A Transportation Decision Support System in Agent-Based Environment

Flavien Balbo1,2 Suzanne Pinson1

1 Lamsade, Paris Dauphine University.
Place Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France.
balbo@lamsade.dauphine.fr
pinson@lamsade.dauphine.fr
2 Inrets - Gretia, Institut National de Recherche sur les Transports et leur Sécurité.
2, avenue du Général Malleret-Joinville, F-94114 Arcueil cedex.

Abstract

This paper presents a Transportation Decision Support System (TDSS) called SATIR that reports the network activity in real-time and thus assists the bus network regulators in their various tasks. We propose a global approach to help regulators to manage the bus network under normal conditions (network monitoring, dynamic schedule management), with data inconsistencies management and under disrupted conditions (disturbance diagnosis and action planning). This global bottom-up approach is based on a multi-agent model where agents encompass the domain knowledge and where the interactions between agents are based on our new model of the environment, the ESAC model (Environment as Active Support of Communication). A formal modeling of the disturbance concept, which makes it possible to capitalize the knowledge available within a monitoring station and to follow up the evolution of the disturbances in real time, is presented, followed by a description of its application to a problem of bus network diagnosis in the STIB network (Brussels Intercity Transportation Company network). The paper concludes by looking briefly at how using the multi-agent paradigm could help to develop new functionalities to improve Transportation Decision Support Systems design.

Keywords: Decision support system, multi-agent system, transportation system, disturbance modeling, bus network diagnosis, environment modeling, ESAC.

1 Introduction

Designing, implementing, and adjusting urban public traffic control systems require a good deal of effort and knowledge. It has been shown that a 2km/h increase in bus speed in all the provincial networks would represent a profit of 0.3 million euros for the French UTP (Union des Transports Publics). The effectiveness of urban traffic control systems depends greatly on their ability to react in real-time to changes in traffic patterns (for example traffic jams, roadwork, one-way streets, passenger clusters). The hypothesis is that it would be useful for human regulators (the staff in charge of monitoring the network) to design Transportation Decision Support Systems (TDSS) that can adjust if the environment changes, that is to say to automatically detect incoherent data, traffic disturbances and then automatically propose solutions to optimize the traffic flow in order to maximize the person and vehicle throughput and to minimize delay.

As urban environment density increases, the number of disturbances increases in bus networks, and management techniques classically used by regulators have become obsolete. Although adapted decision support systems have been created for train [25] or subway [6] networks, the bus regulators have no decision support system adapted to their needs. Regulators use systems known as Automatic Vehicle Monitoring systems (AVM) ([20][27]), which were developed in order to better ensure the success of the transportation plan. An AVM allows the management of 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 the vehicles are running ahead of timetables or are running late.

The use of an AVM is the first step to the computerization of the transportation network activity. However, this system is limited to coping with disturbances linked to unanticipated demands and to traffic conditions but is not able to manage all the difficulties arising from the management of a bus network in real-time: managing the inconsistencies of the data from the sensors which locate the vehicles, assessing a disturbance according to its context, proposing feasible solutions, etc. These limits are due to the collecting and organizing of data that are insufficient to help regulators.
The aim of our decision support system is to propose an adapted and original answer to the real-time problem of bus network management. Real-time transportation systems are characterized\(^1\) by [14][18]:

- Incoming data arriving very frequently representing network changes,
- Dynamically modifying the theoretical supply in order to satisfy the needs according to the changes on the network,
- Distribution of data coming from vehicle sensors.

Agent-based DSS are particularly relevant in domains where human operators have to make operational decisions regarding the management of complex organizational processes [32], which are inherently distributed (spatially, logically and/or physically). We propose to use the multi-agent paradigm to model our real-time DSS. The autonomy of a multi-agent system where autonomous entities, called agents, interact with each other and its ability to adapt and react to the changes in the environment are essential in the field of transportation where the environment is dynamic, open and uncertain [34]. Some research in artificial intelligence [28] and in the multi-agent community [31] [11] has been done in the transportation domain. Most of it is limited to a particular problem: the connection problem [8] [22], real-time timetable management [23], commercial speed improvement [21]. The MAS approach has also been used for traffic simulation [5] [15], in order to improve the cooperation between information systems [19] [17] [33] or as a Decision Support System for traffic congestion management [26].

These systems present several drawbacks: they are mostly simulation systems [29] [9] that are not integrated into the transportation Decision Support System, and they are not directly fed with real-time data coming from vehicle sensors.

