dyadic interaction solutions are not adapted to the transportation domain, we propose to base our interaction model on the mutual awareness principle. An important part of the interaction in real-life situations comes from other means than direct transmissions [6], and is related to a particular state of the participants: awareness. Although it has long been considered as a passive state, we consider that awareness is an active state and not only the result of stimuli. Work in the fields of psychology and sociology have discussed whether or not there also has to be an active participation of the "perceiver". For example, Heath [8] says that awareness is not only the perceiver's availability to be aware of the environment, but also his ability to "filter relevant information which is of particular significance". Mutual awareness is based on the sharing of interactions. To be efficient, this principle implies that agents share a common communication media. As a consequence, an agent has to find only messages that it is interested in. In the reactive agent community, the environment is already used as a common interaction medium. In the cognitive agent community, we have proposed the EASI model (Environment as Active Support of Interaction) [11], which enables cognitive agents to use the environment to exchange messages. More precisely, it enables an agent to send messages to another agent that is located by the environment, and also enables agents to perceive every message exchanged.

For this purpose, we consider that the environment contains descriptions of messages and agents. The problem is how the agents use these descriptions to locate messages according to the environment state. This implies matching those descriptions and the needs of the agents. We therefore propose to represent all the components of the environment (agents and messages) as entities. Each entity has a Public Layer containing the properties, accessible through the environment. Agents have the ability to put filters in the environment and these filters are logical expressions on properties. When a message is added to the environment, these filters determine by pattern matching whether the agent is interested in it, in which case it will receive it. In this way, the filters enable agents to create their communication space where each filter corresponds to a precise communication need:

- Reception filters: communication is based on a need that is common to two agents. The sender specifies the values of the characteristics searched for in the receiver. This description is matched against a communication filter of the category of the agents contacted.

- Emission filters: communication is based on a need of the sender which doesn't match any expectation of the category of contacted agents. The consequence of this lack of interest among the receivers is the absence of suitable filters which would make it possible for the environment to find the agents concerned. Consequently, the sender must put the appropriate filter in the environment at the same time as it sends its message. An emission filter may require the comparison of the potential receivers and therefore requires the use of predicates. This implies that the filter matches all the potential receivers of a message and then searches for the exact receiver of that message.

- Interception filters: the interception filters are the sensors which allow the agents to receive the messages which are not sent to them but the content of which may be of interest for them. The goal of these filters is to make full use of the environment as a shared work context where every communication is potentially available for all of the participants.

The organization modeling has to take into account the hierarchical organization of the information and the treatment of this information. The hierarchical organization corresponds to the view of the agents according to different abstraction levels. A STOP agent is located on a line and a route. The gathering of the agents according to a line or a route gives the needed hierarchical level. The monitoring of the network under normal condition is based on this organization. In order to isolate in this data set the information related to a disturbance, another organization will gather the agents involved.

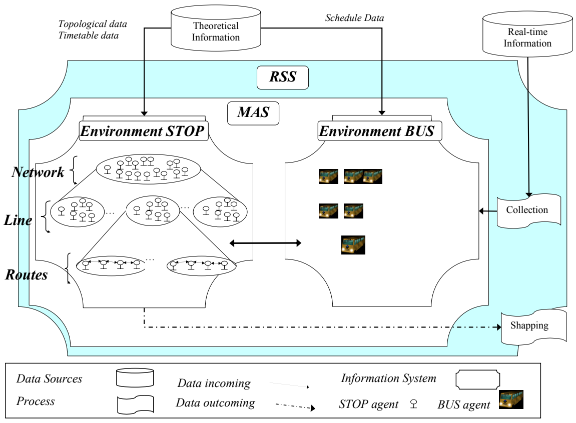


Figure 4: A Transportation Regulation Support System application: the SATIR system

Figure 4 shows the proposed multi-agent model. The agents are gathered in separate environment according to their functionalities. In each of these environment, the communication needs are not the same and the set of communication filters is therefore adapted. The communication is allowed inside and between environments.

