Agent-based coordination model for designing transportation applications

Fabien Badeig\(^1,2\), Flavien Balbo\(^1,2\), Gérard Scemama\(^1\), Mahdi Zargayouna\(^1,2\)

\(^1\) INRETS - GREThA team, avenue du Général Malleret-Joinville, F-94114 Arcueil Cedex, France. 
\(^2\) University Paris-Dauphine - LAMSADe, Place du Maréchal de Lattre de Tassigny, F-75775 Paris 16 Cedex, France. 
\{badeig,zargayou,scemama\}@inrets.fr balbo@lamsade.dauphine.fr

Abstract—This paper presents an environment-centered approach to design multi-agent solutions to transportation problems. Based on the Property-based Coordination Principle (PbC), the objective of our approach is to solve three recurrent issues in the design of these solutions: the knowledge problem, the space-time dimension and the dynamics of the real environment. To demonstrate the benefits of our approach, two completely different applications, a demand-responsive transportation system and a simulator for crisis management, both based on this principle are presented. For each of them, we show how these recurrent issues have been solved.

I. INTRODUCTION

Research in the field of transportation is not new for the multi-agent community, whose approach deals with systems consisting of many physically and/or logically distributed interacting components that possess some level of autonomy. These components are able to perceive their environment and to react to changes in that environment as well, in accordance with their goals. Focusing on a bottom-up paradigm as an approach to system design makes it easier to comprehend a complex reality, through the reification of the components of the system to be managed. In the transportation domain, the objective of many applications is the management of distributed entities and, in this context, the multi-agent approach simplifies an approach by analogy between an agent and a component.

In our previous developments of transportation applications \([2], [13]\), we have identified three recurrent issues. The first is related to knowledge processing. In many problems such as urban network regulation \([2]\), the knowledge is incomplete or is only known by human experts. The knowledge issue is also related to the coordination of many information sources as in the traffic regulation problem \([8]\). The second issue is the space-time dimension of the problems. For instance, the traveler information systems have to provide the right information and/or services at the right time and the right place \([9], [13]\). The last issue is the dynamics of the real environment, which impacts the quality of the information, makes direct communications difficult and costly and/or implies that mobile entities appear and disappear (e.g. Automatic Guided Vehicles management \([12]\)).

In this paper, we argue that the design of a multi-agent environment compliant with the Property-based Coordination (PbC) principle is a solution to these issues. PbC promotes an effective separation between an observable description of the abstract and real MAS components and the way the agents use these descriptions to coordinate themselves. The multi-agent environment contains the recorded descriptions and supports their processing. This paper describes two applications that are based on an environment modeling that is compliant with the PbC principle. For each of them, we have focused on the solutions to the transportation issues defined above.

The remainder of this paper is organized as follows: section II presents the role of the environment in the design of transportation applications and how it is a support for the PbC principle. Section III details a demand-responsive transportation system. Section IV describes a simulator for crisis management. Section V proposes general conclusions and considers future lines of research.

II. ENVIRONMENT COMPLIANT WITH THE PbC PRINCIPLE

A. Environment role

The environment has responsibilities \([11]\) that have successfully been applied in the design of transportation applications. Since the environment is a shared space for the agents, resources and services, its first responsibility is the structuring of the MAS. Environment modeling is a solution to give a space-time referential to a transportation application. For example, the intelligent transportation information systems or more generally location-based services use the information about the location of the mobile users in order to adapt their processing, and a system is required to manage the relation between the mobile user’s location and the local services. In multi-agent solutions, this task can be achieved by the environment. The environment either centralizes the user agent locations and matches them against the available information/services \([9]\) or is distributed and gives a homogeneous support to the service design \([6]\). Moreover, its privileged intermediary role makes the environment a good candidate to support spatially and temporally decoupled coordination models and therefore it is appropriate for applications taking into account a dynamic real environment. In \([12]\), the environment contains fields that are propagated through it within a certain range and are used by the agents to assign tasks for Automatic Guided Vehicles. The MAS environment becomes the common referential that enables agents to adapt their behavior according to the dynamics of the real environment.
The second responsibility of the environment is to maintain its own dynamics. Following its structuring responsibility, it can also manage the dynamics of the real transportation environment, ensuring the coherence of the MAS. In [12] for instance, the environment ensures the propagation of the fields. Moreover, the environment can ensure services that are not at agent level or can simplify the agent design. In a traffic light control system [3] the environment, with its global view, gives rewards or penalties to self-interested agents according to their local decision.