This paper is organized as follow: the second section introduced notions of the urban transportation domain. In the third section, a global approach for the design of a Transportation DSS is proposed. The fourth section describes the daily network management process through our multi-agent model. In the fifth section, the disturbance management process is detailed through the integration of a formal model of a disturbance into the multi-agent system. The sixth section shows how the system operates and presents the implementation of the system and its validation on data coming from the Brussels bus transportation network (STIB). This article ends with a summary of the lessons learned and suggestion for future work.

2 A global approach to network management

2.1 Notions of the domain

In urban transportation control, human regulators are located in a control center. They have to manage the transportation network under normal operating conditions (where are the buses located?) and also under disturbed conditions (where are disturbances - bus delays, bus advances- located?) and what action has to be taken to solve the problem?

Different studies of the regulators’ work usually identify three tasks as regulators’ activities [10]:

- **network monitoring**: they collect and aggregate relevant information and analyze what it means,
- **problem diagnosis**: they express disturbances in terms of causal features and analyze the significance of a disturbance according to its context,
- **action planning**: they compute solutions to the problem.

In most networks, vehicles are located through sensors which provide real-time information. This information represents a huge amount of data (for example in the Belgium STIB bus network, data arrives every 40 seconds). Furthermore it may be incomplete (a sensor breaks down) or uncertain (the quality of the data is sometimes poor). This data is collected through the Automatic Vehicle Monitoring systems (AVM). AVM compares the actual positions of the vehicles (captured by the sensors) with their theoretical positions given by pre-registered timetables in order to provide the regulator with an overview of the routes. In this way the regulator can see whether the vehicles are running ahead of schedule or are running late.

Figure 1 shows the AVM management of real-time information coming from sensors and the output of the system. Each line is represented two ways with its stops and its running buses. Each bus location is represented

\(^1\) http://www.its-jp.org/english/arch_e/main.htm
by 1) a number for its theoretical position coming from the theoretical timetable, 2) a colored square for its real location detected by the system. This real location may be erroneous due to sensor breakdowns. Stops are represented by black dots. The gap between the theoretical position and the real position gives information about the bus delays or advances. Colors give the seriousness of the delays or advances.

![Diagram of bus transportation network](image)

Figure 1: Real-time information in a bus transportation network

By comparing the theoretical information with the real information, the AVM system tries to adjust the supply to the real operating conditions. At diagnosis level, it provides a number of primary alarms: advance/delay running, non-ensured next departure, bus train, problems with connections. At control level, some automatic procedures based on the departure control or on the interval control can be applied. Lastly, the AVM system also ensures the statistical assessment of the network operation (service journey run time, passenger demand) enabling in return better transport supply planning.

However, these procedures are unable to assess the seriousness of a disturbance according to its context or to propose feasible solutions. These procedures are also unable to detect false alarms: when a vehicle is not announced to stops it has served, it is declared late whereas it passed by the stop thus generating false alarms and erroneous output on the screen. More precisely, the limitations of the AVM disturbance diagnosis process are the following:

- **Lack of a global vision**: The splitting up of monitoring by line and the high number of lines to be monitored (for example in the Belgium STIB network each regulator tracks 13 lines with 5 to 20 buses running during the day) prevent global management of the network.

  Each regulator has to monitor more lines than he can materially see on the screen. The regulator’s first activity is to choose which lines to observe (in the transportation network under study, each regulator can handle 3 lines out of 13). This complex process involves various information sources (the AVM system for real-time information, theoretical timetables, information coming from drivers or from other regulators, etc.) [10]. An experienced 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. Since this kind of information is less available to novice regulators, they are less efficient in solving disturbance problems and they very much need the help of a DSS.

- **Lack of space-time dimension**: Primary alarms on the advance/delay of each bus provide too instantaneous a picture of line conditions. The disturbance context is not taken into account. As soon as the regulator chooses a disturbance on his screen for processing, he has to complete his knowledge of the problem. This process is complex because disturbances evolve independently along three axes [30]:

  - **Time axis**: this axis measures the severity of a disturbance according to the timetable. For example, a stop near a university should be monitored with timetables corresponding to arrivals and departures of students.

  - **Space axis**: this axis measures the severity of a disturbance with respect to its position on the network. For example, some locations are known to be critical and a disturbance at these locations is more difficult to manage. 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 such that users have to wait a long time for the next bus.

