PERGAMON

Transportation Research Part C 10 (2002) 473–506

TRANSPORTATION RESEARCH PART C

www.elsevier.com/locate/trc

Multiagent architectures for intelligent traffic management systems

Josefa Z. Hernández [a,*], Sascha Ossowski [b], Ana García-Serrano [a]

[a] *Department of Artificial Intelligence, Technical University of Madrid, Campus de Montegancedo s/n, Boadilla del Monte 28660, Spain*

[b] *Artificial Intelligence Group, E.S.C.E.T. University Rey Juan Carlos, Spain*

Abstract

This paper reports our experiences with agent-based architectures for intelligent traffic management systems. We describe and compare integrated TRYS and TRYS autonomous agents, two multiagent systems that perform decision support for real-time traffic management in the urban motorway network around Barcelona. Both systems draw upon traffic management agents that use similar knowledge-based reasoning techniques in order to deal with local traffic problems. Still, the former achieves agent coordination based on a traditional centralized mechanism, while in the latter coordination emerges upon the lateral interaction of autonomous traffic management agents. We evaluate the potentials and drawbacks of both multiagent architectures for the domain, and develop some conclusions respecting the general applicability of multiagent architectures for intelligent traffic management.

Keywords: Intelligent traffic management systems; Multiagent systems; Coordination mechanisms; Knowledge-based architectures; Real-time decision support

1. Introduction

The notion of software agents has become increasingly popular over the last couple of years. Public entities as well as private companies spend a considerable amount of time, effort and money in research development and promotion of the idea of software agents. To some respect

* Corresponding author. Tel.: +34-91-336-6901; fax: +34-91-352-4819.

E-mail addresses: phernan@dia.fi.upm.es (J.Z. Hernández), s.ossowski@escet.urjc.es (S. Ossowski), agarcia@dia.fi.upm.es (A. García-Serrano).

PII: S0968-090X(02)00032-3

this is surprising, as software agents essentially do what one expects a reasonable advanced computer program: it is embedded in an environment in which it is capable of achieving certain tasks with some degree of autonomy, i.e. without constant human guidance or intervention.

Still, the real potential of this technology becomes unleashed when several software agents are put to use in the same environment. In this case, the group of agents is usually conceived as a *multiagent system*, as the successful completion of their tasks is subject to the decisions and actions of other agents. So, in multiagent systems, agents are forced to *coordinate* their activities so as to avoid negative interactions with their acquaintances and to exploit synergic potentials. Initially, most distributed problem-solving systems were based on a distinguished agent, which achieved the coordination of the activities of its acquaintances in a *centralized* fashion (Steeb et al., 1988; Von Martial, 1992). More recently, the focus has shifted to more autonomous agents that coordinate in a *decentralized* fashion (Ephrati et al., 1995; Decker and Lesser, 1998).

An important reason for the growing success of multiagent technology is its potential to cope with *high complexity* problems that show an a priori *distribution* (Ossowski, 1999). From the technical point of view, the inherent distribution allows for a natural decomposition of the system into agents that interact so as to achieve a desired global functionality. By this, reusability is promoted, not just by the agents' modularity, but also based on their autonomy. In addition, the scalability of systems operating in highly complex domains can be improved by choosing among specific coordination models that harmonize agent activities with respect to the multiagent system's task. Through an adequate design of coordination mechanisms for a particular problem, it is believed that, from an economic point of view, multiagent systems may tackle complex problem with an acceptable degree of performance, but at a lower cost than traditional solutions.

Multiagent systems have been applied successfully to a variety of industrial problems (Chaib-Draa, 1998), from electronic commerce (Guttman et al., 1998; Foss, 1998) through energy management (Jennings, 1994; Correra et al., 1994), to road transportation related applications such as parking guidance tools or transportation scheduling systems (Fischer et al., 1996). So, it is natural to ask whether this promising new technology is capable of coping efficiently with advanced road traffic management scenarios. In the end, traffic management problems are intrinsically distributed, and suffer from a high degree of complexity that is constantly increasing in line with the incessant improvements of traffic control infrastructures.

This article reports our research and experiences in this respect: it analyzes how multiagent architectures can be applied to the problem of strategic road traffic management. Section 2 discusses the role of knowledge-based artificial intelligence techniques for traffic management. It argues in favor of the concept of intelligent traffic management systems (ITMS) as a means of *integrating* the increasingly complex and heterogeneous traffic control infrastructure and providing a means of *strategic* support for traffic management. Section 3 shows how knowledge-based ITMS for real-world traffic management problems can be instrumented in a computationally and economically efficient way by means of multiagent architectures. Two ITMS aimed at the management of the Barcelona freeway system, and based on different multiagent architectures, are described and compared. Finally, in Section 4, the potentials and drawbacks of both systems are analyzed in order to come up with a set of conclusions respecting the applicability of multiagent technology to the ITMS domain.

2. ITMS: coupling traffic management and knowledge modeling

In recent years, the evolution of the telematics infrastructure and technology has significantly increased the management possibilities of the traditional traffic control centers (TCCs). Whereas in urban control the focus still lies on traffic light control, the control options in urban and interurban motorways are manifold. Besides the different possibilities for *direct* control on motorways, indicated by Fig. 1, a variety of *indirect* control measures can be applied:

- Recommendations for the drivers by means of VDS (variable direction signs and text panels).
- Warning messages via broadcast, RDS/TMC or cellular-phone-based services.
- Pre-trip information provided by different institutions (e.g. road work survey for the next day by means of newsletters; a congestion calendar for holiday times provided by motorway associations).
- Individual driver information systems.

By consequence, complex information management tools need to be incorporated into these TCCs, in order to take profit from the lots of traffic information periodically received. The most fine-grained unit in the conceptual structure of these tools are different types of sensors, that

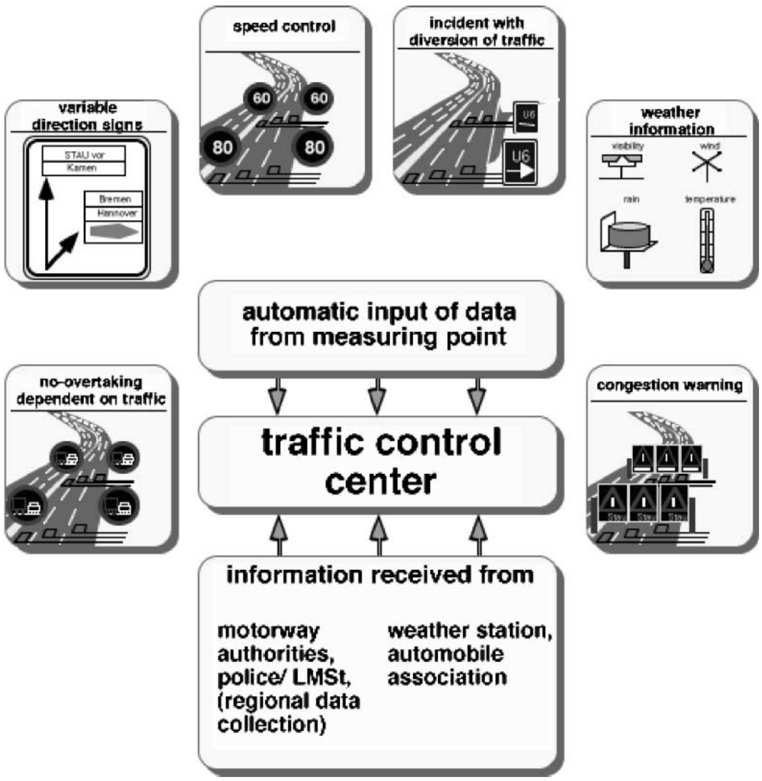

Fig. 1. Current possibilities of traffic control (from Kirschfink, 1999).

collect traffic data (most commonly inductive loop detectors), and in some cases also environmental information. Several such detection sites are assigned to an outstation that regulates isolated control instances, like separate intersections. For stretch control or network control, several outstations are assigned to a subcenter that supervises subnetworks. Finally, for traffic management and traffic monitoring purposes, all subcenters are connected to a TCC. The TCC is usually manned and coordinates all control measures, as well as it interacts with the police to manage congestion situations. Fig. 2 gives an overview of a typical infrastructure for freeway control.

The objective of traffic management systems is to make use of this information infrastructure so as to interpret the different states of traffic flow, and react accordingly in order to maintain or improve the overall performance of the supervised network. This kind of activity obviously requires an intelligent behavior of the system. An effective way of achieving this kind of intelligent behavior, is to endow the system with *knowledge* about the structure and the dynamics of controlled network. This knowledge is supplied by the human operators and is used to identify critical traffic situations as well as to decide on the most appropriate control actions to be taken.

The application of such knowledge-based artificial intelligence systems to traffic management operations has been an active research area in the last two decades (Bielli et al., 1994). Three main advantages can be obtained from the use of these techniques:

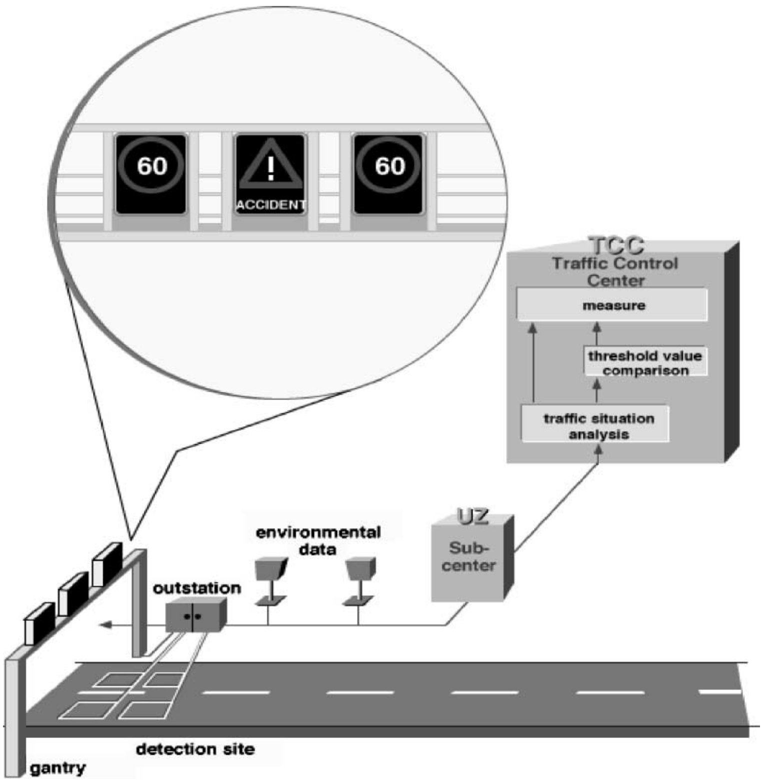

Fig. 2. Typical traffic control infrastructure (from Kirschfink, 1999).

- The possibility of iterative modeling, using declarative representations (rules, frames, etc.). This allows for the use of (mostly empiric) knowledge that cannot be formalized satisfactorily in algorithms, such as heuristics used in decision processes. Conventional software engineering methods are not adequate to model such domains.
- The possibility of user-friendly management of the knowledge, especially for non-programmers, because the declarative content of the application facilitates its understanding. This advantage allows the system to be easily adapted to new requirements.
- The possibility of getting explanations for every output provided by the system. An explanation is a description of the sequence of reasoning steps and the chunks of knowledge used by the system to generate a certain output. If the user is familiar with the procedures that have been applied (which is usual in the different areas of expertise), he/she may understand very well the role played by the different pieces of domain knowledge existing in the model. Getting explanations is particularly important for traffic management systems that support the decisions to be taken in the manned TCC: operators can require explanations so as to obtain a deeper insight in the "reasons" underlying the system's control proposal, before taking a final decision respecting its implementation.

Besides these technical reasons, the specific characteristics and the proper dynamics of integrated road transport favors the application of knowledge-based techniques to the management system:

- Firstly, according to the previous description, the introduction and progressive integration of extended traffic monitoring and management facilities in the new generation of traffic management architectures (e.g. improved monitoring systems, incident detection, collective and individual route guidance systems, etc.) demand for increased, on-line operator support tools to help coping with the complexity of both, the information managed and of the resulting, integrated traffic management schemes.
- Secondly, existing traffic management and control systems show limitations when facing critical traffic conditions and congestion. This is the approach followed by systems like SCOOT (Hunt et al., 1981; Bretherton and Bowen, 1990), UTOPIA (Mauro and Di Taranto, 1989), OPAC (Gartner, 1983), SCATS (Lowrie, 1982), PRODYN (Henry et al., 1983). This is an almost permanent problem in most large/mid-sized urban centers in Europe, which is usually caused by a locally conceived analysis of traffic behavior that needs to be complemented by more strategic, high-level control methods.
- Thirdly (and for this reason) the role of operators of traffic management centers is still crucial in day-by-day operations—i.e. no matter how sophisticated and advanced traffic control technology is, the "man in the loop" paradigm is still a prevailing operational condition in most centralized traffic control systems

In this context, the development of systems capable to reason about the traffic behavior and evolution in similar terms to an expert traffic operator becomes essential. The idea is not to replace the methods used with available traffic control technology such us traffic signal control (i.e. optimization-based dynamic control models such us the five previously mentioned) and variable message signs, rather to complement this by extended capabilities and improved performance.