Because the environment with its own dynamics can control the shared space, its third responsibility is to define rules for the multi-agent system. In a bus network simulation [7], the main role of the environment is to constrain agent perceptions and interactions. For transportation applications that have incomplete knowledge, it simplifies the design of the MAS by a clear separation between the roles of the agents and their organization. In the coordinated monitoring of traffic jams application [5], the environment provides organizations which evolve dynamically according to the current context.

Because the agents are "users" of the services of the environment, and in order to really create common knowledge, the last responsibility of the environment is to make its own structure observable and accessible.

**B. Property-based Coordination principle**

The Property-based Coordination (PbC) principle is based on the observability of the environment. We propose the following definition for PbC: *The function of the Property-based Coordination principle is to design multi-agent components by observable Symbolic Components (SC) and to manage their processing for coordination purposes.*

There are two categories of SC. The first category is the symbolic description (SD) of a real component of the multi-agent system: agent, message, or object. The descriptions are symbolic because they are a representation of the real components. The data structure that has been chosen to give the symbolic description is a set of property-value pairs. For example, an agent has its own process and knowledge, and a corresponding SD is recorded in the environment. Only the descriptions are observable and a control on the SC can be applied by the environment. For example, in the figure 1, the environment contains the SD of two agents, two messages, and three objects.

The second category is related to the abstract MAS components and especially to the coordination component. The objective is to manage the way the agents link the environment content to their behavior. Each link represents the need for an agent or a category of agents and is reified by a Selective Object (SO) (five SOs, figure 1). An SO is a set of constraints on the properties of the SD that defines an agent need. For example, if the SDs of messages or vehicles are recorded in the environment, the following needs could be represented: "I need to send a message to the fastest vehicle" or "when I’m close to a vehicle, I need to slow down".

Managing SC processing concerns the interaction between the agents and the environment. It takes into account the modification of the environment (e.g. the SD update process) by the agents and the way the agents link the environment content to their behavior. The same SO can be used by several agents if they have the same need; in figure 1, SO1 is used for the agents A and B (alternatives 1, 2). These SOs are added (or removed) by the agents or by the designer of the environment. The dynamics of SO management ensures the autonomy of the agents, which adapt their interaction with the environment through management of their SOs. The common process is a matching algorithm that matches the SOs against the SDs. When a SO is triggered, this activates an action (message reception, agent activation (simulation), object perception) for an agent to perform. Figure 1 gives some examples of description combinations. Thanks to the action that is related to the SO, different coordination models can be supported simultaneously. For example, if the coordination is based on a communication model, then the messages have an SD, the SO reifies the interests of the agents for these messages and the matching process applied by the environment gives, for each agent, the messages that are related to it, figure 1 alternatives 1 – 4.

The context is the state of the environment that is composed of the SD, and the behavior of the agents is adapted according to this information. In this way, reification by the SO of agent needs is the starting point for contextual processing of agent behavior. Agents condition their behavior according to their current context including their own state. In figure 1, alternative 6 is an example of contextual activation that takes into account the state of the agents A and B and the object O1.

To comply with the PbC principle, the environment has to give at least the services for the SC management and a matching algorithm. There are no assumptions on the modeling of the SC and the matching algorithm. The two following transportation applications are based on an environment that is compliant with the PbC principle.