  - **Shape axis**: this axis measures the consequences a disturbance may have on the network activity. For example, a vehicle running for its last journey is less critical than if it has to operate a connection.

These three axes which are necessary to evaluate the severity of a disturbance underline the difficulty of regulation. The regulator has not only to establish a diagnosis on the current state problem, but also to consider its possible evolution along these three axes.
2.2 An overall description of our approach

Our global approach takes into account the three regulation tasks of human operators: network monitoring through information synthesis, disturbance diagnosis and action planning. Furthermore, it manages the transportation network under normal conditions (see section 4) (network monitoring, dynamic schedule management), with data inconsistencies management and under disrupted conditions (see section 5).

Figure 2: Our global approach to a Transportation Decision Support System

It also takes into account the data coming from the existing information system (AVM system) and handles the environment characteristics that are as follows:

- **the environment is open**: vehicles “appear” and “disappear” from the information system according to their activity or the regulator’s needs,

- **the environment is uncertain**: in most networks, vehicles are located through sensors that provide information that may be incomplete (a sensor may break down), or uncertain (the quality of the data may be poor),

- **the environment is dynamic**: in the STIB network, information related to the location of vehicles is collected every 40 seconds and represents a large data stream.

Based on this global bottom-up approach, we have developed a TDSS called SATIR (Système automatique de Traitement des Incidents en Réseau). It is built on a multi-agent model where agents encompass the domain knowledge and where the interactions between agents are based on our new model of the environment, the ESAC model: the Environment as Active Support of Communication [1][2] (see section 3.2). An original feature of SATIR is that the same multi-agent model is used to process data and to find solutions: 1) network monitoring is processed through dynamic timetable management; 2) disturbance diagnosis is based on an original model of disturbance taking into account the disturbance context ant its evolution, 3) feasible solutions are computed taking into account the context or profiles of vehicles.
3 The multi-agent model

The agent technology is based on the notion of reactive, autonomous, proactive entities that evolve in a dynamic environment. In open systems, agents may appear and disappear over time. In order to design a multi-agent system, several components have to be defined precisely: the agents, the interactions, the environment and the agent organization.

3.1 The agents

In order to model the three regulation tasks presented in section 2, we propose two categories of agents:

- STOP agents that represent the theoretical structure of the network (organized in lines and routes). They encompass the knowledge of the graph makers: passenger flows and traffic problems used to make up the theoretical timetables. A STOP knows the identifiers of STOP agents situated just before/after it on the same line and route. A database is associated with each STOP agent for each period of time giving the quality of traffic (from 0 for low traffic to 2 for heavy traffic) and of passenger flow (also coded from 0 to 2 – heavy flow).

- BUS agents that represent the dynamic part of the network. Each BUS agent is the abstract model of an actual vehicle running on the transportation network and reports its movements to the STOP agents.

These agents represent the network information system where the access points are the vehicles and the stops. For instance, the identification of a vehicle is used to access the driver timetables. Two more types of agents will be proposed for disturbance processing (see section 5.2).

3.2 A new interaction model using the Environment as Active Support of Communication

The main point of multi-agent systems is that the agents interact with each other in order to carry out their tasks and reach their goals. Thus, the performance of a Multi-Agent System (MAS) is determined by the ability of the agents to manage their contacts, i.e. how they can find the agents needed at minimum cost. When a message is sent, the sender can distinguish two types of receivers: the potentially right contacts, which may answer the sender’s needs, and the useless contacts –for this particular message–, which do not meet the requirements of the sender.

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 TDSS, 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 [12]. In our transportation problem this solution has no sense because STOP agents have the same capability (identifier 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 come from other means than direct transmissions [13], 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 [16] 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 those 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 ESAC model (Environment as Active Support of Communication) [2], 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. The ESAC model proposes to use the environment as an active and intelligent entity that can send the right information to the right agent at the right time.

- 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 via 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 the agents to create their communication space. Three types of filters have been defined in our model: emission filters, reception filters, interception filters. Each filter corresponds to a precise communication need of the sender and/or the receiver agents.

4 Network monitoring using the multi-agent model

4.1 Dynamic timetable management

Timetable management involves three steps: 1) making up the theoretical timetables; 2) monitoring the network activity (modifying the timetables according to where the vehicles actually are); 3) managing the inconsistencies of the data coming from the sensors that locate the vehicles.