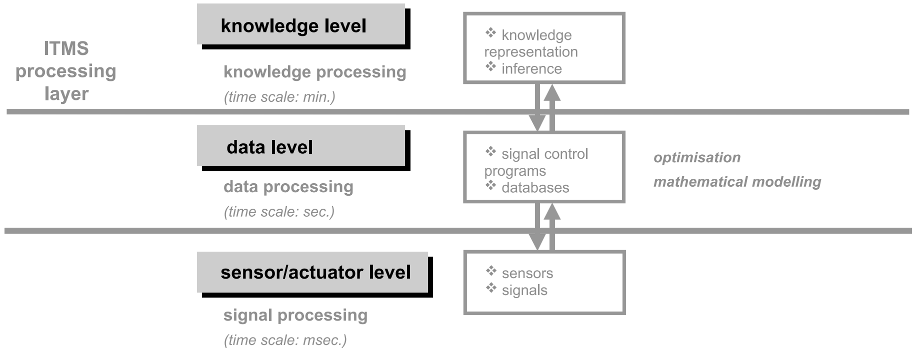

Fig. 3. ITMS operation level in traffic management architectures.

Such concept of system is that of the ITMS: An AI-based tailorable traffic management tool to be integrated in the current traffic management architectures and TCCs, and with an additional functional level in current TCCs, i.e. a "traffic knowledge processing layer" developed on top of the existing traffic control facilities (see Fig. 3). The ultimate goal of this layer is to improve on-line traffic monitoring and management along two main directions: (1) strategic management and intelligent supervision of traffic control strategies within single subsystems, and (2) integrated management of different traffic control services. The approach can be seen in the wider perspective of the multilevel design strategy currently followed in both the European (Bolelli et al., 1994) and American (Gartner et al., 1994) traffic management architectures.

This knowledge layer supports the development and on-line use of models of the traffic operator' knowledge and expertise in the controlled areas. The embedded knowledge models allow improving several traffic monitoring and management operations, including:

- estimation of traffic load levels in space and time all over the network;
- analysis and understanding of traffic demand and routes in the area;
- qualitative prediction of demand and routes;
- detection (prediction) of critical traffic situations and bottlenecks;
- selection and implementation of congestion avoidance/reduction strategies;
- management of conflictive control objectives and priorities in the different controlled areas.

A fundamental characteristic of this layer is that thanks to the explicit knowledge modeling approach, the knowledge and thus the competence of the system (i.e. its traffic control criteria and strategies) can be modified and improved over time as new knowledge is gained about long-term modifications of traffic behavior through day-by-day operation of the traffic network.

The first steps in the development of a knowledge-based traffic management tool developed and tested in a real-life environment have been done in the SAGE system (Forasté and Scemama, 1986). Developed by the INRETS (www.inrets.fr), it is a rule-based expert system implementing the congestion management procedures in use in the TCC of Paris. Its further development has led to the CLAIRE system (Scemama, 1994), which has subsequently been integrated into the new Paris traffic control system SURF2000.

Another number of applications go back to the research carried out at the Institute of Transport Studies, Irvine, California (http://www.its.uci.edu). Most of the work concerns freeway management systems, including aspects of safety and incident management operations. Examples of these include the systems FRED (Ritchie and Prosser, 1990), FIM (Ritchie and Stack, 1993) and ARTIST (Deeter and Ritchie, 1993). Further work includes also the development of hybrid models integrating symbolic (knowledge-based) and subsymbolic (connectionist) models (Molina et al., 1995). Several of such systems have been tested in the Orange County highway system.

Other work on knowledge-based architectures for traffic network management has been carried out under the European Commission DRIVE research programmes in projects like KITS (Cuena et al., 1992) and ARTIS (Cuena et al., 1994). Both systems address motorway traffic management and follow and architectural approach that is conceptually founded in the TRYS system (Cuena et al., 1995, 1996a, b). Such systems have undergone testing and evaluation in the city of Florence in the case of KITS and Madrid for ARTIS. In the last years, this EC financed research has been oriented to more integrative concepts like human–computer interaction oriented ITMS design (FLUIDS, Cuena et al., 1998; Hernández and Molina, 1998) and the concept of advanced traffic information centers (TICs) (ENTERPRICE, Behrendt et al., 1997; Boero et al., 1997) that allow collecting, processing and distributing information from a variety of sources to a large number of users of the transportation systems using varied channels. These TICs have acquired a strategic role in the system approach adopted for traffic and mobility services management in the Trans European Transport Network.

Today, the first commercial applications integrating knowledge-based techniques into operational systems for network traffic management have started to appear on the market. As an example, the French company Alcatel is offering the previously mentioned CLAIRE system as a commercial product. Provided by Steria (http://www.steria.com), Traffic Expert is a motorway traffic management tool that includes KB elements to monitor traffic, trigger management actions and help operators planning ahead.

3. The TRYS intelligent traffic management systems family

This section describes the applications of multiagent architectures to knowledge-based ITMS. In the first place, the TRYS traffic management generic architecture is outlined. Subsequently, the knowledge-based reasoning model of TRYS traffic management agents is outlined. Finally, in Section 3.3, the different multiagent coordination architectures of two real-world ITMS applications are discussed and illustrated by an example.

3.1. TRYS architecture overview

TRYS (Cuena et al., 1995, 1996a) is an agent-based environment for building ITMS applications for motorway networks. It provides a generic and modular knowledge model supporting an intelligent reasoning layer that can complement conventional traffic management application capabilities. The TRYS approach has been applied to develop several ITMSs, all of them installed and tested on-line in TCCs, giving rise to the so-called TRYS family of systems. Two relevant members of this family are the integrated TRYS (InTRYS) (Cuena et al., 1996b) and TRYS autonomous agents ($TRYSA_2$) (Ossowski et al., 1998; Ossowski, 1999) applications, described in more detail in this section.

InTRYS is the result of the technical work oriented to integrate the TRYS environment as part of the existing traffic management infrastructure in the TCC in Barcelona. Conceptually, the main difference between the original TRYS approach, applied for instance in Madrid, and the InTRYS one lies on the bigger complexity of the Barcelona network with side ways parallel to the main road along the whole ring road supervised and including ramp metering devices.

$TRYSA_2$ is a decentralized multiagent system for traffic management that targets the same problem domain as InTRYS. Contrasting InTRYS, $TRYSA_2$ has been developed as an experimental prototype only for laboratory purposes, in order to gain experiences with decentralized multiagent architectures for their future industrial application. [1]

Despite a minor reorganization of problem areas, large parts of the *local* knowledge and reasoning models of the InTRYS traffic management agents (as well as the corresponding software) has been reused.

Therefore, InTRYS and $TRYSA_2$ are two agent-based ITMS for the Barcelona road network that show significant similarities:

(a) Both are knowledge-based applications for real-time traffic management in periurban networks.
(b) The topological structuring approach is the same: the organization of knowledge and reasoning matches the topological organization of the traffic network and its spatial break down into a set of distinct but overlapping *problem areas*.
The decomposition of the city into problem areas allows a better analysis and understanding of the causes and evolution of traffic problems than if performed from a global perspective. This split does not define a set of disjointed areas whose sum is the whole city, but every area represents a part of the city where a determined traffic behavior is usually present and where a set of signal elements can be managed to influence this behavior. Then, a problem area may overlap with surrounding areas sharing, for instance, some signals but using them from different points of view. So, a problem area is a part of a city where traffic behavior is locally studied and suitable control actions may be defined to improve the traffic state.
(c) The knowledge and reasoning related to different problem areas are kept in distinct *agents*. These traffic agents receive traffic data, detect traffic problems in their problem area, diagnose their causes and generate signal and/or VMS actions to overcome them.

However, both systems differ significantly in the way that these traffic agents are *coordinated* (as shown in Fig. 4).

Centering the description on the Barcelona problem area and in line with the traffic engineer's logical subdivision of the road network into *problem areas*, the InTRYS system relies on a set of 18 knowledge-based traffic control agents, each responsible for traffic management in one such area. This number of agents was obtained considering both senses of traffic for every major ad-

[1] A project aimed at the employment of this new technology to a traffic management infrastructure in the Spanish autonomous region of Navarra is currently under evaluation.

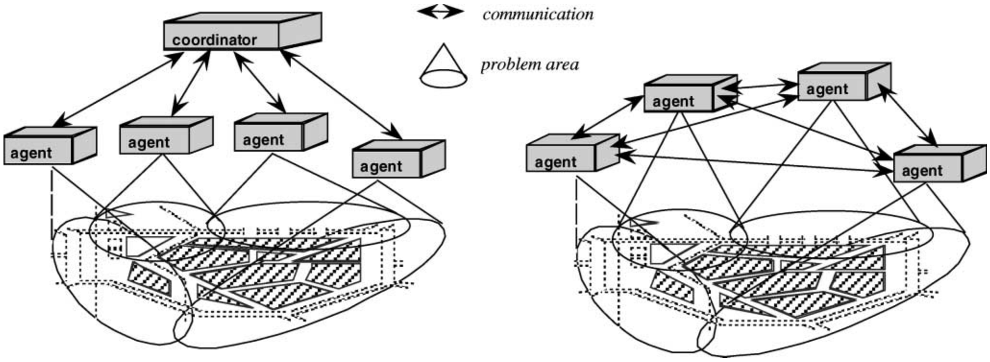

Fig. 4. Centralized (InTRYS) and decentralized coordination (TRYSA$_2$).

jacent road, and a partition of the Barcelona ring road in four parts. As mentioned above, the network areas assigned to the agents are control dependent, as some agents have to share control devices. The InTRYS system faces this problem by means a special coordinator agent endowed with knowledge on how to integrate local control proposals into a coherent global signal plan for the whole traffic network. It receives local control proposals from the traffic control agents, resolves conflicts between them, and sends the resulting globally consistent local signal plans back to the traffic agents.

In TRYSA$_2$, 11 spatial problem areas are controlled by *autonomous* (self-interested) traffic agents that coordinate laterally, based on a mechanism called *structural cooperation* (Ossowski and García-Serrano, 1999). Control devices "belong" to certain agents, and the corresponding mutual dependence (agents may exchange favors respecting the use of "their" devices) provides a potential for cooperation. Normative prescriptions permit or forbid the use of some devices for certain agents, thus biasing agent interactions and influencing self-interested agent behavior, so as to make it functional with respect to the desired traffic management functionality (Ossowski, 1999).

In the sequel, we will describe relevant details of both architectures. First, a description of the local traffic management agents common to both systems is provided, although a more detailed description of these components can be obtained from (Cuena et al., 1995). Then, the two different approaches for coordination are presented and illustrated with an operation example.

3.2. Traffic management agents

The goal of a traffic management agent is to provide two types of information: (i) diagnosis of the traffic problems present in a local area together with an explanation justifying such a diagnosis, and (ii) proposed control actions for the available signal devices to improve traffic conditions using the diagnosis information. In order to achieve this goal agents are endowed with four main types of knowledge: possible traffic problem scenarios, potential traffic control proposals, and regarding traffic behavior in the network, the network structure and historic traffic demand. This knowledge endowment has been organized in so-called knowledge units (KUs) (Cuena and Molina, 1997). A KU identifies a knowledge block where both what the unit knows and what the unit does with this knowledge are put together. Then, a knowledge model is a hierarchically organized collection of KUs.

The *behavior KU* includes all the information related to traffic behavior in the network. This information is organized in two other KUs: the *physical structure KU* and the *demand knowledge KU*. The physical structure KU perceives the scenario data from loop detectors etc., records it, and analyses it to deduce signal information, basic traffic qualitative magnitudes (i.e. speed, occupancy, flows) and certain aggregated measures (i.e. traffic volume generated in an entrance node, spatial gradient of speed). This unit includes both static information about the topological structure of the problem area (components and their relationships) and the dynamic aspects of this structure. The physical structure is formulated using a declarative description of the network as a graph with nodes and links together with abstraction functions associated to components.