**III. COORDINATION ENVIRONMENT FOR DEMAND-RESPONSIVE TRANSPORTATION SYSTEMS**

We have proposed a demand-responsive transportation system (DRTS) as a MAS in which the agents’ activities are coordinated through the environment, following the PbC principle.
A. Demand-Responsive Transportation Systems

A DRTS is a system designed to answer online customers that desire to be transported from one point in the network to another. Customers specify a time window associated with each point (departure and arrival) inside which they want to be visited. The criteria to evaluate the efficiency of a DRTS are, first, to minimize the number of vehicles used to serve all the customers, and then to minimize the total length of the routes.

A DRTS has to deal with the three issues referred to the introduction as follows. First, the knowledge issue is related to the fact that all the customers are unknown before the system is run. In our system, all information is put in common in the environment as soon as it becomes available. Second, the space-time dimension of the problem is important, since the space-time distribution of the customers and the mean width of their time windows condition the behavior of the vehicles and their coordination, and determine the quality of the solutions offered by the system. The environment structures the MAS components temporally and spatially, so that the interaction between agents is driven by their perception of it. The interaction between customers and vehicles is driven by their space-time positions, and the environment is modeled accordingly. The last issue is the dynamics of the environment because when modeled as an MAS, DRTS are open MAS, since agents (e.g. customers and vehicles) join and leave the system freely. In such a dynamic environment, limiting communications is very important, since broadcasting all the available information is very costly. We use a coordination model adhering to PbC, in which communication is decoupled in space and time, so as to offset the loss in information in dynamic environments. More precisely, the MAS environment is a persistent data repository in which data is maintained by the agent (including data representing the agents themselves). This allows for agents to interact without obliging their run times to overlap, or their run location to be the same.

B. System description

We have designed a distributed model for a DRTS, in which two agent categories are defined: Vehicle Agents (VA) and Customer Agents (CA). Both VAs and CAs are generated dynamically: a new CA is associated to each new customer connected to the system, and a new VA is associated to each new vehicle creation (which occurs when no available vehicle can serve a new customer). The SDs in this system are related to VAs and CAs (see figure 2). A VA is described symbolically by its current position and its remaining available seats. A CA is described by its departure and arrival nodes, its time windows, the vehicle veh, and its successor (property *suc)* and predecessor (property *pred*) customer in the route of *veh*. A CA that doesn’t belong to any VA route has a property *veh* equal to unknownveh, and a property *suc* and *pred* equal to unknowncust.

In our application the environment structures the MAS components temporally and spatially, so that the interaction between agents is driven by their perception of it. The SOs are defined so that VAs will perceive only those customers they can insert in their route. The SD of a new CA (with unknown *veh*, *suc* and *pred*) is perceived when: (i) there are two nodes in the route of the same vehicle between which its departure node can be inserted without violating any of the time windows of the customers in that route, (ii) if there are remaining available seats when this insertion occurs, (iii) if there are two successor nodes in the route of this same vehicle between which the arrival node of the customer can be inserted, without violating any time window of the route.

As a consequence, the use of PbC allows VAs to discover a new CA only when they could be interested by its insertion, without knowing it in advance, thus limiting communication in the system to only those agents that can reach an agreement (an insertion in the route of a vehicle). It is worth noting that the VAs that don’t perceive a CA can use their time to be candidates for the insertion of other customers.

The protocol followed in the MAS is a negotiation mechanism between CAs and VAs. When a new customer connects to the system, a CA is created, and is perceived by the available VAs (that is, which are not already involved in the insertion of another customer). Each VA computes an insertion price for the insertion of this customer, and proposes it to the CA, which will choose the VA proposing the lowest price. Our use of this protocol simulates a distributed version of the well-known insertion heuristics [10], which are the fastest heuristics since an insertion decision is not reconsidered afterwards. However, these heuristics are known to be myopic [4], because the computed price is generally function of the additional distance for the vehicle resulting from the insertion of the customer being considered. The future is therefore not taken into account in the price calculation. We have decided to keep the negotiation protocol, but propose a new method to compute the insertion price of a new customer.