In our Agent-Based DSS, STOP agents have the knowledge of graph makers (traffic problems and passenger flow) to compute theoretical timetables. This important feature allows the multi-agent system to be autonomous. Ten minutes before a vehicle leaves its station, each STOP agent computes its timetable using its knowledge and the identifier of the next STOP agent while taking into account the time of day. For example, a stop P1 knows, at time 8pm, the estimated value of traffic (equal to 1) and the estimated passenger flow (equal to 2). The first STOP agent computes its timetable and sends the information to the following one. Although the STOP agent knows the identifier of the next STOP agent, the agents interact using our ESAC communication model in order to keep a single communication model. The identifier is a visible property of the Public Layer of each agent and a filter, called $F_{\text{Identifier}}$, enables a message to be addressed according to this information.

The basic event in the network is reproduced in our MAS by an interaction: when a vehicle passes a stop on the real network, a warning message is sent from the BUS agent to the corresponding STOP agent. The STOP agent updates its timetable by removing this vehicle from the list of vehicles due (Figure 3). A STOP agent that does not receive any message detects an anomaly and triggers the disturbance processing presented in this paper.

![Figure 3: Agent monitoring of the network activity](image)

4.2 Inconsistencies management

One of the difficulties of timetable management concerns the management of inconsistencies which arise from the data sent by sensors located in built-up areas. Some vehicles may not be located at a significant number of stops and this may result in the triggering of false alarms.
The incorrect location of a vehicle may lead to inconsistent situations with “virtual overtakings” (a vehicle is announced before the vehicle which precedes it). In order to detect these anomalies, we define two protocols based on communication between the STOP and BUS agents involved (Figure 4):

- **Interception Protocol**: when a vehicle is no longer located, this means that the STOP agents on the bus route have not been warned about the passage of the bus. With our interaction model, they will intercept all new transit announcements sent by vehicles not running to timetable. When the interceptor agents receive the message, they update their timetable.

- **Update Protocol**: in the case of “virtual overtakings” a STOP agent receives a transit announcement of a vehicle which is not the bus it is expecting. The receiver of the transit announcement detects an anomaly and sends a message to its predecessor STOP agent on the route in order to announce this event. If this STOP agent is expecting this BUS, it updates its timetable and forwards the message to its own predecessor. If this predecessor has already received the message of the expected BUS, it means that it does not expect the BUS anymore and it does not forward it.

### 5 Disturbance processing using the multi-agent model

#### 5.1 Static modeling of a disturbance

Describing a disturbance using the delay of a vehicle is not sufficient. For example, a vehicle may be running late, but the distance between the previous and the following vehicles is preserved. In this case, a human regulator will not take the disturbance into account and will be more interested in a vehicle with a shorter delay, but which leads to an imbalance along the line. To measure qualitatively the seriousness of a delay, we have taken into account its context and its consequences on the activity of the network. For this purpose, we put together within a specific organization, called the *Incident model* [4], all STOP agents and BUS agents related to a given disturbance, according to the repercussions on the network.

Unlike other transportation information systems which propose an instantaneous image of the network state that regulators have permanently to analyze, we propose the integration of data updates in the disturbance model, thus allowing an incident to be analyzed from start to finish.

The aim of our disturbance model, called “*Incident model*”, is threefold:

1. To automatically search for and collect information necessary for regulators to analyze and solve the problem. This collected information represents the decision context [7].
2. To take into account the real-time characteristics of the system which means dynamically updating this knowledge over time: we propose a formal model of the information sets and their evolution according to changes on the network.
3. To summarize relevant information through several indicators: we propose several measures that are based on the comparison of the relative changes in the information sets.
For this purpose, we have defined three information sets, also called areas:

- a **Successor** area: This area brings together all the STOP agents 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).
- a **Critical** area: This area brings together all the STOP agents where the vehicle is late, it measures the risk assessment of a gap (the late vehicle is left behind by the preceding bus).
- a **Predecessor** area: This area brings together all the STOP agents where the late vehicle is due but not yet late, it measures the risk assessment of a gap (the late vehicle is left behind by the preceding bus).

![Figure 5: The Incident model](image)

By drawing a distinction between the Successor and Critical areas, it is possible to compare incidents in terms of severity. For two incidents with the same number of stop agents between the late bus and its predecessor, the incident with the greatest number of stops in the Critical area is considered as the most severe. The set of these three areas represents the model that we call the Incident model (Figure 5). The Incident model gathers in a single entity all the information necessary to manage a given disturbance. Since the regulators spend a large part of their time gathering information, this model will help them in their daily work.