The procedures applied with this kind of knowledge are data abstraction methods that compute qualitative values from basic traffic parameters provided by sensors, i.e. speed, occupancy, saturation. Fig. 5 shows the functions used to abstract the speed, saturation and occupancy numerical values. These functions are possibility functions (typically used in fuzzy logic) and although default functions exist, it is possible to use customized versions of these functions for specific parts of the problem areas.

The *demand KU* contains historical information of traffic flow distribution among the different entry and exit points in the network that can be used in a twofold way: both to support the diagnosis of traffic problems and the evaluation of control proposals. This information is represented by using hierarchies of patterns of demand associated to temporal intervals, in such a way that is possible to determine demand scenarios for a given present or future state. Fig. 6 shows a partial example of such a demand pattern. The first set of slots is used to define the temporal range where this demand is active, e.g. in this example, a working day from 6:30 to 9:30 am. Then, slots representing origin–destination pairs with an approximate estimation of the traffic flow are included. For instance, from the origin Mataró to the destination Badalona Norte there will be an approximate flow of 1000 veh/h. This type of frames are not general, i.e. they must be given for each particular traffic area.

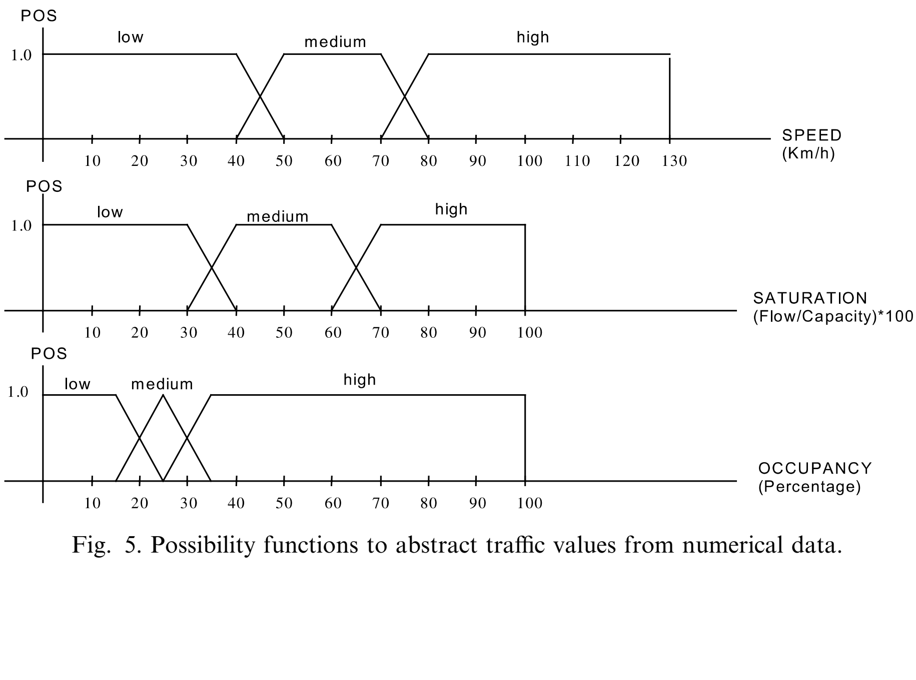

Fig. 5. Possibility functions to abstract traffic values from numerical data.

```
FRAME morning peak TYPE weekday pattern
    type of day:          labour day,
    temporal interval:    6:30 - 9:30,
    ....
    Mataro -> Badalona Norte:            1000 veh/h,
    Mataro -> Badalona Centro Sur:       1000 veh/h,
    Mataro -> Zona Urbana:               2000 veh/h,
    Mataro -> Glories:                   2000 veh/h,
    NII -> Badalona Norte:               500 veh/h,
    NII -> Badalona Centro Sur:          500 veh/h,
    ....
    Badalona Centro Sur -> Glories:      1000 veh/h.
```

Fig. 6. Partial example of a pattern of traffic demand.

The *traffic problems KU* is specialized in the detection and explanation of different traffic situation that may appear in its problem area, like free flow, dense traffic, incident congestion or overflowing congestion. The goal is to analyze the data recorded by sensors to (i) point out the current or foreseeable presence of problems taking into account the recent trends of traffic conditions and, (ii) explain the severity and possible cause of these problems to be used later in the selection of the appropriate control actions.

In general, an agent understands traffic problems as an imbalance between capacity and demand which generates an increase in density that impedes the fluidity of traffic. In a city network, the problems appear when a queue of vehicles propagates to the surrounding streets, blocking intersections and generating a so-called congestion tree. In a motorway, the problems are a consequence of a loss of capacity due to an unexpected incident or to the characteristics of the network.

The problems are analyzed not only by observing the instant where the problem has been detected but with certain deepening in time. In this way, the presence of short-term congestion can be foreseen. This capability of short-term prediction is very important for a system willing to propose control actions that contribute to the prevention of problems, and not only the solution of currently detected problems. This aspect is important because the effort needed to avoid congestion is significantly less than the one necessary to solve an existing congestion.

The approach made to describe a problem and to explain its causes is based on associating a traffic problem with an *excess* traffic volume: it is conceived that a problem appears because the traffic demand received by a section of the network is bigger than the capacity offered by that section. According to this, the characterization of a problem is based on the following issues:

- *Where is the problem?*
  A problem is located in a so-called critical section, i.e. the point where the congestion starts due to a lack of capacity. Examples of critical sections are works on the road disabling a lane, an

entry ramp to a highway with a very high traffic demand, a turn movement controlled by a signal with a short green time, an accident that blocks a lane, etc.

- *How severe is it?*
  The size of the imbalance between the network capacity and the traffic demand is a measure of problem severity. This imbalance is characterized by the excess of arrivals (measured in veh/hour), which is understood as the minimum demand decrease (or capacity increase) needed to solve the actual problem. It may be computed as the difference between the flow demand arriving at the section where the problem is located, obtained upstream of the congested area, and the capacity in that section. An accurate estimation of problem severity is a key aspect to pose an effective control plan.
- *What is the cause of the problem?*
  Finally, the concept of participation is also managed to distribute the causes of the problem among the paths that cross the critical section carrying a significant traffic flow. This information is important to understand the origin of the problem and to pose adequate actions on the points of the network that generate the traffic.

The goal of problem identification reasoning is to analyze data from sensors to detect which are the active critical sections, indicating the excess of arrivals and the paths involved in the problem. In order to carry out this task, the necessary knowledge is represented using frames. The frame language provides a traffic language that can be used by an expert to express his/her experience in detecting and explaining problems. A frame is a collection of state variables for traffic and signals, characterizing the situation in an area, which represent prototypical situations (each one of these variables is called a slot). The left box in Fig. 7 shows an example of one of these frames. The reasoning method checks which problem prototypes in the frame knowledge base match the current situation described by sensor data, and for every matched frame, a potential explanation of the problem is obtained, providing criteria to later select suitable control actions.

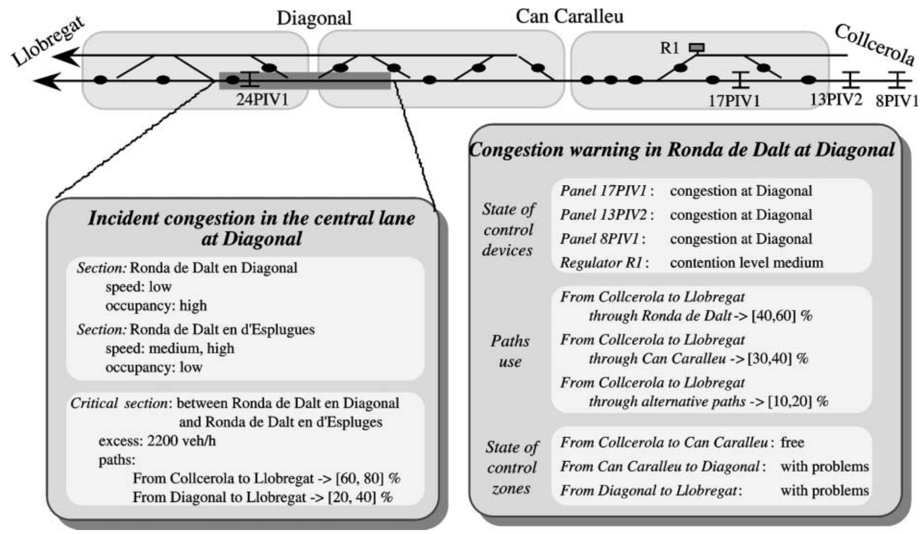

Fig. 7. Problem scenario and path use frames in the model for Barcelona.

Once a traffic problem is detected, the *traffic control KU* is responsible of selecting proposals of VMS panel displays which induce drivers to take paths that do not pass through congested areas and ramp signal plans to decrease the incoming flow to the congestion. The starting point is the set of paths with the greatest influence on the state of the critical sections, called problem paths. The objective is to find recommendations (warnings of slow traffic, warnings of congestion, recommended paths, etc.) that decrease traffic on the set of problem paths.

In this sense, there is another kind of frame used to describe control recommendations. These frames are called *path-use frames*. Every path-use frame includes the following groups of information:

- *Control plan*: It includes a description of a control plan, which consists of a collection of VMS panels with a message to be displayed and, if applicable, a signal plan for the regulators at the entry ramps.
- *Path uses*: It includes a description of the expected impact on the flow distribution with the control plan. It consists of a list of origin–destination pairs with a path and a percentage of the vehicles that may use that path on their way from the origin to the destination point.
- *State of control areas*: It includes a description of the traffic state that should be observed in the problem area in order to apply the previous control plan. The traffic state of the whole problem area is expressed with the state of its parts, called control areas. A control area identifies a part of the network affected in the same way, in terms of traffic behavior, by the impact of any control plan.

Path-use frames are written in a language similar to the one used to write problem frames. The right box in Fig. 7 shows an example of one of these frames.

It is interesting to remark on the way of using this frame base: it may be applied in two reasoning modes: (i) in the prediction mode to evaluate the possible impact of a set of messages on the traffic state and, (ii) in the control mode to identify which frames produce a significant change in the state of the problem focus with respect to the current situation. In the prediction mode, messages are premises and focus states are conclusions. In the control mode, the change in the state of the problem focus is the input and the messages to be displayed are the output. So, the same declarative knowledge is used to achieve two different goals.

The reasoning performed by this traffic control KU is based on a *generate and test* method (Stefik, 1995): First, all the path use frames defined to overcome conflictive situations in the control areas where problems have been detected are selected. Then, the associated signal plan of these frames is evaluated through a simulation of its effects on the current traffic situation, i.e. its influence on the excess. This simulation is supported by the expected redistribution of traffic included in the frame definition. If this simulated scenario shows an adequate decrease in the excess flow associated to the congestion then the signal plan is delivered to the coordinator. Otherwise it is rejected.

3.3. Coordination models

Next, a detailed description of the different coordination mechanisms applied by the InTRYS and $TRYSA_2$ systems is included supported by an example coordination scenario, common for both approaches.

3.3.1. An example coordination scenario

As an example, consider the traffic situation shown in Fig. 8:

- An incident at T1, in the problem area *Ring road from Trinidad to Collcerola*, detected and evaluated by the corresponding agent with a medium severity and an excess of 800 veh/h.
- A usual problem at T2, in the problem area *Ring road from Port Vell to Trinidad*, detected and evaluated by the agent with a high severity and an excess of 1300 veh/h.
- A pre-congestive situation at T3, in the problem area *A-19 from Mataró to Barcelona*, also detected and evaluated by the corresponding agent with a low severity and an excess of 150 veh/h.