At the beginning of the process, the theoretical data are distributed between the STOP agents. The STOP agents have the topological information on their network position: the line, the route, the distance to the next STOP agent (with the identification of this agent) and more particular information like the physical possibility to do a U-turn. A data table is associated with each stop of the network for each period of time (classified from 0 (fluid) to 2 (heavy traffic)). These data are used to define the theoretical state of traffic and of passenger demand and to compute schedules. For instance, at the stop p1 the estimated value of traffic is 1 (the circulation is normal) and the estimated value of flow of passengers is 2 at 8 pm (p1 may be a school). The dynamic schedule computing replaces the use of a theoretical timetable that not remains a valid reference when the network is disturbed (see 2.2). The STOP agents have the knowledge (traffic problems and passenger flow) used by the graph makers (staff that compute the theoretical supply) in order to establish a theoretical timetable. This knowledge is also used in the assessment process and in the search for solutions to a disturbance.

When a vehicle passes a stop on the real network, a warning message is sent from the BUS agent to the involved STOP agent. The STOP agent updates its timetable by removing this vehicle from its list of expected vehicles. A STOP agent which does not receive any message detects an anomaly and triggers the disturbance processing presented in paragraph 4.2.

In order to only process the important disturbance, the trigger of the disturbance management process depends on the value of a time parameter that may be different for each STOP agent. A default value of 7min has been used (STIB norm). When a vehicle is no longer located, the STOP agents on the bus route have not been informed about the passage of the bus, but they will intercept all new transit announcements sent by vehicles not running to timetable. The interceptor agents receive the message and update their timetable. Overhearing provides an efficient solution to the problem of the inconsistent data, in that it needs few information: the BUS agents have no information on the topology of the routes; and only one warning message is sent for a transit event in the network and the MAS state is updated according to the reaction of agents to this event in their own context.

The BUS agents are the intermediates between theoretical data (STOP agents) and real-time data. This repartition of roles ensures the modularity of our proposition. For example, a modification in the data collects implies a modification of the BUS agents, the rest of the MAS remaining the same. Moreover, this decomposition facilitates the development of more evolved functionalities as diagnostic or planning.

4.2 Multi-agent modeling of the diagnostic process

The STOP agents have knowledge about the theoretical structure of the network and the BUS agents have knowledge about the actual activity of the network and also the theoretical activity of the vehicles (each BUS agent manages its own timetable). We have proposed to put together within a specific organization called the Incident model ([1]) all the STOP agents and BUS agents which are related to a given disturbance. For this purpose, we define three information sets, also called areas:

- *The Successor area*: This area brings together all the stops waiting for the successor of the late bus, it measures the risk assessment of a bus train (the late vehicle is caught up by the following one).

- *The Critical area*: This area brings together all the stops where the vehicle is late, it measures the risk assessment of a gap (the late vehicle is left behind by the preceding bus).

- *The Predecessor area*: This area brings together all the stops where the late vehicle is due but not yet late, it measures the risk assessment of a gap.

The set of these three areas constitutes the Incident model. The Incident model gathers in a single entity all the information necessary to manage a given disturbance thus helping the regulator to do its diagnosis job. The figure 5 pictures three disturbances (c) with their respective areas. For each disturbance, a specific environment is created (a) where the organization of the agents is hierarchic as explain below (b).

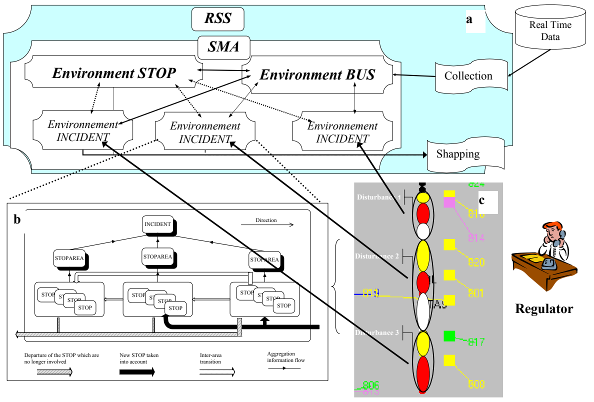


Figure 5: Multi-agent dynamic disturbance modeling in a TRSS

To measure qualitatively the importance of a delay, we have taken into account its consequences on the activity of the network. We have defined two measures of risk linked to a disturbance that are detailed in ([1]). These measures are based on the study of a priori progression difficulties of vehicles involved with the disturbance and take into account the intrinsic dynamics of a disturbance.