C. Insertion price calculation

Instead of focusing on the distance traveled by the VAs in order to compute the insertion price of a customer, we choose to focus on the future availability of the VAs. In this context, the use of a common shared PbC-compliant environment helps us to devise a new method to materialize the VAs’ availability. Remember that the MAS environment in a DRTS can be modeled as a space-time network in which there are not simply nodes, but node-time pairs. We call a perception

\[ \text{Fig. 2. Symbolic descriptions} \]
field of a VA the space-time nodes that it can visit, i.e. a set of nodes it can visit together with the corresponding times, given the customers already inserted in the route of the VA. The insertion of a new customer in the route of a VA implies that there will be several nodes that it won’t be able to visit any more, at least during a certain period, thus reducing the possibility of participating in future insertion negotiations. As a consequence, the wider the perception field of a VA, the more chances it has to be a candidate for the insertion of future customers (in the absence of other information, we assume a uniform distribution of customers).

The notion of a perception field is illustrated in figure 3 with an Euclidean problem, where the network is a plane. The environment of the MAS is illustrated as a cube and represents all the virtually possible space-time positions of all the vehicles in the network. The cone in the left-hand figure represents the perception field of a VA located in \((x_0, y_0)\) at moment \(t_0\). This VA cannot be a candidate for customers that have a departure or an arrival node with a corresponding time that is outside the cone (because, even if the vehicle leaves at \(t_0\), it won’t be able to serve it, given the Euclidean metric). In the right-hand figure, we illustrate the effect on the perception field of a VA of the insertion of a node in its route. The space-time zone that is outside the bold shape but is inside the initial cone is the loss in the perception of the VA resulting from the insertion of this node.

In the negotiation process, the price that is proposed to the CA is the difference between the old perception field (i.e. before the insertion) and the new one (i.e. after the insertion). The main goal of the negotiation process becomes the optimization of the perception fields of the agents, with one objective: the minimization of the number of mobilized vehicles to service all the customers. In [14], preliminary results show that this protocol behaves better than the traditional insertion heuristics with respect to the number of vehicles used, but traditional heuristics behave better with respect to the total distance traveled by all the vehicles. In addition, the experiments demonstrate the pertinence of using the environment to limit communication costs, since the gain in terms of exchanged messages is more than proportional to the number of customers connected to the system. This means that the more agents we have in the MAS, the more network costs we save with respect to a solution based on a broadcast of all the customers to all the vehicles.

In this application, the use of an MAS environment as a materialization of the PbC principle has a twofold benefit. First, it structures agent interaction and coordination, and makes it more efficient to interact in a dynamic environment where agents appear and disappear without maintaining knowledge about the others and where communications can be disturbed, costly, etc. Second, it allows the definition of a new objective: the optimization of agents’ perception fields on the environment, that has been shown to be relevant in the context of DRTS, and is intended to be generalized to certain categories of transportation applications to be defined.

IV. INTERACTIONAL ENVIRONMENT FOR CRISIS MANAGEMENT

In this section we discuss how the PbC principle-compliant environment-centered approach can resolve the difficulties presented above in the case of agent-based simulation of a crisis situation concerning road use.

A. The crisis context

The knowledge related to the management of a crisis is incomplete, because each crisis situation is complex and unique. A crisis situation is a dynamic phenomenon defined by the initial situation, which depends on place and time, and by the impact on population and infrastructure. To deal with traffic crisis situations, traffic management plans have been developed. One plan defines the set of agencies involved (emergency services, the media, traffic control managers) and describes the action protocol for each agency. The efficiency of a plan depends on the organizational efficiency of the agencies involved in the crisis situation. Our proposal is to design a decision-support system for the evaluation and validation of the traffic management plan.