### 5.2 Dynamic modeling of a disturbance

The initial organization of the multi-agent system (in lines and routes) does not enable the disturbance process to be modeled dynamically. It must be completed by a hierarchical organization of agents linked to a disturbance (Figure 6). In order to aggregate the information, we have defined two more types of agents: the STOPAREA agents and the INCIDENT agent. The advantage of this organization is that, at each level of the hierarchy, information is aggregated by the agents.

![Figure 6: Dynamic organization of agents](image)

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 information synthesis. They collect basic information such as theoretical traffic evaluation and passenger flow from the STOP agents linked to them and compute the progression coefficient (cf. §6.3). The INCIDENT agent represents the top of the hierarchy where risks are computed and contains the interface between regulators and the system.

This organization is dynamic because from one cycle to the other the 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.
The process of hierarchy creation takes place in three stages:

- **Stage 1:** The STOP agent that detects a delay creates the INCIDENT and STOPAREA agents necessary to solve the problem.
- **Stage 2:** This STOP agent sends messages to STOP agents that are expecting the late bus or its following one. These receiver STOP agents send messages to the STOPAREA agent to which they depend.
- **Stage 3:** The STOPAREA agent contacts the BUS agents link to the disturbance, the late vehicle and its following one, to inform them about the disturbance that has been detected.

Each disturbance depends on the network activity and mostly on the appearance and disappearance of the vehicles. As an example, let us consider a particular case study: the planned insertion of a new vehicle by a regulator. The new vehicle is inserted between the late bus and its following one. The fact that the new vehicle becomes the following bus of the late vehicle modifies the definition of the Successor area. The STOP agents that detect a modification of the next vehicle they are waiting for contact the STOPAREA agent concerned to inform it about this event. This agent reacts by considering the new vehicle as the vehicle linked to its area. The STOPAREA agent sends a message to the STOP agents situated between the new successor and the former one to tell them that they are no longer concerned by the disturbance.

When the disturbance disappears, this organization stays during some cycles to keep the continuity of the disturbance process. After a certain period, the agents created with respect to the disturbances disappear.

### 5.3 Measurement indicators of a disturbance

As explained before, one of the advantages of the Incident model is to bring together information on a disturbance. In this section, we show how this aggregated information allows the regulator to obtain a quantitative evaluation of the detected disturbance. We propose to measure two types of risk: the risk of a bus train $R_{bt}$ and the risk of a gap $R_{gap}$. These two indicators are based on the theoretical difficulties that the given vehicles may meet on a network subsection [4].

A data table is associated with each stop of the network for each period of time. For instance, at the stop $p_i$ the estimated value of traffic is 1 (the traffic is normal) and the estimated value of passengers flow is 2 at 8pm ($p_i$ may be a school).

The disturbance dynamics may lead to a gap between supply and demand; when the late vehicle has to answer a strong demand, the following one deals with a weaker one. In order to take this gap into account, we have modified the initial values of the theoretical passenger flow evaluation according to their link to the areas of the disturbance. We have reduced them for the stops in the Successor area and increased them for the stops in the Critical and Predecessor areas. After various observations, we have defined and evaluated a progression indicator $I_p(y)$ for each area $y$, so as to take into account the quantity of the passenger flow in the area $y$. It varies from -1 to 1.

**Definition 1: NextBusWaitingArea**

We define the set of STOPs that a particular vehicle has to serve and which will not be served by another vehicle before its passage (Area 1 in Figure 4).

Let $a_{L}^{i}$ be the $k^{th}$ stop on route $i$ of line $L$, let $b$ be a bus, $b_p$ its predecessor. Let $\text{WaitingArea}(b, t)$ be the set of STOP agents waiting for BUS agent $b$ at time $t$:

$$\text{NextBusWaitingArea}(b, t) = \{ \ a_{L}^{i} \in \text{WaitingArea}(b, t) - \text{WaitingArea}(b_p, t) \}$$

**Definition 2: Progression Coefficient (PrC)**

The progression coefficient measures the disturbance state compared to a normal situation (where traffic evaluation $T(k)$ and passenger flow evaluation $F(k)$ are equal to one for each $k$). If this coefficient is positive, this means that this disturbed area may cause the bus to slow down. If the coefficient is negative, this means that the area may allow the bus to make up for lost time.

