Multiagent Environments for Dynamic Transportation Applications

Mahdi Zargayouna(✉)

Université Paris-Est, IFSTTAR, GRETTIA, Boulevard Newton,
Champs sur Marne, 77447 Marne la Vallée Cedex 2, France
hamza-mahdi.zargayouna@ifsttar.fr

Abstract. Dynamic transportation applications have long been a domain of choice for the multiagent paradigm. Indeed, the presence of distributed entities, the highly dynamic character of these applications and the often presence of human actors in the system makes it very suitable for a multiagent design. This paper advocates for the primary consideration of multiagent environment design when dealing with such dynamic transportation applications. Transportation applications can greatly benefit from the use of the multiagent environment since most of them consider a dynamic geographical positioning of the system components. Indeed, the simultaneous consideration of the time and space dimensions makes the environment, which is shared and accessed by all the agents of the system, a candidate of choice to capture the dynamics of the application. The environment design can be envisioned at several levels of the system construction. It can be used as a medium for interaction between distributed entities. It could be used as a coordination entity of the system components. It can finally be designed as a mental model for the agents that they can use in their reasoning. We illustrate the possible uses of the environment with two transportation applications dealing with traveler information.

1 Introduction

The multiagent paradigm is proven to be a powerful model to design and implement transportation applications. Indeed, the multiagent approach deals with systems consisting of many physically or logically distributed interacting components that possess some level of autonomy. These components are able to perceive their environment and also react to changes in that environment in accordance to their goals. That is why the multiagent approach is adapted to the transportation domain since it facilitates an approach by analogy in a domain where the objective is the management of distributed entities. The authors in [1] list several reasons for the privileged use of multiagent systems in these applications, such as the natural and intuitive problem solving, the ability of autonomous agents for the modeling of heterogeneous systems, the ability to capture complex constraints connecting all problem-solving phases, etc. Indeed, the concept

G. Jezic et al. (eds.), *Agent and Multi-Agent Systems: Technology and Applications*, Smart Innovation, Systems and Technologies 74,
DOI 10.1007/978-3-319-59394-4_2

of an agent is well suited for the representation of travelers in transit or road traffic scenarios [2,3]. They are autonomous entities which are situated in an environment, adapt their behaviors to the dynamics they perceive and interact with others agents in order to achieve specific goals. For Parunak [4], "Agent-based modelling is most appropriate for domains characterized by a high degree of localization and distribution", which is the case for complex and dynamic transportation applications.

In the multiagent community, there is a growing conscience that the multiagent environment should be considered as a primary design abstraction, of equal importance as the agents. Models and architectures have been proposed in the literature for multiagent environments design, validated in a variety of application domains [5]. We believe that one of the domains of choice for the multiagent environment modeling is the transportation domain. Indeed, transportation applications always have some kind of representation for the environment, typically the transportation networks. The environment in transportation application has its own dynamics (e.g. traffic conditions, dynamic rules, weather, etc.), which advocates for its independent and explicit representation. Transportation systems are open, with entities joining and leaving the system (e.g. travelers, drivers, vehicles, regulators, etc.), generally in a nondeterministic way. The multiagent environment can also be the privileged interlocutor of the newcomer entities.

In this paper, we illustrate different design angles of the multiagent environment when dealing with transportation applications. The environment design can be envisioned at several levels of the system construction. It can be used as a medium for interaction between distributed entities. It could be used as a coordination entity of the system components. It can also be designed as a mental model for the agents that they can use in their reasoning. To illustrate the possible uses of the environment, two applications are considered: traveler information and information dissemination in disturbed transit networks.

The remainder of this paper is structured as follows. In Sect. 2, we present a generic design of multiagent environment, in the form of a specification language and the traveler information application built with the language, and using the environment to support agents interaction. In Sect. 3, we present a representation of the multiagent environment that is specific to transportation applications, based on space-time graphs. The chosen application example is information dissemination in disturbed transit networks. Section 4 concludes the paper and provides some future works.

2 Generic Environment Model

In dynamic transportation applications such as advanced traveler information or dial a ride systems, travelers, clients and vehicles join the system in a nondeterministic way, and might leave it anytime as well. When specifying such open systems, the designer has to define an architecture that allows for the integration of unknown agents. Newcomer agents have to be able to find the agents that have the properties, the capabilities or the resources that they need. To deal with this problem, known as the connection problem, the authors in [6] propose the concept of middle agents, who are the privileged interlocutors of agents looking for specific capacities. The author in [7] proposes recommendation systems, enabling the linking of distributed agents in open MAS. This approach allows for the progressive and distributed construction of an address book for the agents. However, in the dynamic transportation systems, the desired capabilities and sources of information are generally known: the transport operators, or the vehicles, or the real-time traffic information providers, etc. The problem is to know which of the information generated by these sources are relevant for the new agents, which context and needs are usually continuously changing. The multi-agent environment is also used for agents matching based on the properties of agents and the exchanged objects [8].