One issue is that the agencies have their own communication and action processes depending on their own goals, and each agency respects a local hierarchical organization that constrains its interaction protocol. Consequently, an agency can be seen as a knowledge source with its own organization and an efficient global organization has to be found. Note that an agency is composed of decision makers and operators. The decision makers adapt the decision-making process to the perception of the crisis situation and the operators implement the actions of the decision makers. The operational level provides information about the situation to enable the decision-making process to build a consistent perception of the situation. Depending on their agency, role and goal, the operators have a selective, i.e. goal-oriented perception of their environment.

From a multi-agent point of view, an agent can be either a cognitive agents (decision makers) or a reactive agent (operators) and they must both be taken into account in the same simulation. Following the PbC principle, each agent continuously updates its symbolic description (SD) in the environment, and its perception is ensured thanks to the selective objects (SO). A detailed description of the symbolic component \(SC\) specifications is available in [1]. PbC is used to reproduce the partial perception of the real environment. Knowledge of the crisis situation is divided into time and space and is available to one or other agency. The difficulty of this incomplete information is its space-time dimension. By definition, the environment is a common space and therefore the environment becomes the space-time referential. For
the space referential, the reactive agents are located in the spatial environment, which ensures the validity of the agent actions. For the time referential, the simulator needs a global clock to synchronize the agents, and the environment ensures the processing of the global time. Consequently, the space-time information is shared between the components of the simulation and appears in the time information is shared between the components of the simulation and appears in the SD of the agents. It is therefore used to compute the context of an agent and to constrain the communication and action of the agents when defining the SOs. For instance, an agent can add an SO that matches the agent context against the SD of a resource in order to perform an action, where the context expresses the constraint about the proximity of the resource to the agent.

The last difficulty is to simulate the dynamics of the real environment in which a crisis evolves and which impacts on the behaviors of the stakeholders. Since the environment is used to constrain the behavior of the agents, it has to manage the simulation process which bring together the services that are not at agent level. Thus, the environment applies the simulation scenario that is being tested (see next section), and the scenario can affect the interactional process. For instance, the designer can specify a failure probability of a communication channel and can easily modify the simulation behavior by changing the definition of the scenario.

B. Functional architecture of the crisis simulator

This section presents the functional architecture of the crisis simulator; figure 4 depicts this architecture where the environment is the space-time referential and ensures the management of the interaction and execution processes.

Our simulator is composed of the simulation components (agents, resources and messages), the environment and four modules. There are two types of agents: cognitive (decision-making level) and reactive (operational level). All the agents are linked to the environment because they take part in the interaction process; for the reactive agent the environment is the space referential with the graph of the transport network. The resources correspond to the safety equipment, to the emergency service vehicles and to the infrastructure. The four modules are related to the processing of the simulation execution (Life Cycle Module (LCM)), to the processing of the traffic management plan (Traffic Management Plan Module (TMPM)), to the communication with a external data source (Interface Module (IM)) and to the observation of the simulation variables (Observation Module (OM)).

The simulator parameters are defined by two inputs and the results of the simulation are given by one output. The first input corresponds to the Traffic Management Plan (TMP) and is uploaded into the TMPM. The second input, which is used to initialize the simulator, defines the initial situation (transportation network, traffic, etc.) and the probability of successful of actions and communications. The output provides a simulation history with the observation of the interactions defined by the designer. It is the evaluation criteria defined by the designer in the initialization file that specify the output results.