The signal proposals generated by the three local agents, in decreasing order of preference, may be the following:

(A) Ring road from Trinidad to Collcerola agent

A1: (expected local excess reduction: 450 veh/h)
P_1: Congestion at T1, sideway recommended
P_2: Congestion at T1, sideway recommended

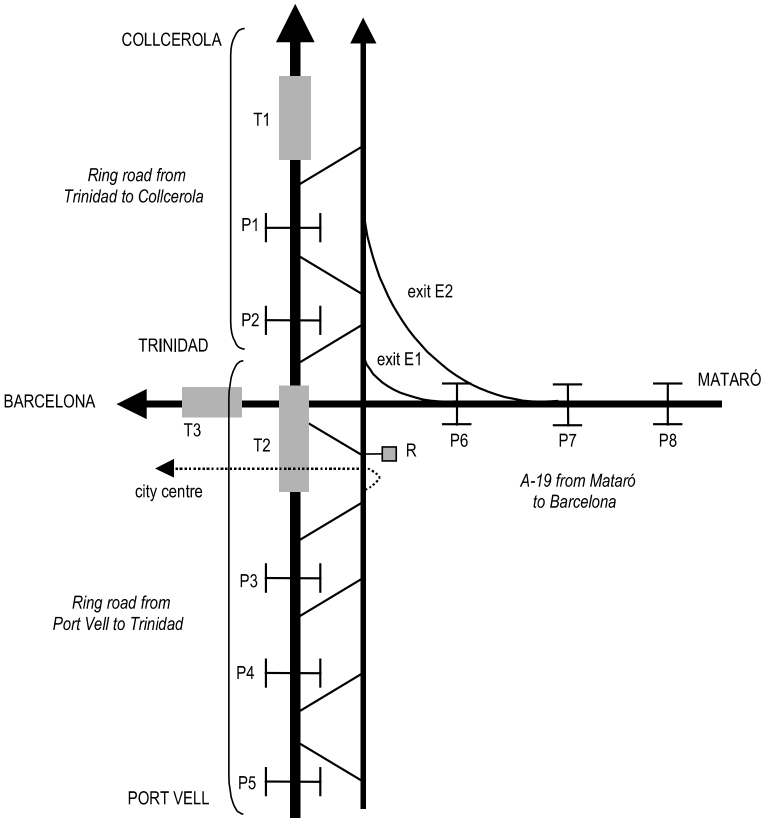

Fig. 8. An example of three problematic situations in part of the Barcelona network.

P_3: Congestion at T1, sideway recommended
R: plan PR3 (always red)
A2: (expected local excess reduction: 300 veh/h)
P_1: Congestion at T1, sideway recommended
P_2: Congestion at T1, sideway recommended
R: plan PR3 (always red)
A3: (expected local excess reduction: 200 veh/h)
P_1: Congestion at T1, sideway recommended
P_2: Congestion at T1, sideway recommended
P_3: Congestion at T1
R: plan PR2 (half time red)

(B) Ring road from Port Vell to Trinidad agent

B1: (expected local excess reduction: 1000 veh/h)
P_3: Congestion at T2, alternative by city centre
P_4: Congestion at T2, alternative by city centre
P_5: Congestion at T2, alternative by city centre
R: plan PR3 (always red)
B2: (expected local excess reduction: 700 veh/h)
P_3: Congestion at T2, sideway recommended
P_4: Congestion at T2, sideway recommended
R: plan PR3 (always red)
B3: (expected local excess reduction: 500 veh/h)
P_3: Congestion at T2
P_4: Congestion at T2, sideway recommended
P_5: Congestion at T2, sideway recommended
R: plan PR2 (half time red)

(C) A-19 from Mataró to Barcelona

C1: (expected local excess reduction: 150 veh/h)
P_6: Very slow traffic at T3, to city centre by E1
P_7: Very slow traffic at T3, to city centre by E1
P_8: Very slow traffic ahead
C2: (expected local excess reduction: 100 veh/h)
P_6: Very slow traffic at T3
P_7: Very slow traffic at T3, to city centre by E2
P_8: Very slow traffic ahead
C3: (expected local excess reduction: 80 veh/h)
P_6: Very slow traffic at T3
P_7: Very slow traffic at T3, to city centre by E2
P_8: switch off

According to the topology of the affected problem areas, described in the figure, the traffic management actions taken in the areas Ring road from Port Vell to Trinidad and A-19 from Mataró to Barcelona have a clear influence, maybe positive or negative, on the traffic evolution in area Ring road from Trinidad to Collcerola. In this way, in some cases part of the traffic diverted from congestion T2 or T3 may enter in the ring road and affect congestion T1. The level of this influence, as consequence of the selection of certain local signal plans, determines the presence of incompatibilities between proposals. In addition, the aim of using a same control device with different messages or plans, as proposed by different agents, originates other type of obvious incompatibilities. All of them have to be solved from a coordination intelligence to configure a global signal plan for the whole network. Next two sections show how these global signal plans are achieved following the centralized and decentralized approaches of InTRYS and $TRYSA_2$.

3.3.2. Centralized InTRYS coordination

In the InTRYS approach, the coherent integration of local signal proposals generated by the agents is responsibility of a *coordinator* agent. The coordinator collects the local proposals and looks for a coherent synthesis of these proposals into signal recommendations for the whole network. This functionality is performed through a *propose and revise* method (Stefik, 1995) supported by three KUs (see Fig. 9):

- Knowledge to evaluate the compatibility of the different proposals delivered by the local agents included in the *proposals compatibility KU.*
- Knowledge to solve coordination conflicts within the *proposals integration KU.*
- Knowledge to deduce complementary signals that improve the global impact of the previous proposals included in the *proposals completion KU.*

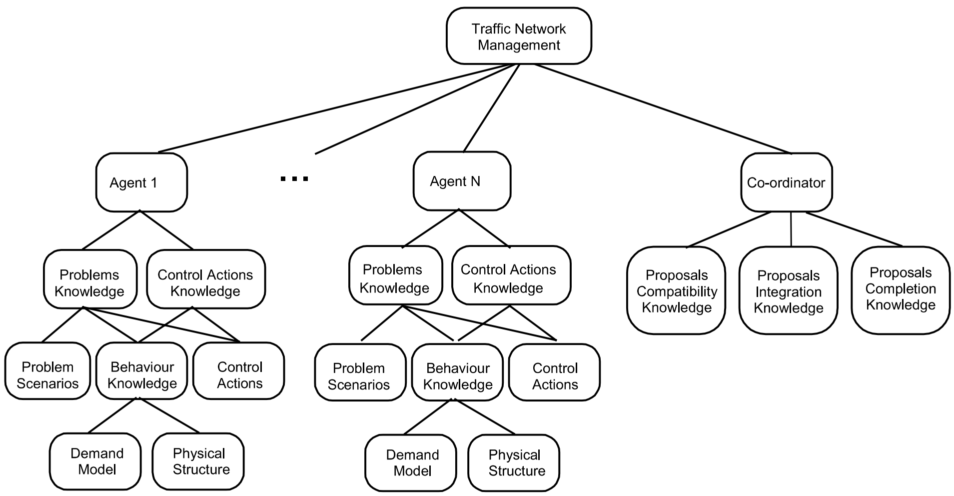

Fig. 9. General structure of a TRYS model.

The knowledge used by the proposals compatibility KU is a set of characterizations of nogood situations, i.e. those where two local proposals are incompatible for being simultaneously part of a global proposal. Two types of nogood situations are considered:

- *Physical conflicts* caused by different actions on the same signal device. For instance, two agents propose displaying different messages on the same VMS panel.
- *Semantic incoherence* between proposals for different signal devices that may be incoherent from the point of view of a user traversing a given path. For instance, a panel recommending a certain speed and another one, in the same or a neighboring area, suggesting a significantly different speed.

The representation used to establish incompatible situations is rule based, expressing the inconsistency between signal actions. For example, a message suggesting a certain route is incompatible with a message of road works in the same path. Rules are managed by a procedure whose inputs are pairs of control actions and whose output is their qualification as compatible or incompatible. Rules are formulated as logic clauses.

According to the traffic situation and proposals described in the previous example, the following conflicts may be detected:

- Between the Ring road from Trinidad to Collcerola and the Ring road from Port Vell to Trinidad agents.
  - Semantic incompatibility between A1, A2 and A3 proposals with B2 and B3, since the traffic using the sideway according to B2 or B3 may reenter the main road and worse congestion T1.
  - Physical incompatibility of different messages for panel P_3 and different plans for the ramp regulator R.
- Between the Ring road from Trinidad to Collcerola and the A-19 from Mataró to Barcelona agents.
  - Semantic incompatibility between A1, A2 and A3 proposals with C1, since the traffic using exit E1 may reenter the main road and worse congestion T1.

After these incompatibilities are detected, the proposals integration KU performs two tasks:

(i) First, it tries to satisfy the most preferred proposals of the agents in conflict if this conflict is caused by a physical incompatibility with VMS panels. In this case, it tries to synthesize the different messages in a common text. For instance, this is the case for proposals A1 and B2 with regard to panel P_3, since a message *Congestion at T1 and T2, side way recommended* may be displayed and satisfy the goals of both agents.

(ii) Then, it looks for a couple of compatible local proposals for the agents in conflict. If the incompatibility between two proposals is solved in the previous step then the output proposal will be the set of the original messages and the synthesized ones. Using the same example with A1 and B2, the resulting proposal would be:

P_1: Congestion at T1, sideway recommended
P_2: Congestion at T1, sideway recommended

P_3: Congestion at T1 and T2, sideway recommended
P_4: Congestion at T2, sideway recommended
R: plan PR3 (always red)

If no solution was found in the previous step then it looks for alternative coexisting proposals that are selected and ordered attending to different control criteria (e.g., the importance of the area, the severity of its problems). For instance, a global proposal including A1, B1 and C1 is not possible since the physical incompatibility with P_3 cannot be solved nor the semantic incompatibility between A1 and C1. But, according to these conflicts, it is not possible the coexistence of any A*i* proposal and B1 nor C1.

Now, as the Ring road from Trinidad to Collcerola agent is the one with the more severe problem it has higher priority to select the control proposal. It means that A1 and the next preferred B*i* and C*i* proposals are studied. Since the physical conflict between A1 and B2 can be solved and there are no problems with C2, then the first recommendation of the system to the traffic operator would be:

P_1: Congestion at T1, sideway recommended
P_2: Congestion at T1, sideway recommended
P_3: Congestion at T1 and T2, sideway recommended
P_4: Congestion at T2, sideway recommended
R: plan PR3 (always red)
P_6: Very slow traffic at T3
P_7: Very slow traffic at T3, to city centre by E2
P_8: Very slow traffic ahead

In the cases where no global proposal can be defined taking into consideration all the agents with problems, the control criteria are applied to decide which one has to give up applying a signal plan. For instance, if all the C*i* proposals may cause semantic incompatibilities with the A*i* ones then no proposal from the A-19 from Mataró to Barcelona agent would be chosen.

This process of generating signal plan recommendations continues up to the top limit given by the operator or when no more proposals can be obtained.

Sometimes, the control proposals generated in the previous steps can be completed with additional messages chosen from a global, strategical view. This knowledge is organized in a collection of prototype situations contained in the proposals completion KU. Once the coordinator has integrated the control proposals for individual problem areas into a consistent global signal plan, the adapted local plans are sent back to the corresponding traffic operator.

Fig. 10 shows some windows of the InTRYS system. The maps on the right are used to warn about the status of the different problem areas using a color code. In this figure a problem in the A-18 outbound area has been detected. The four windows on the left include different kinds of information. In a top–down way, the first one is used to indicate the status of the application and the type of data used in the reasoning (e.g. real-time, simulated). The second one is describing a signal proposal generated by the corresponding agent. The third includes the description of the observed problem and the fourth an associated explanation.

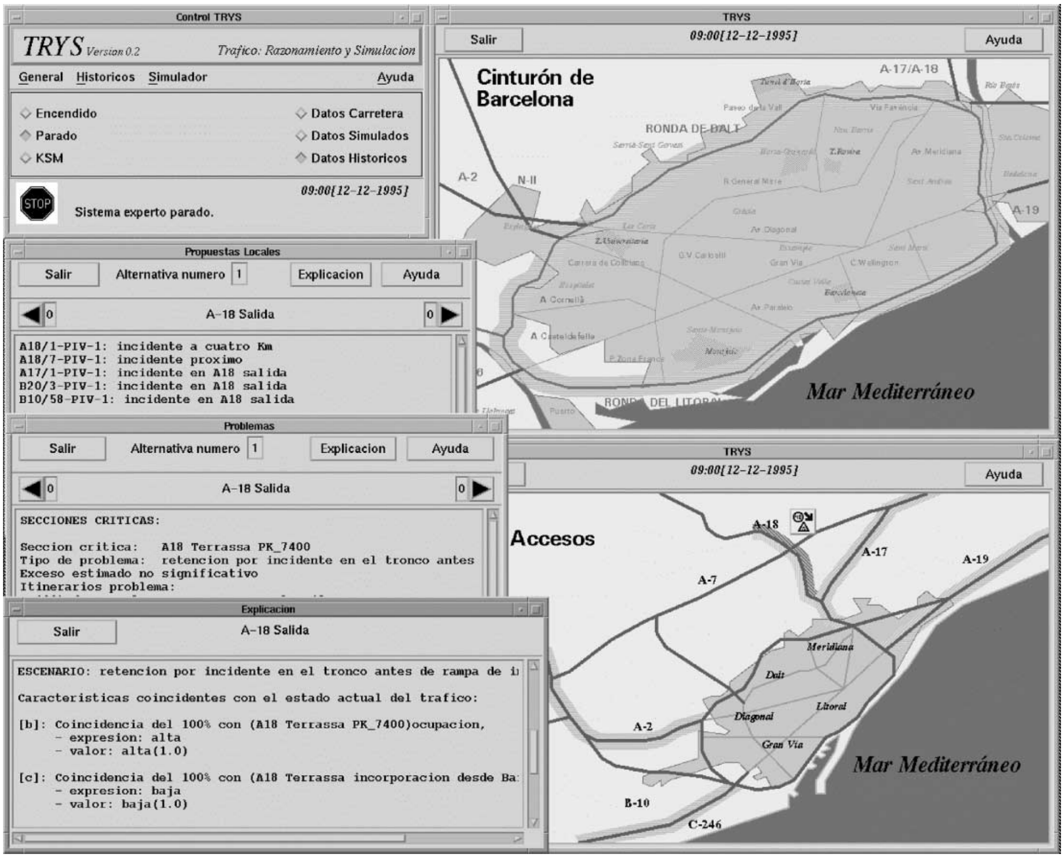

Fig. 10. InTRYS interface.