The initial organization of the multi-agent system (in lines and routes) is completed with a hierarchical organization of the agents in order to aggregate information and to compute feasible solutions (Figure 5 (b)). At each level of the hierarchy, information is aggregated by the agents. The two new types of agents are the STOPAREA agents and the INCIDENT agent. The lowest level of the hierarchy is composed of the elementary entities, the STOP agents. The middle level is composed of the STOPAREA agents that make an initial summary of the information. They collect basic information such as theoretical traffic evaluation and passenger flow from the STOP agents linked to them and they compute the progression coefficient (an indicator of the problem of the area). The INCIDENT agent represents the top of the hierarchy where feasible solutions are computed and is the link between regulators and the system.

This organization is dynamic because at each cycle, STOP agents move from one area to the other within the hierarchy, and from and towards the outside of the organization, according to traffic direction (Figure 5).

When the disturbance disappears, this organization survives during some cycles to keep the continuity of the disturbance process. After this period, the created agents related to the disturbances disappeared too.

The originality of our approach is the dynamic modeling of a disturbance process from its beginning to its end and its integration in a multi-agent system. We have defined a model, called the Incident model that allows information synthesis that is useful for decision making. Through this model, knowledge relative to the network structure and knowledge relative to the network dynamics (stored in STOP agents and in BUS agents, respectively) are gathered within a single entity that is reified by a specific environment (one environment by disturbance) where all the involved agents are brought together. This entity allows the follow-up of the disturbance over space and time; it is deleted when the disturbance is solved.

4.3 Multi-agent modeling of the planning process

When a disturbance has been detected and assessed, our TDSS computes the feasible regulation procedures. Initially, the transport service matches the theoretical demand to the bus supply. However, when a disturbance appears, there is a discrepancy between the service provided and the passenger flow. Thus, the task of the system is to adjust the initial supply in order to satisfy the needs according to the changes to the network.

4.3.1 Static feasible action planning

Thanks to predefined procedures, regulators modify the transport service according to the state of the network and to the possible actions of the buses on the line. They cancel or modify the vehicle timetable in order to shift the service to another point on the network. One of the original features of SATIR is that BUS agents play the role of regulators, enabling a micro-regulation of the network. At the beginning of its activity, each BUS agent receives the list of the runs it is supposed to do (timetable) and that it may modify dynamically, thus acting as a regulator. When a disturbance is detected, the late BUS agent requests a new run that each BUS agent on the same line tries to insert into its timetable. For each BUS agent, this insertion implies a modification of its timetable. And one or more regulation procedures can be used. Timetable processing within the multi-agent system is implemented in three steps.

- step 1, the availability of the BUS agents is computed in order to eliminate vehicles that are not potential solutions. Using conditions related to its own characteristics (example: its size) or related to network rules (example: the last run of a vehicle is never changed), a vehicle is eliminated.

- step 2, the profiles of the available BUS agents are computed. We call profile the characteristics of a group of BUS agents with the same relative position compared to the late BUS agent (i.e. before, after, same direction,etc.). For each profile, the regulation procedures are the same. For example, the AlightingOnly procedure may be feasible for all BUS agents located before the late BUS agent but it is useless for the BUS agents located after it. Using the BUS profile may limit the number of tests: some procedures may be forbidden or limited to specific profiles.

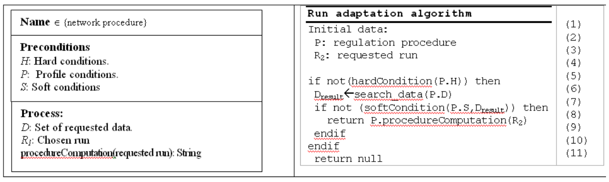
- step 3, the feasibility of the regulation procedures is computed. Every regulation procedure has constraints that BUS agents must satisfy in order to be considered as feasible. For example, a vehicle cannot make a U-turn if there is no location to do so.

Breaking this timetable processing down into three steps offers several advantages. Since network authorities have their own regulation rules, they do not apply the same constraints on vehicles and on regulation procedures. The three steps described above enable a network to adapt the planning process to its own rules and constraints. Moreover, the distribution of BUS agents into profiles limits the solution space to the only feasible procedures. from a micro-regulation viewpoint, another advantage of this planning process is that it can be distributed and automatically applied by BUS agents.