A simulation process is controlled by a life cycle that corresponds to an execution scenario. The LCM defines the simulation parameter and is linked to the environment to initialize the variables of the run process. It is initialized by the input initialization file. The TMPM manages the translation of the traffic management plan for the cognitive agent corresponding to the officer in charge of the decision-making level. This module represents a data source for the officer, who can thus adapt the decision-making process according to the TMP and perception of the crisis situation. The module is initialized by the input file TMP. The IM corresponds to an interface between the simulation platform and an external data source, and its role is to communicate and synchronize the spatial environment (i.e. the graph of the transportation network) with a data source. In this way the representation of the transportation network in the simulator is pertinent and the traffic flow is realistic. Our crisis simulator communicates with a macroscopic traffic simulator MAGISTER, a traffic simulation tool built by the french National Institute for Transport and Safety Research (INRETS). It implements several traffic flow models: LWR, LWRACB, METACOR, ARZ. Data is transferred between the interface module of the crisis simulator and MAGISTER using the TCP protocol (Transmission Control Protocol). Therefore the module updates the traffic flow on the transportation network of the spatial environment. The OM is directly connected to the environment; it defines the variables that have to be observed, and the environment sends it the changes of the variable values. The module retrieves data about the running of the scenario execution for a specific crisis situation, and a expert can analyze the results to study a specific point of the simulation.

C. Simulation model specifications

Agent interactions are directed by its local context, which needs a local context computing. This computing complicates modifications to agent behavior and the shift from design to implementation. Our objective is to separate the design process from the simulation process by analyzing the agent context inside the environment. In this way, the environment contains the communication and activation SOs that are the link between a local agent context and an agent action. Consequently this link depends on the local agent context and the
only difference between the activation and communication SO of a agent is the type of action activated by the SO. Interaction processing is unified and makes it easy to put both cognitive agents and reactive agents in the same simulation. Moreover, processing the interactions inside the environment provides a new way to parameterize the simulation.

The environment is the temporal reference for the simulation agents and each agent has its internal time. The environment time is discrete and is the global time of the simulation. At a given time cycle, the simulator ensures that the agents which are ready to act are activated; this is done only once per time cycle. This control is guaranteed by the environment thanks to the comparison of the global time with the internal time of the agent in each activation SO. Therefore, the internal time of agents must be observable which implies the addition of the observable property time to the SD of the agent. When the value of the observable property time is greater than the time of the environment then the agent is not activated. This control implies that the internal time of an agent is updated automatically when it is activated. The activation of the agents depends on the SO inside the environment. In order to ensure a default activation, a default SO is systematically added to the environment and its triggering depends on the comparison between the global time and the internal time of the agent. This modeling offers two advantages: (1) an agent can choose to be inactive for a period in the simulation, thus enabling a gain in runtime; (2) the same agent cannot be activated more than once in the same time cycle. Moreover, it can facilitate complex interaction protocol like overhearing. In this case, SOs are added to the environment to enable the agents to receive messages that are sent by agents close to them.

Based on the PbC principle, the coordination model unifies the communication and activation framework to support the design of complex multi-agent simulations, and it facilitates the change of simulation parameters.

The activity of an agent is modeled using a behavior automaton. Each state of the automaton is a reference to a behavior which is a coherent sequence of actions. The transition from one state to another corresponds to the context. To build the behavior automaton of each agent, the designer first has to identify the activation contexts of each behavior, and then build the behavior library. This library reifies the link between the agent activation and its context thanks to SOs.

An SO becomes active when the agent adds it to the environment, and is deactivated when the agent retracts it. The agent dynamically chooses the interaction SO that it adds to the environment; each additional activation SO triggers a particular activation context for the agent. The same behavior can be activated from various SOs and thus in various contexts. Conversely, the same SO can be used for several behaviors, enabling an agent to modify its reaction to the same context. This approach facilitates the implementation of various scenarios. Figure 5 depicts the relation between the agent behaviors and its activation SO.

V. CONCLUSION

When designing transportation systems, we observed three recurrent difficulties which have to be resolved. Although the multi-agent paradigm facilitates modeling by analogy, it is not sufficient. In this paper, we suggest that using the environment as a common coordination medium is an efficient solution. The environment is thus not only a common space but is part of the management of the system components. We have presented two transportation problems to which the model based on the PbC has been applied: the dial-and-ride problem and crisis simulation. It is a generic coordination model able to answer various transportation problems by focussing on the observability and accessibility of system components. The considered components are not only the system entities, but also their interactions.

REFERENCES