3.3.3. Decentralized $TRYSA_2$ coordination

In the sequel, we first describe the mechanism by which decentralized coordination is achieved within $TRYSA_2$. The knowledge requirements of the mechanism are outlined, and the distributed algorithm that instruments the coordination mechanism within $TRYSA_2$ is exposed. After sketching the tools to tune the outcome of $TRYSA_2$ coordination, we briefly review some methodological and implementation issues.

3.3.3.1. Coordination mechanism. Decentralized coordination within $TRYSA_2$ is based on *structural cooperation*, a generic coordination mechanism developed for societies of autonomous problem-solving agents (Ossowski and García-Serrano, 1999). By applying this coordination mechanism to the TRYS traffic management framework, the local problem-solving of the traffic management is augmented by a model of self-interested pursuit of (local) goals.

InTRYS agents are "benevolent", in as far as that they accept to take part in any global signal plan that the coordinator generates. By contrast, in $TRYSA_2$ the coordination decision is decentralized: traffic management agents are autonomous, as they are primarily interested in improving traffic problems in their local problem areas. Still, as local signal plans are interdependent, agents can mutually influence the effectivity of their acquaintances' local plans. By consequence, autonomous, self-interested $TRYSA_2$ agents are interested in coordination: any of them can win by finding an *agreement* of mutually adapting the local signal plans in certain ways, but there is a conflict of interest respecting the "details" of that agreement, i.e. respecting the *way* in which this mutual adaptation should be realized. The essential idea underlying the mechanism of structural cooperation is that the more autonomous an agent is (i.e. the less its local signal plans can be influenced by others), the more weight will have its opinion respecting which local plans to modify and how to adapt them. Still, sometimes the resulting basic coordination does not lead to globally optimal control plans, so the system designer can issue rights or prohibitions for some agents to use control devices in certain ways. By this, he/she modifies the degree of autonomy of some agents, *biasing* the outcome of coordination (i.e. the resulting global signal plan) in a desired direction.

We will introduce the basic terminology and the dynamics of the $TRYSA_2$ coordination mechanism with relation the aforementioned example. In the first place, we need the notion of *consistent global signal plans*. It is defined as a set of local signal plans, one for each agent, that do not show neither physical conflicts nor logical incompatibilities (i.e. semantic conflicts) between them. From the discussions in the previous sections, it can be easily deduced that in our example there are four consistent global signal plans:

- GP_1: (A_1, B_2, C_2)
- GP_2: (A_2, B_1, C_2)
- GP_3: (A_1, B_2, C_3)
- GP_4: (A_2, B_1, C_3)

Each element of the vector defining the global signal plan GP_1 determines a local signal plan for a traffic management agent. So, GP_1 represents the following set of control device states:

- P_1: Congestion at T1, side way recommended
- P_2: Congestion at T1, side way recommended
- P_3: Congestion at T1, side way recommended
- P_3: Congestion at T2, alternative by city center
- P_4: Congestion at T2, alternative by city center
- P_5: Switch off
- P_6: Very slow traffic at T3
- P_7: Very slow traffic at T3, to city centre by E2
- P_8: Very slow traffic ahead
- R: Plan PR3 (always red)

In $TRYSA_2$, the agents' local preferences for their signal plans are expressed in a quantitative way, by assigning a local *utility* value to their plans. $TRYSA_2$ agents use the reduction of traffic excess in their corresponding problem area as a utility measure of their local signal plans. [2] In the example we get:

[2] A more complex utility function that incorporates further additional of the traffic situation may also be used.

- $U_a(A_1) = 450$; $U_a(A_2) = 300$; $U_a(A_3) = 200$;
- $U_b(B_1) = 1000$; $U_b(B_2) = 700$; $U_b(B_3) = 500$;
- $U_c(C_1) = 150$; $U_c(C_2) = 100$; $U_c(C_3) = 80$;

The quality of a consistent global signal plan is determined by the vector that comprises the local utilities of each agents' component plan. The *utility vectors* for the global signal plans in the example are

- $U(GP_1) = (450, 700, 100)$
- $U(GP_2) = (300, 1000, 100)$
- $U(GP_3) = (450, 700, 80)$
- $U(GP_4) = (300, 1000, 80)$

As $TRYSA_2$ agents are autonomous, they are supposed to be *individually rational*. So, they will only agree on a global signal plan, if they are better off than taking a chance alone. If an agent failed to reach an agreement, it risked that its acquaintances jointly try to harm it by setting local signal plans which are incompatible with its own ones. Usually, all local signal plans can be turned down in that way, so in case of disagreement an agent would have to apply the "neutral plan" of doing nothing. Obviously, this does not result in any reduction of its local traffic excess, so an agent's *disagreement utility* is 0. The disagreement utilities of all agents constitute the *disagreement point d*, whose utility in our example is given by the vector

$$U(d) = (0, 0, 0)$$

It can easily be seen that all consistent local signal plans are individually rational, as the utility of any of them exceeds the disagreement utility for each agent. [3]

Still, we would like the outcome of $TRYSA_2$ coordination to be *efficient*. However, efficiency is hard to measure, especially in a decentralized system of autonomous traffic management agents. For instance, measuring the overall quality of a solution simply by the sum of its utilities obviously would not do. Fortunately, the concept of vector domination can be used instead. A global signal plan GP_i *dominates* another plan GP_j, if all agents obtain the same of more utility from GP_i than from GP_j (formally: $U(GP_i) \leqslant U(GP_j)$). In order to be efficient, a consistent global signal plan may not be dominated by any other consistent global plan. In our example, GP_1 and GP_2 are the only undominated plans, as:

$$U(GP_3) \leqslant U(GP_1) \quad \text{and} \quad U(GP_4) \leqslant U(GP_2)$$

Undominated global signal plans are also called *pareto-optimal*. Pareto-optimality assures that no option to improve (or at least: not to deteriorate) an agreement in the eyes of all agents is wasted. With respect to $TRYSA_2$, it states that no chance to reduce traffic excess in some problem area should be left out, as long as this does not have a negative impact on neighboring areas.

Still, as the example shows, there are usually several pareto-optimal plans. To illustrate how structural cooperation in $TRYSA_2$ chooses among them, the concept of a *mixed global signal plan*

[3] We will see that $TRYSA_2$ agents never really end up in disagreement, but that the designer can use modifications of the disagreement utility to bias the outcome of coordination.

is needed. Agents may not only reach an agreement to enact a specific global signal plan, but may also agree on taking a gamble respecting which global plan to set. A mixed global signal plan is determined by a probability distribution over the set of consistent global signal plans. Suppose that there are m consistent global signal plans, GP_1 to GP_m, and that p_i is the probability of choosing GP_i. A mixed global signal plan MGP is a vector:

$$\mathrm{MGP} \equiv (p_1, \ldots, p_m), \quad 0 \leqslant p_i \leqslant 1, \ \sum_{i=1}^{m} p_i = 1.$$

The notion of utility vectors can be extended to this kind of gambles. The *expected* utility of a mixed global signal plan for an agent 'ag' is given by the sum of its utilities from each consistent global signal plan weighed by its probability.

$$U_{\mathrm{ag}}(\mathrm{MGP}) = \sum_{k=1}^{m} p_k U_{\mathrm{ag}}(\mathrm{GP}_k)$$

As before, the agents' utilities from a mixed global signal plan can be comprised in a vector. In the example, suppose a mixed plan in which agents gamble with equal probability between the four consistent global signal plans. Consequently, we get:

$$\mathrm{MPG} \equiv \left(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\right) \quad \text{and} \quad U(\mathrm{MPG}) = (325, 850, 90)$$

In order to choose among mixed global signal plans, besides *individual rationality* and *pareto-optimality*, structural cooperation imposes further requirements for determining the agreement that the autonomous agents will reach. In the first place, it requires a coordination outcome to be *symmetric*, stating that in the beginning no agent has an a priori advantage among others (however, this may change due to modifications of the disagreement point introduced by the designer). Second, it is supposed to be *invariant* under certain types of linear transformations of the utility function. Finally, we would like it to be *independent of irrelevant alternatives*, as this makes sure that the agreement is determined only by efficient (i.e. pareto-optimal) global signal plans. Under these conditions, it has been shown that autonomous (self-interested) agents will agree on the mixed signal plan, that maximizes the product of gains for each agent, compared to the situation of disagreement where every agent tries to cope with its local problems without taking into account its acquaintances. Formally, the agreement is the mixed global signal plan MGP^* that maximizes the function

$$N(\mathrm{MGP}^*) = \prod_{i=1}^{n} (U_i(\mathrm{MGP}^*) - U_i(d))$$

This fact, a fundamental theorem in axiomatic bargaining theory, is called the Nash bargaining solution (Thomson, 1994). It makes sure that the selected mixed global signal plan MGP^* complies with the above conditions. So, it provides a reasonable baseline coordination from which to set out. In our example, the agreement that TRYSA_2 agents will reach is the mixed signal plan MGP^* determined by the probabilities and utilities

$$\mathrm{MPG}^* \equiv \left(\tfrac{1}{3}, \tfrac{2}{3}, 0, 0\right) \quad \text{and} \quad U(\mathrm{MPG}^*) = (350, 900, 100)$$

Still, in certain situations this baseline coordination may not be sufficiently effective. In these cases the designer may issue *normative prescriptions*, that issue prohibitions or grant permission for some agents to use certain control devices. By this, a certain traffic management agent can be favored, as it may become autonomous for executing some of its local signal plans: if all acquaintances are forbidden to use control devices that would make the agent's local signal plan incompatible, then it can use this plan's utility as its disagreement utility. By consequence, that agents' disagreement utility increases. The property of *disagreement point monotonicity* of the Nash solution assures that in this case the agent's preferences will also gain weight in a potential agreement (Thomson, 1994).

In our example, suppose that we grant permission to traffic management agent A, managing the problem area from Trinidad to Collcerola, to enact the local signal plan A_2, which leads to a new disagreement utility vector of

$$U(d') = (200, 0, 0)$$

This should bias the agreement in line with the interest of agent A, which prefers a consistent global signal plans that contains A_1. In fact, this is precisely what happens, as the function N is maximized when GP_1 is enacted with probability one, i.e.

$$\mathrm{MPG}^* \equiv (1, 0, 0, 0) \quad \text{and} \quad U(\mathrm{MPG}^*) = (450, 700, 100)$$

3.3.3.2. Coordination knowledge model. In order to *instrument* this coordination mechanism in TRYSA_2, we now have a closer look at the traffic management agents' knowledge endowment. As described in the previous sections, the different InTRYS KUs for local traffic management are used by TRYSA_2 agents to generate a set of alternative local signal plans for their problem areas. In addition, this knowledge is used to obtain an expectation of the reduction of local traffic excess associated to each local plan, which is used as a measure the plan's local utility. Still, besides the aforementioned domain knowledge, structural cooperation requires each TRYSA_2 agent to be endowed with KUs containing *social* knowledge for coordination.

The *plan interrelation* unit expresses physical relationships between local signal plans and the possible ways of dealing with them in terms of states of control devices. This knowledge is represented by rules, that obey to the following format:

$$[\mathrm{cdev}_1 \ldots, \mathrm{cdev}_n] \Rightarrow [\mathrm{cdev}_m, \ldots, \mathrm{cdev}_j]$$

or

$$[\mathrm{cdev}_1 \ldots, \mathrm{cdev}_n] \Rightarrow [\mathrm{nogood}]$$

The operational semantics of such a rule determines that the control device states of the antecedent can be substituted by those of consequent without any important changes in the effect of the signal plans (e.g. by merging different messages "congestion at A" and "congestion at B" to be displayed on the same VMS into one "congestion at A and B" message). If control devices are merely incompatible, the consequent is the constant *nogood*.