Let $y$ be an area, $x$ the number of stops in this area, $T(k)$ the traffic evaluation at the $k^{th}$ stop of area $y$ and $F(k)$ the passenger flow evaluation of the $k^{th}$ stop at area $y$.

Let $y_{NBW} = \text{NextBusWaitingArea}(b, t)$, $x_{NBW} = \text{Card}(y_{NBW})$. 

Let $y_{NBW} = \text{NextBusWaitingArea}(b, t)$, $x_{NBW} = \text{Card}(y_{NBW})$. 

9
\[ k_0 = \min \{ k \mid a^L_{ik} \in \text{NextBusWaitingArea}(b,t) \} \text{ then} \]
\[ PrC(x_{NBW}, y_{NBW}) = \sum_{i=1}^{k_0} \left\{ \left( F(a^L_{ik}) \ast (1 + Ip(y_{NBW})) \right) + T(a^L_{ik}) \right\} - 2x_{NBW} \]

The progression coefficients of the areas linked to a disturbance are used to evaluate the risk of a bus train \( (R_{bt}) \) and the risk of a gap \((R_{gap})\).

**Definition 3: Risk of a bus train \((R_{bt})\)**

Let’s consider the case where the first vehicle may be slowed down but not the following one. If the first vehicle is the late bus, then the risk of a bus train can be computed. If the first vehicle is the predecessor of the late bus, then a possible modification of the disturbance is identified. The actual disturbance may disappear and the predecessor may create a new disturbance. To calculate the risk of a bus train, we compute the difference between the Successor area progression coefficient and the Critical area progression coefficient.

Let \( y_c = \text{CriticalArea}(b,t) \),
\( y_s = \text{SuccessorArea}(b,t) \),
\( x_{min} = \min(\text{Card(CriticalArea}(b,t)), \text{Card(SuccessorArea}(b,t)) \)
\[ R_{bt}(b, t) = \frac{[PrC(x_{min}, y_c) - PrC(x_{min}, y_s)]}{\text{Card}(y_c)} \]

Since an Incident moves on the network, the number of stops in each area varies from one cycle to the next. However, to obtain comparable measures, the number of stops must be the same in the two areas. Consequently, we have chosen to compute the progression coefficient for a number of stops (noted \( x_{min} \)) equal to the minimum number of stops in the two areas. The greater the number of stops belonging to the Successor area is, the lower the risk value.

**Definition 4: Risk of a gap \((R_{gap})\)**

The first vehicle may accelerate but not the following one. If the first vehicle is the predecessor of a late bus, the risk of a gap is identified. If the first vehicle is the late bus, then a possible modification of the disturbance is identified. The actual disturbance may disappear and the successor may create a new disturbance. To compute the risk of a gap, we compute the sum of the progression coefficients of the Critical area and of the Predecessor area. The risk of a gap takes into account the number of stops between the late bus and the previous vehicle.

Let \( b \) be the late bus at time \( t \), \( x_c \) be the number of stops in the CriticalArea\((b, t)\) and \( x_p \) the number of stops in the PredecessorArea\((b,t)\).

Let \( y_p = \text{PredecessorArea}(b,t) \),
\[ R_{gap} = [PrC(x_c, y_p) + PrC(x_p, y_p)] \times [x_c + x_p] \]

In section 8 where results are analyzed, we will see how these indicators allow the regulators to measure the seriousness of the disturbance in quantitative terms. More details on these indicators can be found in [4].

6 Feasible action planning

6.1 Static feasible action planning

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.
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 (Figure 7):

- **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.

### 6.2 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.

---

2 The late vehicle makes a run with alighting passengers only.
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 8).

<table>
<thead>
<tr>
<th>Name</th>
<th>∈</th>
<th>[network procedure]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preconditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H: Hard conditions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P: Profile conditions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S: Soft conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D: Set of requested data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1: Chosen run</td>
<td></td>
<td></td>
</tr>
<tr>
<td>procedureComputation(requested run): String</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: The regulation procedure

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 \( \text{AlightingOnly} \) 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 (\( H = \emptyset \) because \( \text{AlightingOnly} \) is always possible):

- The distance (number of stops) between the BUS agent and the late BUS agent has to be 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 has to be 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. For the \( \text{AlightingOnly} \) 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 8).

Let \( R_j \) be the run that will be modified to insert the requested run. \( R_j \) is the current run or a run that is chosen according to its departure time. For the \( \text{AlightingOnly} \) 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.