4.3.2 Dynamic feasible action planning

The adaptation process begins with an inform message sent by an INCIDENT agent to the late BUS agent. Because computation of the regulation procedures depends on its position and on the information related to its current run, the late BUS agent looks for the missing information on the network. Firstly, it sends a message to the STOP agents located between its own position and the end of its current run to collect the missing data, such as passenger flow and run length; secondly it forwards it to the BUS agents that are on the same line by sending a request message. The next step is done by the BUS agents that apply the steps of the planning process described above. Thanks to its local knowledge of the network and of its own activity, a BUS agent may propose feasible regulation procedures. We propose a general model of the regulation procedure as follows (Figure 6).

For each regulation procedure we define three preconditions and a computation process. The preconditions are related to the steps of the planning process and the computation process computes the insertion of a new run requested by the late BUS agent into the BUS agent timetable. Let H be the set of hard preconditions related to the characteristics of the BUS agent (see step 1). Each condition takes into account the internal state of the BUS agent and does not require any additional information to be evaluated. Let P be the conditions related to the profile of the BUS agent (see step 2). The BUS agent computes its own profile given the current disturbance and computes the regulation procedure that is linked to its profile. Let S be the soft conditions that are related to the availability of the procedure (see step 3). If we take the example of the AlightingOnly procedure it is limited to the profiles of BUS agents that are located at the beginning of the line and before the late vehicle. The soft preconditions, defined by the STIB Belgium bus network, are the following (H = ∅ which means that AlightingOnly is always possible):



Name ∈ {network procedure}

Preconditions
H: Hard conditions.
P: Profile conditions.
S: Soft conditions

Process:
D: Set of requested data.
R_1: Chosen run
procedureComputation(requested run): String

```
Run adaptation algorithm
Initial data:
 P: regulation procedure
 R_2: requested run

if not(hardCondition(P.H)) then
 D_result ← search_data(P.D)
 if not (softCondition(P.S,D_result)) then
   return P.procedureComputation(R_2)
 endif
endif
 return null
```

Figure 6: The regulation algorithm

- the distance (number of stops) between the BUS agent and the late BUS agent is less than 5. This procedure means that the passengers of the late vehicle have to wait for the following vehicle and that the waiting time must not be too long.
- the position of the terminus of the BUS agent involved must be superior to the position of the late vehicle terminus. If this is not the case, the procedure will be done only on part of the run of the late vehicle.
- the distance between the late vehicle and its predecessor is greater than 10 stops. Within this procedure the late vehicle will make up lost time. The aim of this precondition is to avoid the creation of a bus train with the previous bus.

In order to compute these conditions, the BUS agent has to look for the missing information. The vehicles charge is not taken into account because the STIB network does not have captors to give this information. For the AlightingOnly procedure, it has to know the position of the predecessor of the late vehicle. The name of this data and that requested for the computation process are recorded in the data set called D (figure 6).

Let R1 be the run that will be modified to insert the requested run. R1 is the current run or a run that is chosen according to its departure time. For the AlightingOnly procedure, the chosen run is the current run of the BUS agent and the adaptation consist of the computation of a new timetable. To compute this timetable, the BUS agent looks for the following data: number of stops, traffic and passenger flow values between the vehicle and the late one. In Figure 6, the algorithm gives the steps of the new run computation for a BUS agent. Only the procedures that belong to the profile of the BUS agent are taken into account. If the hard conditions are false (hardCondition line 5), the BUS agent looks for the missing information (search_Data line 6).

If the soft conditions are true, then the function procedureValidation returns the result of the computation. Using the data on the chosen run R1, the requested run R2 and the collected data D_{result}, this function computes the run-time of the regulation procedure.

To close this process, each BUS agent sends the result of its computation to the INCIDENT agent that gathers and organizes this information for the regulator. In the next section, the result of this process is presented.