In our example, the initial incompatibility of between agent A's plan A_1 and agent B's plan B_2, due to different uses of panel P_3, can by solved by merging this messages, fact that is expressed in the following way

$$\begin{aligned}&[(\mathrm{p}_3,\ \text{``}\textit{Congestion at } T1,\ \textit{side way recommended}\text{''}),\\ &(\mathrm{p}_3,\ \text{``}\textit{Congestion at } T2,\ \textit{side way recommended}\text{''})] \Rightarrow\\ &[(\mathrm{p}_3, \text{``}\textit{Congestion at } T1 \textit{ and } T2,\ \textit{side way recommended}\text{''})]\end{aligned}$$

Still, most physical incompatibilities cannot be solved in that manner. The first of the following examples of plan interrelation rules states that different recommendations for deviations on panels cannot be merged. The second determines that in the case of regulators, all plans are mutually exclusive:

$$[(P, \text{“}\mathit{Congestion\ at\ T1,\ side\ way\ recommended}\text{”}),$$
$$(P, \text{“}\mathit{Congestion\ at\ T2,\ alternative\ by\ city\ centre}\text{”}),$$
$$\mathit{type}(P) = \text{panel}] \Rightarrow \text{nogood}$$
$$[(\mathrm{R}, \mathrm{Plan1}), (\mathrm{R}, \mathrm{Plan2}), \mathrm{type}(\mathrm{R}) = \text{regulator},\ \mathrm{Plan1} \neq \mathrm{Plan2}] \Rightarrow \text{nogood}$$

Finally, plan interrelation rules can be used to express logical incompatibilities. In our example, if agent C, managing the area A-19 from Mataró to Barcelona, aims at setting signals that make drivers leave the access motorway by E1, then these drivers will join the main lane before T1, and worsen a potential problem at that point. In terms of plan interrelation rules, this is expressed as:

$$\begin{aligned}&[(p_1,\ \mathit{Congestion\ at\ T1,\ side\ way\ recommended}\text{”}),\\&(p_6,\ \mathit{Very\ slow\ traffic\ at\ T3,\ to\ city\ centre\ by\ E1}\text{”})] \Rightarrow \text{nogood}\end{aligned} \tag{1}$$

The above knowledge merely defines relations between states of control devices and, consequently, between signal plans. The agents that may enact these signal plans are not mentioned. Still, a traffic management agent not only needs to know *how* others can influence its local signal plans, but also *who* are the agents that can do so. For this purpose, it is endowed with an *agent dependence* KU, that models these social relations between agents in the shape of rules of the form

$$[\mathit{cdev}_1, \ldots, \mathit{cdev}_n] \Rightarrow [\alpha_1, \ldots, \alpha_m]$$

If all control devices $cdev_1$ to $cdev_n$ switch to new states, then this concerns the agents α_1 to α_m. For instance, if agent α_i may set a message M_i on VMS P, and α_j possibly displays a message M_j on the same panel, while both messages are incompatible, then the knowledge base of α_i's dependence unit will contain a rule stating that setting M_i on VMS P concerns agent α_j.

Consider agent B, managing the ring road from Port Vell to Trinidad. In our example, its use of the panel p_3 and the regulator concern the local signal plans of agent A. So, its agent dependence knowledge base will contain rules such as:

$$[(p_3, \text{“}\mathit{Congestion\ at\ T2, side\ way\ recommended}\text{”})] \Rightarrow [\text{agentA}]$$
$$[(\mathrm{R}, \mathrm{Any}), \mathrm{type}(\mathrm{R}) = \text{regulator} \Rightarrow [\text{agentA}]$$

Note that these rules actually compile knowledge about the capabilities of an agent's acquaintances upon the background of potential plan interrelations. For instance, if agent α_i may set a message M_i on VMS P, and α_j is capable of establishing some incompatible control device state (say displaying a message M_j on the same panel which cannot be merged with M_i), then the knowledge base of α_i's dependence KU will contain a rule stating that setting M_i on VMS P concerns agent α_j. So, agent dependence rules do not explicitly state *what* plans others can enact, but that they are capable of doing *something* that might turn down the agent's local signal plans.

The agent dependence knowledge serves two purposes: when used with forward inference, it allows an agent to deduce which agents are to be informed about its change of local signal plans;

using backward inference, it enables an agent to determine its degree of autonomy, by deducing which agents can enact plans that are incompatible with some of its local signal plans.

3.3.3.3. Coordination algorithm. The coordination knowledge of $TRYSA_2$ agents is put to use within a distributed algorithm, that determines the outcome of coordination based on structural cooperation, by computing directly the Nash bargaining solution, as implied by the agents local signal plan utilities and their interdependencies. Fig. 11 sketches the structure of this coordination algorithm. In stage 1, setting out from the local sets of alternative signal plans, agents repeatedly exchange messages, so as to determine the set of undominated consistent local signal plans. This is done in an asynchronous distributed fashion, which allows for local and temporarily incompatible views of the overall state. The agent that detects the termination of stage 1, takes the initiative in stage 2. On the basis of the outcome of stage 1, it approximates the mixed global signal plan that maximizes the product of the traffic agent's local utilities. So, the outcome of stage 2 is the Nash bargaining solution, i.e. a vector of probabilities with which each consistent global signal plans shall be enacted. Finally, in stage 3 a consistent global signal plan is selected by means of a lottery (in accordance with the result of stage 2), and the agents are informed about the corresponding outcome.

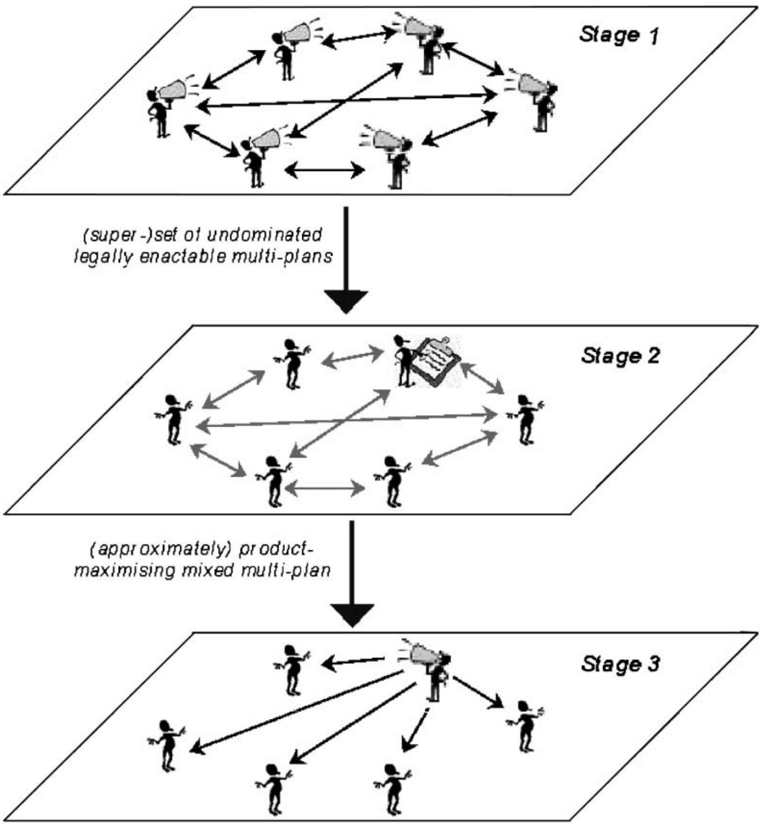

Fig. 11. Sketch of the $TRYSA_2$ coordination algorithm.

The coordination knowledge of $TRYSA_2$ agents is put to use in stage 1 of the algorithm, that determines the set of pareto-optimal consistent global signal plans. $TRYSA_2$ is primarily a decentralized system, so we use an asynchronous, distributed algorithm in stage 1. [4] Our particular approach is based on Yokoo's (1995,1999) *asynchronous weak commitment search*: a dependency-directed backtracking algorithm adapted to constraint satisfaction problems. We have extended this algorithm to cope with the specific characteristics of coordination in $TRYSA_2$, which is rather a constraint *optimization* scenario: plan interrelations constrain the set of solutions (consistent global signal plans), and vector domination on utilities defines a preference order over them.

The algorithm assumes that each $TRYSA_2$ agent maintains a model of its neighboring acquaintances which, among others, stores their "current local signal plan". From the viewpoint of an external observer, the current local signal plans constitutes a global signal plan, and these are generated in order of decreasing preference (respecting utility vector domination), until a consistent global signal plan is found.

To do so, each $TRYSA_2$ agent follows the same simple *agent program*: in every instant of time, an agent chooses from the set of its consistent local signal plans the one that, according to its acquaintances models, results in a current signal plan that is "locally consistent" and most preferred by it. To do this, it relies on the *plan interrelation* KU. If it changes its current local plan, then the agent dependence KU is used to determine which agents are to be informed about this. The agents' attempts to achieve local consistency produce a chain reaction, which eventually leads to a stable state. Local consistency in conjunction with this local choice strategy assures that in stable states, when all agents are waiting for messages, their choices of local signal plans constitute a consistent global signal plan.

A major difficulty in an asynchronously acting agent system is to avoid cycles, i.e. to prevent infinite processing. This is achieved by dynamically generating *nogoods*. Nogoods are partial global signal plans that cannot be extended to a consistent global signal plan involving all agents. As such, they describe new, dynamic constraints: all (partial) global signal plans that constitute a superset of a nogood are not consistent.

We are interested in the potential of finding all undominated global plans. Therefore, once a solution has been found, the search process is restarted until no further solutions are present. For this purpose, previously found solutions are recorded as another type of dynamic constraint. These solution constraints assure that the same consistent signal plan cannot be encountered twice. A detailed description of the algorithm can be found in Ossowski (1999).

3.3.3.4. Normative biasing. The mechanism of structural cooperation within $TRYSA_2$ allows the designer to bias the outcome of agent coordination towards a specific agent. As discussed in the description of the coordination mechanism, for this purpose he or she issues prescriptions, thus altering the disagreement point d and the corresponding disagreement utility. Then, the coordination algorithm described above can be used to determine the modified Nash bargaining solu-

[4] Such algorithms are different from parallel/distributed methods for constraint satisfaction, in that there is an a priori distribution of problem knowledge among asynchronously acting agents, while the latter aim to design a distributed architecture in order to generate solutions more efficiently through parallelism.

tion. By consequence, the lottery among consistent signal plans in stage 3 is realized with modified probabilities.

It remains to be discussed how normative biasing is achieved. In $TRYSA_2$, the *norms* KU deduces prohibitions for some agents to use certain control device or to set certain control device states. Although the set of these normative prescriptions is quite stable over time, they must vary along with important changes in the traffic demand structure. So, the norms KU contains temporally qualified sets of normative prescriptions, which are represented by frames containing two parts:

- *Temporal classification*
  The current date and time is classified in terms of two categories: type of day and type of season. Values for the former are either working day, sunday, saturday or holiday. Rules specify how the value holiday is derived. The category season may be instantiated to Xmas, Easter, Summer, or Normal. Again, the values are related to the current date and time by means of rules;
- *Normative structure*
  As a function of the type of day and the type of season, a normative structure is defined by two categories: prohibitions and permissions. Each of these categories may be instantiated by a set of control devices.

In $TRYSA_2$, the agents' norm knowledge bases are globally consistent: if one agent is allowed to use a control device, all others that might access it are prohibited to use it. The associated reasoning method first classifies the current date and time, in order to match the resulting temporal categories against the normative patterns. By means of this information, the outcome of the distributed algorithm that computes the outcome of structural cooperation is biased.

3.3.3.5. Methodological and implementation issues. The $TRYSA_2$ system is an example of a *constructionist* approach to multiagent system design (Jennings and Campos, 1997). Instead of relying on a top–down design process, the designer builds smaller subsystems that interact by default in a certain way. The integration of the subsystems is to "emerge" from these interactions. In order to tune the integration and to achieve a functionality of sufficient quality, the designer experimentally biases the interactions in a certain direction.

The particular way in which the constructionist approach has guided the design of $TRYSA_2$ is determined by the mechanism of structural cooperation. The design process essentially goes through three major stages.