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.

7 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). 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.

In the experiments presented here, we first give the results of the disturbances assessment with an example of the evolution of risk indicators over time and how they represent the seriousness and evolution of the disturbance. This is followed by the solutions proposed by the system and how they are displayed to the regulator through an interface.

### 7.1 Disturbance assessment

The three Incident model areas defined in section 5.1 enable a detailed analysis of a disturbance linked to a problematic network section. Figure 9 shows the evolution of the number of stops in the three Incident model areas.

![Disturbance evolution](image)

**Figure 9: Evolution of a disturbance linked to a problematic network section (source [4])**

If we analyze more precisely the evolution of the critical area, we can distinguish 4 steps:

1) Cycles 0 to 22: the number of stops increases considerably (from 3 to 8), which means that the late vehicle is falling more and more behind.

2) Cycles 22 to 30: at the end of this period the number of stops in the area is equal to 4, which means that the vehicle has made up for lost time.

3) Cycles 30 to 35: the number of stops is around 7 at the end of this period, which means that the vehicle is falling behind again.

4) Cycles 35 to 70: the number of stops stays at around 8. This means that the vehicle does not make up for lost time. The decrease in the predecessor area at the end of the disturbance (Figure 9 after cycle 55) is due to the arrival of the predecessor vehicle at the terminus (no new stop is added to the predecessor area).

This behavior may be explained by analyzing the behavior of the Successor and Predecessor areas through the study of our indicators (Figures 13 and 14). The disturbance is created and then the number of stops increases in the Critical area because the late vehicle has stopped. After this, the late vehicle is freed (at cycle 22) and the number of stops in the Critical area decreases (from 8 to 5) before stabilizing at around 7 stops. The stops lost by the Critical area are gained by the Successor area and stabilize at around 6.

![Risk of a bus train](image)

**Figure 10: Evolution of the risk of a bus train (source [4])**

These indicator values may help the regulator to decide when to intervene on the network. Figure 10 shows that, if the regulator decides to intervene if the risk of a bus train or a gap increases over several cycles with a value greater than 2, this must be done at around cycle 10. The decrease of the risk after cycle 15 illustrates the risk of transfer of the disturbance from the current late bus towards the following one.
Figure 11 shows the evolution of the risk of a gap. Since its value is less than 2 this means that the disturbance does not require any intervention from the regulator.

![Figure 11: Evolution of the risk of a gap (source [4])]()

This example illustrates how our Incident model allows a very precise disturbance assessment. Existing systems detect the vehicle delay but do not put the seriousness of these delays into perspective. Thanks to the Incident model, the regulators are able to know the real difficulties of a vehicle.

7.2 Solutions proposed by the system

As said before, the system has been tested using real data from the Brussels Intercity Company network (STIB). In the following example, line number 54 has been studied as follows:

- **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.

- **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.

- **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 (Figure 12 and 16). Every 3 min, 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 12.A); the second choice is a run with alighting only (M = AlightingOnly, Figure 12.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 12 (A and B)). For each procedure, several items of data are given: the current delay (7 min 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.
Let us take the following situation where BUS agent #54806 is running 7 mn late, behind BUS agent #54827 which is on time. BUS agent #54827 proposes to decrease the gap on its next run by modifying its own departure time, through a procedure call derive. In Figure 13 #54827 leaves 3mn and 32s after its scheduled departure time and #54806 will leave 7mn after its scheduled departure time. The objective is not to manage the current delay but to ensure that the late bus runs at regular intervals.

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

7.3 The user interface

We have designed a special interface structured in such a way that all the computed data (detected disturbances, risk indicators, proposed solutions) may be clearly displayed to the regulators (Figure 14). This interface gives a summary of all of the data related to the problems on the network. The location of each disturbance is represented by a color code. It is possible to access its description by clicking on the reference of the late vehicle (here #54806) and of the following one (#54830). The information given is its timetable. The assessment of the risks, risk of a bus train and risk of a gap, is given by a triangle, the distortion of which informs the regulator of the relative seriousness of the risks. Where the letter P appears in the list of disturbances, solutions have been proposed to cope with the problem. In our example (Figure 14), a bus can either stop for alighting passengers only or go to another point on the line without any passengers (a dead run). The proposed dead run covers 15 stops (from MAELBECK to ST DENIS) and makes up the initial 7 min. delay of vehicle #54806.
Model validation is an important part of the modeling process as this is when the suitability of a model with respect to its potential use is examined.