5. EXPERIMENTATION AND RESULTS

A prototype has been implemented in C++. In order to study the feasibility of our SATIR system, the prototype was tested using real data recorded every 40 seconds from buses on the Brussels Intercity Transport Company network (STIB). In this section we give some result about the planing process. The result about the assessment process are given in [1]. The data was recorded on tape for around 30 buses, on one line, over 8 days and represents more than 43,000 items. The SATIR system was run over time through cycles on this data representing the movement of buses on the network; it detected 300 incidents and it recorded the disturbances data on file. For each day, the run time is only a few minutes.

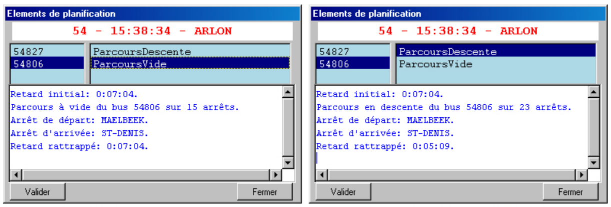
In the following example, line number 54 has been studied as follows: 1) Day 1: the monitoring data of the regulator that manages line 54 was recorded. Each of the managed disturbances and the chosen solutions were recorded. 2) Day 2: SATIR was tested with the data related to the line activity of day 1. A serious disturbance managed by both the regulator and SATIR was identified. 3) Day 3: SATIR was shown to the regulators and the solutions to the same disturbance were compared.

The disturbance is the following: a badly parked vehicle blocked bus #54806 that ran more and more behind schedule. The regulator was informed by the driver at 15:33. The regulator chose to call the vehicles near this problem in order to organize a diversion, but there is a technical problem and the driver was unable to get the call. The consequence was a bus train at 15:56:14.

Since the disturbance was detected by the regulator after a call, SATIR did not have this information. As a consequence the disturbance was detected by SATIR after 7 min (which avoids a false alarm), at 15:38. The vehicle was at the stop called ARLON and its STOP agent triggered the disturbance assessment process (Figures 7). Every 3min, the new INCIDENT agent updated the assessment. This disturbance was close to line number 80. The regulator used this proximity to choose a vehicle that was at the end of its runs to substitute for the blocking bus. This new vehicle does not exist in the AVM system and a fictitious reference was created for it. Since vehicle #54806 had a substitute it made a U-turn to make up for lost time. The regulator planned the U-turn where the substitute bus was inserted (called vehicle #1). When the blocking car had gone, vehicle #54806 continued its runs to the stop where vehicle #1 was waiting. At this location, vehicle #54806 made a U-turn and the vehicle continued the run of vehicle #54806. SATIR was not able to find this regulator's solution because this solution implied an external resource (i.e. a new bus). Bearing this in mind, the SATIR planning process detected two BUS agents that could provide a feasible solution, vehicles #54806 and #54827.

Vehicle #54806 has two proposals that take into account the next vehicle (#54830). The common objective is to increase the speed of vehicle #54806 by modifying the run mode. The first feasible procedure is an empty run (M = Without-passengers, Figure 7.A); the second choice is a run with alighting only (M = AlightingOnly, Figure 15.B). These procedures take into account the next vehicle because they are feasible if this vehicle is close enough to pick up the passengers that are not picked up by vehicle #54806. All this information is displayed to the regulators through the interface (Figure 7 (A and B)). For each procedure, several items of data are given: the current delay (7 mn and 4s), that part of the delay that can be made up, how many stop are necessary to make it up. For instance, in the first procedure (M = Without-passengers), 15 stops are needed and in the second one (M = AlightingOnly), 23 stops are needed.

Each regulator proposes solutions according to his own knowledge, habits and experience. For the situation described in this paper, the regulator may propose several solutions that are different to those proposed by SATIR. In our example, the regulator has chosen external resources although there are internal solutions. In order to validate our model, the SATIR proposals were studied by the regulators, who approved them as being feasible. If they did not choose the AlightingOnly procedure chosen by SATIR, this is because it may be efficient for certain lines but not for



```
Retard initial: 0:07:04.
Parcours à vide du bus 54806 sur 15 arrêts.
Arrêt de départ: MAELBEEK.
Arrêt d'arrivée: ST-DENIS.
Retard rattrappé: 0:07:04.
```

```
Retard initial: 0:07:04.
Parcours en descente du bus 54806 sur 23 arrêts.
Arrêt de départ: MAELBEEK.
Arrêt d'arrivée: ST-DENIS.
Retard rattrappé: 0:05:09.
```

A) Empty run

B) Alighting only run

Figure 7: Two SATIR proposals with vehicle #54806 to made up a delay

others (remember that SATIR was tested on only one line and more testing has to be done).