We adopt this environment-centered approach, because it focuses on the shared data and allows for the selection of relevant information without having to know or to maintain knowledge about the emitters of these data. We propose a generic representation of the environment, shared by all the agents of the system, that allows for the associative discovery of the other agents and the exchange of information between them. Agents do not maintain an address book of each others and delegate the matching of their preferences with others properties to the environment. They also can describe the properties of the agents they want to interact with and the messages they want to receive. The presence of a shared environment and the possibility to define complex interaction constraints makes this model an excellent candidate for the design of open and dynamic transportation systems.

2.1 Model

The Fig. 1 illustrates the architecture of a multiagent system (MAS) following our generic model. The modeled MAS executes on a host, where (local) agents add, read and take objects to/from the environment. Every agent is either independent (like agent 1), or representing a non-modeled external system/user in the MAS (like agents 2 and 3).

For the specification of agent behavior, we adopt four primitives inspired by Linda [9] and a set of operators borrowed from Milner's CCS [10].

A MAS written following our generic model is defined by a dynamic set of *agents* interacting with an *environment* - denoted Ω_{ENV}, which is composed of a dynamic set of *objects*. Agents can *perceive* (read only) and/or *receive* (read and take) objects from the environment. Agents are defined by a behavior (a process), a state and a local memory in which they store the data they perceive or receive from the environment. The primitives allowing these actions are the following:

$$\mu ::= add(sds) \mid spawn(P, sds) \mid look(sds_p, sds_r, e) \mid update(sds)$$

The primitive $add(sds)$ adds to the environment an object described by sds. For instance, $add(position \leftarrow 1)$ adds the property-value pair $(position \leftarrow 1)$

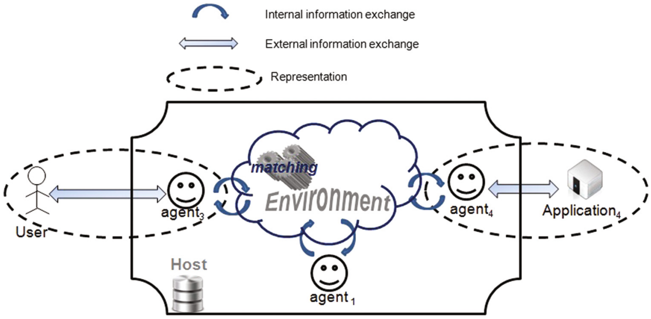

Fig. 1. Architecture of the MAS

to Ω_{ENV}. The primitive $spawn(P, sds)$ launches a new agent that behaves like P and which state is described by a description sds. For instance, $spawn(add(position \leftarrow 1), \{id \leftarrow a_1, position \leftarrow 1\})$ creates an agent that has a_1 as id and 1 as $position$ and whose behavior is $add(position \leftarrow 1)$. The primitive $look(sds_p, sds_r, e)$ allows both object perception and reception (perception and removal from the environment). It blocks until a set of objects becomes present in Ω_{ENV} such that the expression e is evaluated to $true$; the objects associated with the variables in sds_p are perceived and those associated with the variables in sds_r are received. For instance, the following instruction:

$$look(\{ticket \leftarrow t\}, \{paper \leftarrow p\}, t.destination = \text{“}Berlin\text{”} \wedge t.price \leq budget$$
$$\wedge p.decision = \text{“}accepted\text{”})$$

looks for two objects that will be associated with t and p. The object associated with t will be perceived while the object associated with p will be received. After the execution of this instruction, the two objects will be present in the local memory of the caller agent. The latter will have two additional properties: *ticket* that refers to the object associated with the variable t and *paper* that refers to the object associated with p. The perceived *ticket* has “Berlin” as destination and a *price* lower than the budget of the executing agent, while the *received* paper is “accepted” (the property *decision* is equal to “accepted”).

2.2 Environment-Centered System for Traveler Information

In this section, we describe an application based on our model. We modeled and implemented a traveler information server. The purpose of the server is to inform online travelers about the status of the parts of the transportation network that concerns them. Transportation Web services are represented with agents in the server and their properties are related to the service or the information that they provide. The problem in this kind of applications concerns the information flows that are dynamic and asynchronous. Indeed, each information source is hypothetically relevant. An agent cannot know *a priori* which information will interest him, since this depends on his own context, which changes during execution.

The objective in this application is to ensure the information of a traveler about his ongoing trip (disturbances, alerts, alternative itinerary). This process is difficult because the information sources are distributed and the management of the followup assumes a comparison of all the available information. Using our model of the environment for this application allows to design an information server parameterized by its users (the travelers). We have defined two categories of agents, the first concerns the agents representing the users (that we call PTA for Personal Travel Agent) while the second concerns the agents representing the transportation services (that we call Service Agent).