We have undertaken two types of validation: face validity with the regulators and the decision-makers in charge of the STIB network and predictive validity.

- **Face validity:** This is where a group of experts who know the domain give a surface impression of how realistic the model is. We proposed our model to decision-makers, who quickly understood the principles of the model provided we didn't insist on details of multi-agent modeling.

- **Predictive validity:** this is where the results of the models are tested and compared with reality. We tested the predictive validity of the outputs of the system in terms of the number of disturbances detected and type of solutions proposed.

As said above, we ran our system on the data recorded by the STIB. The system detected more than 300 disturbances whereas the regulators were able to detect only 10% of the disturbances. This can be explained by the fact that when operators detect a disturbance, they immediately try to solve it which is time-consuming, thus preventing them from detecting other disturbances.

Concerning the type of solutions proposed, we said in the previous paragraph that the regulators proposed a solution with external resources. Since this option is not yet available in our system (in future work we plan to introduce this option in the system), we asked them to propose solutions with internal resources; we found that SATIR proposes the same solutions as the regulators.

To test the predictive power of our indicators, the first step was to obtain a coherent set of disturbance data. Since we had to deal with a huge amount of data, we limited our study to only one bus line which is problematic. We wanted to know if our indicators are able to predict the behavior of a disturbance after a number n of cycles (remember that each cycle lasts around 40s). In this example, we have chosen n=7 (around 5mn). After consolidating the data, about 50% of the initial number of disturbances was kept. For each disturbance, Rtb is computed at each cycle and compared with the observed evolution of the disturbance. If Rtb was positive then a decrease in the number of STOPs in the Successor area is forecast. Figure 15 shows that in 70% of cases our prediction model is correct within a time horizon of 7 cycles. As expected, the curve decreases as the time horizon increases.
8 Conclusion

In this paper, we have presented a Transportation Decision Support System (TDSS) called SATIR based on a multi-agent model. Unlike the other approaches that only deal with a specific task, the original feature of SATIR is that it proposes a global approach to the regulation function. The global approach models in the same system the three regulation tasks of human operators: network monitoring through information synthesis, disturbance diagnosis and action planning. Since the multi-agent model is integrated in the decision support system, it detects disturbances, manages them and looks for a solution to reduce the problems resulting from a vehicle being late. For this purpose, we have defined a new model, called the Incident model, that allows dynamic information synthesis useful for decision-making.

Several key contributions should be highlighted. First, the multi-agent paradigm and framework provide an adequate structure to develop a distributed DSS for urban public transportation system management. Second, the original concept of active environment that we have introduced allows the agent to interact directly, sending and receiving messages through logical emission, reception and interception filters. The technique of the active environment is used to make the model consistent when a bus has reappeared after several stops. Logical Interception filters enable the passed stops to update their next expected bus time. Third, the multi-agent approach allows the dynamic management of the bus timetable, bus timetable follow-up in real-time and comparison with the theoretical bus timetable during the day. The detection and display of the disturbance enable the operators to quickly identify the malfunctioning of the lines of the bus network.

Used as the basis of a decision support system, our multi-agent system has been assessed over several test days in real size. Using real data from the Brussels bus network, the system has been shown to work despite sensor failures.

More research has to be done in several directions: 1) the system has to be fully tested and validated; 2) the dynamic management of the filters needs to be improved, so that the agents can add and remove filters from the environment. This dynamic management requires the use of knowledge on category acquaintances. This new type of agent is expected to increase its autonomy by building its own communication filters according to the evolution of its needs; 3) the user dialog and interfaces must be developed to allow human regulators to ask for explanations of the disturbances; 4) statistical tools have to be added to better understand the network operations; 5) more than one line and interchange problems must be taken into account.

More generally, the use of the multi-agent paradigm for network management opens new perspectives. Since our system reproduces the network activity, we can run it:

1) to simulate new timetables. When disturbances are detected, the regulators and graph makers solve the problems by changing the timetables; the simulator could validate the changes by reproducing the network activity using data concerning recurrent difficulties. One way to measure the improvement of the service would be to look at the new number of disturbances.

Figure 15: Validation of Rtb (risk of a bus train) and Rgap (risk of a gap)
2) to serve as a training tool. The behavior of the vehicles could be simulated in order to create disturbances while the system could assess the behavior of the regulator according to the number of disturbances on its test network.

9 Bibliography