6. CONCLUSION

In this paper, we have presented a Transportation Regulation Support System that represents a global approach to the regulation task on a transportation network. Our proposition is based on a multi-agent modeling of the transportation network. The MAS organization enables to reproduce the functionalities of the existing information system and thus manages the transportation network under normal operating conditions (where are the buses located ?). Under disturbed conditions (where are disturbances (bus delays, bus advances) located ? What action has to be taken to solve the problem?), this organization is completed by a new organization of the agents concerned by a disturbance. The experience gained from the development effort of our system supports the proposition that the multi-agent paradigm is an appropriate framework for transportation network modeling and simulating.

7. REFERENCES

[1] Flavien Balbo and Suzanne Pinson. Dynamic modeling of a disturbance in a multi-agent system for traffic regulation. *International Journal of Decision Support System*, 41 (1):131–146, 2005.

[2] Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multiagent Systems*, 10(1):131–164, March 2005.

[3] P. Brezillon, C. Gentile, I. Saker, and M. Secron. Sart : a system for supporting operators with contextual knowledge. In *Proceedings of the Conference on Modeling and Using Context*, Brasil, 1997.

[4] Zhao J. Bukkapatnam S., Dessouky M. distributed architecture for real-time coordination in transit networks. Technical Report metrans project 00-13, 2003.

[5] Paul Davidsson, Larry Henesey, Linda Ramstedt, and all. An analysis of agent-based approaches to transport logistics. In *Transportation Research part C: emerging technologies*, volume 13(4), pages 255–271. Elsevier, 2005.

[6] J. Dugdale, J. Pavard, and B. Soubie. A pragmatic development of a computer simulation of an emergency call center. In *Designing Cooperative Systems: The Use of Theories and Models*, pages 241–256. IOS Press, 2000.

[7] Robrecht Haesevoets, Bar Van Eylen, Danny Weyns, and all. Context-driven dynamic organizations applied to coordinated monitoring of traffic jams. In Danny Weyns, Sven Brueckner, and Yves Demazeau, editors, *Proceedings of Workshop Engineering Environment-Mediated Multiagent Systems*, pages 126–143, 2007.

[8] Christian Heath, Marcus Sanchez Svensson, Jon Hindmarsh, Paul Luff, and Dirk vom Lehn. Configuring awareness. *Comput. Supported Coop. Work*, 11(3):317–347, 2002.

[9] David Meignan, Olivier Simonin, and Abderrafiaa Koukam. Adaptive traffic control with reinforcement learning. In *4th Workshop on Agents in Traffic and Transportation (ATT)*, pages 50–56, 2006.

[10] Sascha Ossowski, Josefa Z. Hernández, and all. Decision support for traffic management based on organisational and communicative multiagent abstractions. In *Transportation Research part C: emerging technologies*, volume 13(4), pages 272–298. Elsevier, 2005.

[11] Julien Saunier and Flavien Balbo. Regulated multi-party communications and context awareness through the environment. *International Journal on Multi-Agent and Grid Systems*, 0(0), 2008. to appear.

[12] R. Schleiffer. Intelligent agents in traffic and transportation. In *Transportation Research part C: emerging technologies (special issue)*, volume 10C 5-6, pages 325–527. Elsevier, 2002.

[13] R. Vahidov and B. Fazlollahi. Pluralistic multi-agent decision support system: a framework and an empirical test. *Information and management*, 41:883–898, 2004.

[14] Fei-Yue Wang. Agent-based control for networked traffic management systems. *IEEE Intelligent Systems*, 20(5):92–96, 2005.

[15] Danny Weyns, Nelis Boucké, and Tom Holvoet. Gradient field-based task assignment in an agv transportation system. In *AAMAS '06: Autonomous agents and multiagent systems*, pages 842–849, New York, NY, USA, 2006. ACM.

[16] Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for multiagent systems, state-of-the-art and research challenges. *Lecture Notes in Computer Science Series*, 3374:2–52, 2005.