1. *Individual stage*: design of local problem-solving.
   Problem areas are defined and assigned to agents. Knowledge bases and reasoning capabilities are designed in order to enable the agents to cope with their local problems. No reference to potential interference with other agents is made. The basic traffic control agents of the TRYS family are precisely of this type.
2. *Social stage*: modeling of autonomous agent interaction.
   The possible interdependencies between the agent's local plans are determined, giving rise to the multiple possibilities of conflict and synergy between them. An operational model of self-interested agent behavior is developed to account for these interdependencies. In the $TRYSA_2$

system we have used a distributed search algorithm to determine the Nash bargaining solution implied to the sets of local signal plans of $TRYSA_2$ agents and their expected reduction of local traffic excess.

3. *Normative stage*: design of an instrumental normative bias.
The "equilibrium" that results from autonomous agent interaction need not always correspond to a functional behavior at society level. So, normative prescriptions have to be designed that bias the result of agent interaction in a desired direction. In the $TRYSA_2$ system prescriptions are used to augment the relative importance of a problem area, by giving the associated agent the right to enact certain signal plans.

The problem-solving autonomous agent ($ProsA_2$) architecture has been developed to support the instrumentation of this design strategy. It constitutes the basis of the implementation of the $TRYSA_2$ system. $ProsA_2$ is a vertically layered agent architecture (Müller et al., 1997) reflecting the layering principle the different stages of structural cooperation: each layer can be designed and tested separately in accordance with the design steps outlined above.

The general architecture of a $ProsA_2$ agent is depicted in Fig. 12. It comprises three subsystems. The *perception subsystem* is endowed with perceptors that capture stimuli of the outside world. In the $TRYSA_2$ system it is in charge of perceiving data from the road sensors as well as messages from its acquaintances.

The result of the perception process is passed to the *cognition subsystem*, where the (problem-solving and deontic state) information models are updated. This new information provided by perceptors can bring them into an "inconsistent" state. The three layers of the cognition sub-

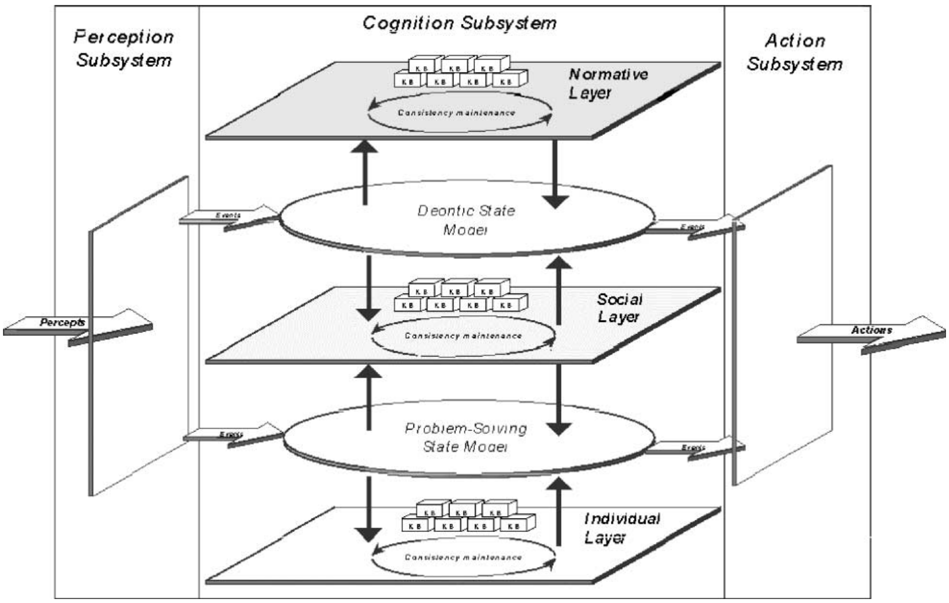

Fig. 12. $ProsA_2$ agent architecture.

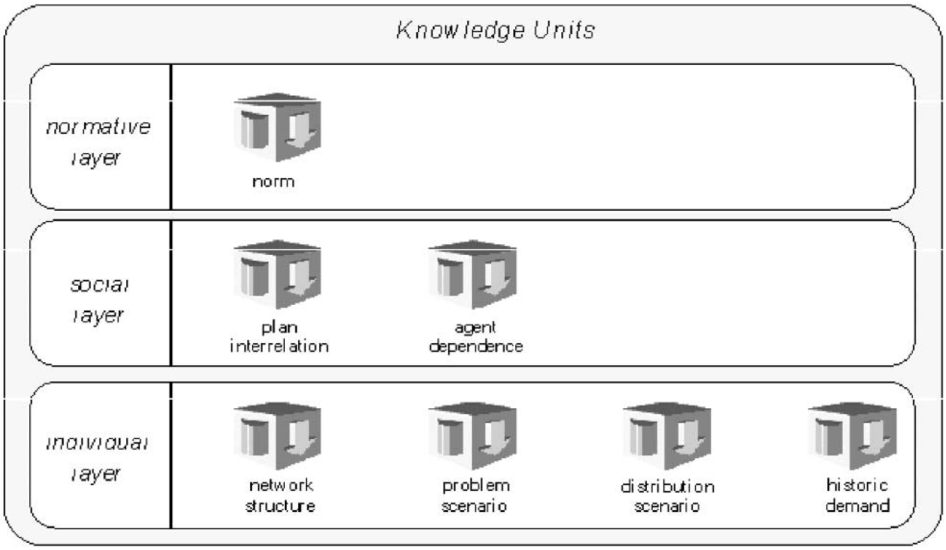

Fig. 13. Knowledge endowment of $TRYSA_2$ agents.

system are in charge of reacting to these changes by restoring model consistency. Based on their particular layer knowledge (see Fig. 13), they all run different instantiations of the same three phase control loop: first, significant changes in the information models are detected (e.g. new data from some sensors); second, the reasons for such inconsistency are determined; third, the adequate model updates are determined (e.g. generating new local signal plans). Each layer is responsible for maintaining the consistency of particular parts of the information models. In $TRYSA_2$ agents, the individual layer generates alternative signal plans on the basis of available traffic data and ranks them so as to maximize their positive impact in the traffic flow of the agent's local problem area. The social layer sets out from the interdependencies between the agents' local signal plans, modifies these local proposals and/or their ranking accordingly, and indicates pertinent messages to be sent. On the basis of contextual information, the normative layer deduces permissions or prohibitions to use certain control devices and, in consequence, to enact certain signal plans.

Finally, the *action subsystem* checks for changes in the information models and manipulates the agent's effectors accordingly. In the case of $TRYSA_2$ agents, it is in charge of sending messages to other agents and of informing about newly proposed road signal plans.

The ProsA agent architecture has been implemented in an object-oriented extension of Prolog. Each $TRYSA_2$ traffic management agent is a specific instance of the $ProsA_2$ agent architecture, endowed with the particular traffic management knowledge of the problem area that it controls. At the implementation level, it constitutes a separate process. So, $TRYSA_2$ is implemented a set of separate Prolog processes (with some extensions in C++), that may run on different networked workstations and communicate via sockets. A special observer agent has been implemented in Tcl/Tk in order to visualize the problem-solving process and its results. Fig. 14 shows a snapshot of the interface window of the observer agent.

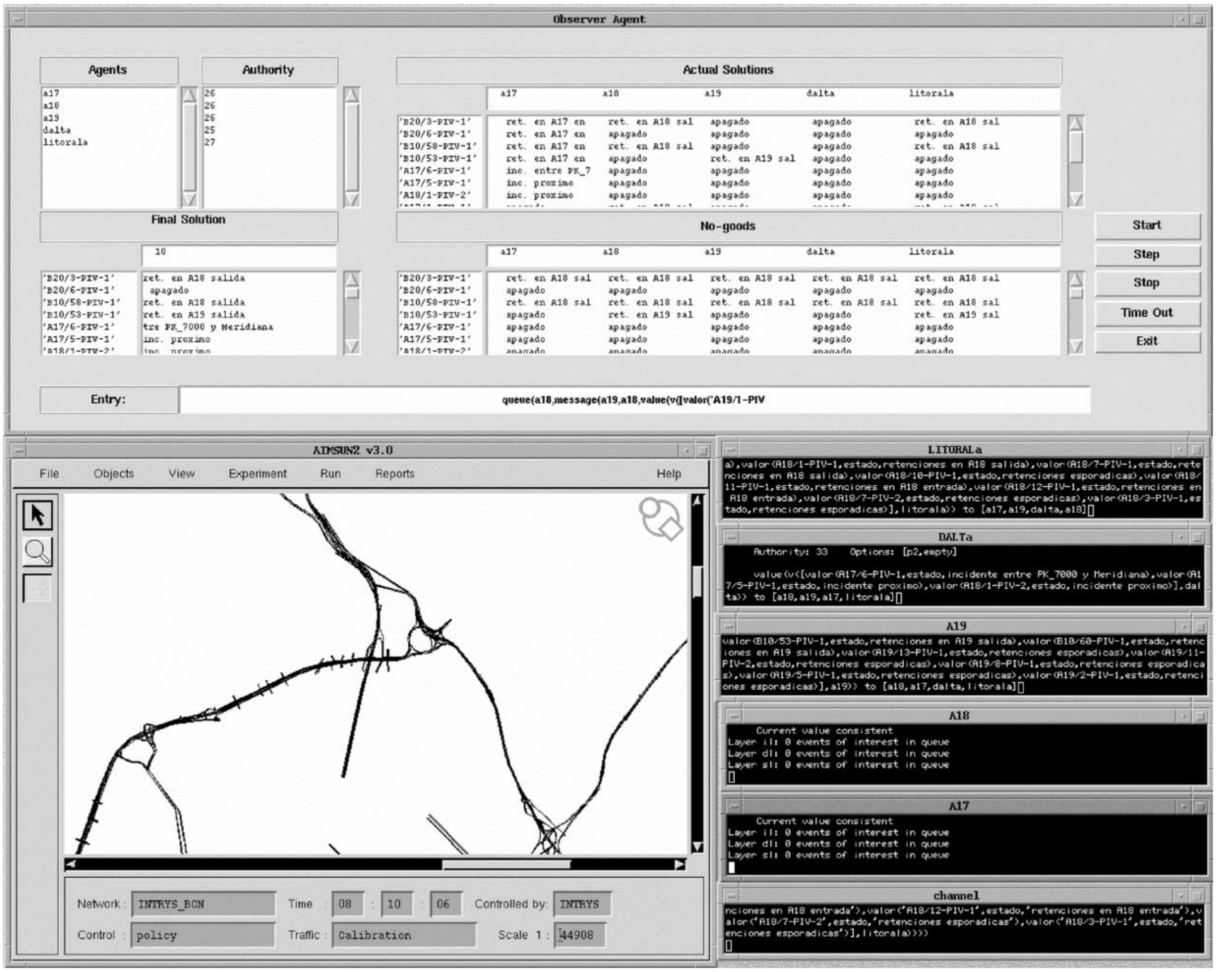

Fig. 14. TRYSA$_2$ interface.

4. Conclusions

In our view, from the work reported here, as well as in the several EU and US research initiatives previously mentioned, the use of artificial intelligence techniques to develop traffic management systems provides a clear added value to conventional systems. From the users point of view, the ITMS may be seen as an intelligent assistant capable to provide useful information for the decision making activity. This utility mostly relies on the fact that ITMS share a common language with the users, i.e. they apply reasoning procedures that resemble those of the traffic operators, delivering conclusions that can be explained in familiar terms for the operators.

In addition, the possibility for the traffic experts to access the knowledge bases and the expression of this knowledge in understandable terms for these operators made feasible and easier the maintenance and improvement of the knowledge model by their own. The possibility of improvement derives both from the personal analysis of the expert and by inspection of the explanations provided by the system for specific decisions during on-line operation.

From the software engineering point of view, additional advantages can be outlined from the experience with InTRYS and $TRYSA_2$ applications. The InTRYS system for Barcelona was developed during one year (four persons) which meant a significant decrease in the required effort compared to the development of previous TRYS-based applications, and in particular to the TRYS model for Madrid (Cuena et al., 1995). This was possible thanks to the possibility of reusing most part of the generic traffic management model previously applied to Madrid and the operational support based on software components. The main activities to develop this system for Barcelona were focused on knowledge elicitation following the KUs approach established for the generic model, together with activities related to software development in order to integrate the knowledge model in the control center. As a result, this experience showed that the use of knowledge models at generic level based on high level structuring concepts (tasks, problem-solving methods, KUs, etc.) was extremely useful in order to increase the productivity and the quality of the final system in the development of complex and large knowledge systems as is the case of decision-support applications for traffic management.

However, the development of modular, agent-based architectures with a considerable volume of different types of information distributed among the architecture components have a coordination cost that has to be considered. Regarding the coordination policies of InTRYS and $TRYSA_2$, the first important difference concerns *synchronisation* issues. The InTRYS coordinator enforces a synchronisation of agent activity during the coordination process. It activates the InTRYS traffic agents by instructing them to produce sets of alternative local signal plans and waits until the last set of local proposals has arrived to start generating a coherent global signal plan. $TRYSA_2$ agents by contrast, do not rely on a strict synchronisation mechanism. Synchronisation is easy to achieve in the case of the current implementation of InTRYS, where all agents are part of the same computational process. However, it may cause problems to centralised systems whose traffic agents are independent, spatially distributed processes.

Strongly related to the above question is the topic of *incrementality*. In each activation cycle, the InTRYS coordination constructs a global signal plan from the scratch. The synchronous and centralised conception does not account for the possibility of adapting an existing global plan to local changes. By contrast, the $TRYSA_2$ policy “reuses” parts of the previous global signal plan: only the necessary adaptations are made incrementally. Obviously, this may considerable reduce the coordination overhead.

Regarding the complexity of the coordination task, the InTRYS approach defeats the decentralized solution of $TRYSA_2$. This is due to the fact that both methods implicitly apply different “search paradigms”. InTRYS combines the locally most preferred signal plan from the set of each traffic agents control proposals. The possibility of solving conflicts within this plan by merging control devices is exploited exhaustively. If some conflict persists, the global plan is rejected and a new one is constructed. Otherwise the signal plan constitutes the coordination outcomes. By contrast, the $TRYSA_2$ strategy may imply an exhaustive search for plans to be selected by the involved agents.

Still, our claim is also that the $TRYSA_2$ approach promotes *scalability*. When a new agent is introduced into the system, agents still need to be informed about its capabilities and, if the newcomer may enact previously unknown signal plans, the interrelation of these plans with existing control actions is to be added to the agents' knowledge bases. Still, the introduction of the new agents produces a shift in the social equilibrium, leading to a new base-line coordination without any further modifications to the agent knowledge. In the centralized approach, however, this effect can only be achieved by completely reconsidering priority relations that a distinguished coordinator agent is endowed with.

Finally, even though the focus of the research in $TRYSA_2$ was on the experimentation with new coordination mechanisms for multiagent systems, some evaluation was performed from a traffic management perspective. An exhaustive evaluation of the benefits obtainable from the InTRYS approach was performed by the authors during the EU project KITS. On-line and laboratory tests were developed with very positive results (KITS, 1995), on the basis of a with/without analysis of the traffic evolution supported the AIMSUN microsimulator (Barceló et al., 1998). AIMSUN simulator was also used to evaluate the performance of the $TRYSA_2$ traffic agents, obtaining very similar responses to those of the InTRYS system. This was not surprising since the traffic knowledge of the $TRYSA_2$ and InTRYS agents was identical, except for the traffic control knowledge, which was reformulated in TRYSA2 according to the requirements of the decentralized approach.

Acknowledgements

The work reported in this article is largely based upon the close work of the authors with Professor José Cuena (1999†), from the Technical University of Madrid, who was a main inspirator and provided the conceptual and scientific foundations of these systems and models. None of the results and achievements presented would have been possible without his continuous contributions, precious suggestions, strong and friendly support and enthusiasm. This work is dedicated to his memory.

References

Auman, R., Hart, S., 1994. In: Handbook of Game Theory, vol. 2. Elsevier, Amsterdam.

Barceló, J., Casas, J., Ferrer, J.L., García, D., 1998. Modelling advanced transport telematic applications with microscopic simulators: the case of AIMSUN. Simulation technology, Science and Art, In: Bargiela, A., Kerckhoffs, E. (Eds.), Proc. 10th European Simulation Symposium, pp. 362–367.

Behrendt, Riegelhuth, Stern, 1997. The ENTERPRICE Project—A step towards information and mobility management. In: Proc. 4th World Congress on Intelligent Transport Systems (no. 2351), Berlin, Germany.

Bielli, M., Ambrosino, G., Marco, B. (Eds.), 1994. Artificial Intelligence Applications to Traffic Engineering. Utrecht, VSP.

Boero, M., Kirschfink, H., Barceló, J., 1997. Real-time traffic management supporting intermobility and strategic control. In: Proc. 4th World Congress on Intelligent Transport Systems, Berlin.

Bolelli, Leighton, Morello, 1994. A top–down approach to open European IRTE architecture: requirements, development paths and assessment methods. In: Proc. 1st World Congress on Applications of Transport Telematics and Intelligent Highway Systems, París, pp. 18–25.

Bond, A., Gasser, L. (Eds.), 1988. Readings in Distributed Artificial Intelligence. Morgan Kaufmann, Los Altos, CA.

Bretherton, Bowen, 1990. Recent enhancements to SCOOT—SCOOT version 2.4. In: Proc. 3rd Int. Conf. on Road Traffic Control/IEE, London.

Chaib-Draa, B., 1998. Industrial applications of distributed AI. In: Huhns and Singh, 1998.

Correra, J., Laresgoiti, I., Cockburn, D., Cross, A., 1994. A cooperative approach towards the solution of complex decision problems in energy management and electricity networks. In: Proc. Int. Conf. On Electricity Distribution.

Cuena, J., Molina, M., 1997. KSM: an environment for design of structured models. In: Tzafestas, S.G. (Ed.), Knowledge Based Systems: Advanced Concepts, Techniques and Applications. World Scientific Publishing Company, Singapore.

Cuena, J., Ambrosino, G., Boero, M., 1992. A general knowledge-based architecture for traffic control: the KITS approach. In: Int. Conf. on Applications of Artificial Intelligence in Transportation Engineering, San Buenaventura, CA, pp. 153–171.

Cuena, J., Hernández, J.Z., Molina, M., 1994. Case presentation of the use of knowledge-based models for traffic management Madrid. In: World Congr. on Applications of Transport Telematics and Intelligent Highway Systems, Paris, pp. 564–571.

Cuena, J., Hernández, J.Z., Molina, M., 1995. Knowledge-based models for adaptive traffic management systems. Transportation Research, Part C 3 (5), 311–337.

Cuena, J., Hernández, J.Z., Molina, M., 1996a. Knowledge oriented design of an application for real time traffic management: The TRYS system. In: Wahlster, W. (Ed.), Eur. Conf. on Artificial Intelligence, ECAI'96. John Wiley and Sons, New York, pp. 308–312.

Cuena, J., Hernández, J.Z., Molina, M., 1996b. An intelligent model of road traffic management in the motorway network around Barcelona. In: Proc. 14th Int. Federation on Information Processing (IFIP) World Computer Congr., IFIP'96. Chapman & Hall, London, pp. 173–180.

Cuena, J., Hernández, J.Z., Molina, M., 1998. Advanced user interfaces for decision support in real time transport management. In: Proc. 5th Int. Conf. on Applications of Advanced Technologies in Transport Engineering, ASCE'98. American Society of Civil Engineering, New York, pp. 67–76.

Decker, K., Lesser, V., 1998. Designing a family of coordination algorithms. In: Huhns and Singh, 1998, pp. 450–457.

Deeter, Ritchie, S.G., 1993. A prototype real-time expert system for surface street traffic management and control. In: Proc. 3rd Int. Conf. on Applications of Advanced Technologies in Transportation Engineering, ASCE'93, Seattle, USA.

Ephrati, E., Pollack, M., Ur, S., 1995. Deriving multi-agent co-ordination through filtering strategies. In: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI-95). Morgan Kaufmann, Los Altos, CA, pp. 679–685.

Fischer, K., Müller, J.P., Pischel, M., 1996. Cooperative transportation scheduling: an application domain for DAI. In: Special Issue on Intelligent AgentsJournal of Applied Artificial Intelligence 10 (1).

Forasté, Scemama, 1986. Sourveillance and congestion traffic control in Paris by expert system. In: IEE 2nd Int. Conf. on Road Traffic Control, London.

Foss, J., 1998. Brookering the info-underworld. In: Jennings and Wooldridge, 1998, pp. 105–123.

Gartner, N.H., 1983. OPAC: a demand responsive strategy for traffic signal control. Transportation Research, Record 906.

Gartner, Stamatiadis, Tarnoff, 1994. Development of a real time traffic adaptive control strategies for IVHS. In:1st World Cong. on Applications of Transport Telematics and Intelligent Highway Systems, Paris, pp. 423–430.

Guttman, R., Moukas, A., Maes, P., 1998. Agent-mediated electronic commerce—A survey. Knowledge Engineering Review 13 (2), 147–159.

Henry, J.J., Farges, J.L., Tuffal, J., 1983. The PRODYN real-time traffic algorithm. In: Proc. of the 4th IFAC/IFORS Conf. on Control in Transportation Systems, Baden Baden.

Hernández, J.Z., Molina, M., 1998. Advanced human–computer interaction for decision support systems using knowledge modelling techniques. In: Proc. 15th Int. Federation on Information Processing (IFIP) World Computer Congr., IFIP'98. IT&KNOWS Conference, Information Technology and Knowledge Systems, Austrian Computer Society, pp. 400–414.

Huhns, M., Singh, P. (Eds.), 1998. Reading in Agents. Morgan Kaufmann.

Hunt, Robertson, Bretherton, Winton, 1981. SCOOT. A traffic responsive method of coordinating signals. Transport and Road Research Laboratory, Report no. LR 1014, TRRL, Crowhome, UK.

Jennings, N., 1994. Cooperation in Industrial Multiagent Systems. World Scientific, Singapore.

Jennings, N., Wooldridge, M. (Eds.), 1998. Agent Technology Foundations, Applications, Markets,. Springer.

Kirschfink, H., 1999. Collective traffic control in motorways. In: 11th EURO-mini Conference on Artificial Intelligence in Transportation Systems and Science and 7th EURO Working Group Meeting on Transportation. Tutorial on Intelligent Traffic Management Models, Helsinki.

KITS: Knowledge Based and Intelligent Traffic Control Systems (V2039), 1995. Evaluation of the KITS Model, Deliverable D18.

Lowrie, 1982. The sydney coordinated adaptive traffic system. Principles, methodology, algorithms. In: IEE Int. Conf. on Road Signalling, London.

Mauro, V., Di Taranto, C., 1989. UTOPIA. In: IFAC/IFIP/IFORS Conf. on Control, Computers and Communications in Transport, Paris.

Molina, M., Logi, F., Ritchie, S.G., Cuena, J., 1995. An architecture integrating symbolic and connectionist models for traffic management center decision support. In: Int. Conf. on Applications of Advanced Technologies in Transportation Engineering, Capri.

Ossowski, S., 1999. Co-ordination in Artificial Agent Societies. LNAI 1535. Springer-Verlag, Berlin.

Ossowski, S., García-Serrano, A., 1999. In: Social Structure as a Computational Co-ordination Mechanism in Societies of Autonomous Problem-solving Agents. Intelligent Agents V. Springer-Verlag, pp. 133–148.

Ossowski, S., Cuena, J., García-Serrano, A., 1998. A case of multiagent decision support: using autonomous agents for urban traffic control. In: Proc. Iberoamerican Conf. of Artificial Intelligence, IBERAMIA-98.

Ritchie, S.G., Prosser, 1990. A real-time expert system approach to freeway incident management. Transportation Research Record 1320, 7–16.

Ritchie, S.G., Stack, R., 1993. A real-time expert system for freeway incident management in Orange County, California. In: Proc. 5th Int. Conf. on Computing in Civil Engineering, Anaheim, USA.

Scemama, G., 1994. CLAIRE: a context-free AI-based supervisor for traffic control. In: Bielli, M., Ambrosino, G., Boero, M. (Eds.), Artificial Intelligence Applications to Traffic Engineering, pp. 137–156.

Steeb, R., Cammarata, S., Hayes-Roth, F., Thorndyke, P., Wesson, R., 1988. Architectures for distributed air traffic control. In: Bond and Gasser, 1988.

Stefik, M., 1995. Knowledge Systems. Morgan Kaufmann, Los Altos, CA.

Thomson, W., 1994. Co-operative models of bargaining. In: Auman and Hart, 1994, pp. 1238–1284.

Von Martial, F., 1992. Co-ordinating Plans of Autonomous Agents. LNAI 610. Springer, Berlin.

Weiss, G., 1999. Multi-agent Systems: A Modern Approach to DAI. MIT Press, Cambridge, MA.

Yokoo, M., 1995. Asynchronous weak commitment search for solving distributed constraint satisfaction problems. In: Proc. Int. Conf. on Principles and Practice of Constraint Programming (CLP-95). LNAI 976. Springer, Berlin, pp. 88–102.

Yokoo, M., Ishida, T., 1999. Search algorithms for agents. In: Weiss, 1999, pp. 165–200.