We have implemented a multiagent system running on a Web server for traveler information, where each Web service has a representant in the multiagent environment, which is responsible of the convey of messages from the server to the transportation Web service and conversely. Every user is physically mobile and connects to the server via a transportation assistant app (TAA) and has during his connection a PTA agent representing him inside the server, which is his interlocutor during his session. The context of the example is the following: inside the system, there is an agent representing a trip planning service and an agent representing a traffic service responsible of the emission of messages related to incidents, traffic jams, etc. These agents are persistent, since they are constantly in relation with the system providing the service. On the contrary, PTA agents representing the TAA in the system are volatile, created on the connection of a user and erased at the end of his session i.e. when he arrives at destination.

Every stop of the network is described by a line number *line* to which it belongs, and a number *number* reflecting his position on the line. A user u is also described by his current position in the network (the properties *line* and *number*). In a basic execution scenario, u has a path to follow during his trip i.e. a sequence of tuples $\{(line, number_{source},\ number_{destination})_i \mid i \in I\}$, with I the number of transportation means used by the traveler. Every tuple represents a part of the trip, without transfer. To receive his plan, the TAA connects to the information server, and the agent u representing him is created. Then, the user is asked to specify his departure as well as his destination. Once these information entered, u adds his planning demand in the environment. A demand is an object described by his properties: *emitter*, *subject*, etc. Afterwards, u keeps on listening to messages that are addressed to him, this way: $look(\emptyset, \{message \leftarrow x\}, x.receiver = id)$. The agent representing the trip planning service is listening to messages asking for a plan: $look(\emptyset, \{request \leftarrow x\}, x.subject = \text{“}plan\text{”})$. As soon as he receives the message, he creates a message addressed to the trip planning Web service and awaits for the response. When he receives the answer, a message is added to the environment addressed to u with the received plan as body: $add(\{emitter \leftarrow id, receiver \leftarrow request.emitter, body \leftarrow plan\})$. The agent u, when he receives the message, analyzes it and displays the result on the user's TAA. Then, the agent u restrains his interaction to the messages concerning events coming up on his way. To do so, he executes the following action:

$$look(\emptyset, \{event \leftarrow x\}, \{x.subject = \text{“}alert\text{”}\}, x.line = line \wedge x.number \geq number)$$

The agent u is interested by the alerts concerning his transportation plan, which are expressed by the preceding *look* action. Let us assume that the agent representing the alert service adds an alert message concerning an accident on the way of u resulting on a serious delay for him. The traveler, via his representing agent u, is notified concerning this alert event. Since the properties *line* and *number* are updated (with an *update* action) at each move of u (each time he moves from stop to stop), the segment concerned by the alert messages gets gradually reduced until the end of the trip. The use of the environment, the constant update of the properties of the PTA agents, together with the use of *look* actions allowed us to maintain a constant awareness of the traveler about problems occurring during his trip, without relying on continuous requests to the server.

The proposal of an environment-centered system for traveler information shows how our model allows for the design and implementation of a dynamic and open transportation system. Agents join and leave the system freely and have complex interaction constraints. In this application, the interaction constraints concern the current positions and the itineraries of the travelers.

3 Space-Time Environment for Traveler Information

In this section, we present an approach where the notion of multiagent environment is used in a different way. The environment model presented above is useful for transportation applications, but remains general-purpose, and can be used for any open MAS where interaction is complex and involves several agents at the same time. The environment model that we present in this section is directly inspired and usable for transportation applications, or at least for applications involving mobile entities on a graph, a grid or a plane. The general idea is to propose a space-time representation of the environment, which can either be used as a mental model of the agents or to synchronize agents actions and movements. This representation has been used in the past in different applications: dial a ride, vehicle routing, etc. [11–13].

3.1 Generic Space-Time Model of the Environment

Given a transportation network $G = (V, E)$, with a set of nodes $V = \{(v_i)\}, i = \{0, \ldots, N\}$ and a set of arcs $E = \{(v_i, v_j) | v_i \in V, v_j \in V, v_i \neq v_j\}$. Let two matrices $D = \{(d_{ij})\}$ and $T = \{(t_{ij})\}$ of costs, of dimensions $N \times N$ (the arc (v_i, v_j) has a distance of d_{ij} and a travel time of t_{ij}). The representation of the multiagent environment is made of a duplication of G, H times, with H the maximum allowed time of the considered application: $G(t) = (N(t), E(t))$, with $N(t)$ a set of nodes at time t and $E(t)$ a set of directed links at time t, and $0 \leq t \leq H$. The time copies of G are not necessarily identical (cf. Fig. 2). Indeed, we could have different travel times between two copies to reflect the traffic status. Some nodes can be present in one copy while absent in another to reflect the expansion of a crisis situation. Arcs also can be absent to reflect vehicles timetables in public transport as in the application described in the following section.

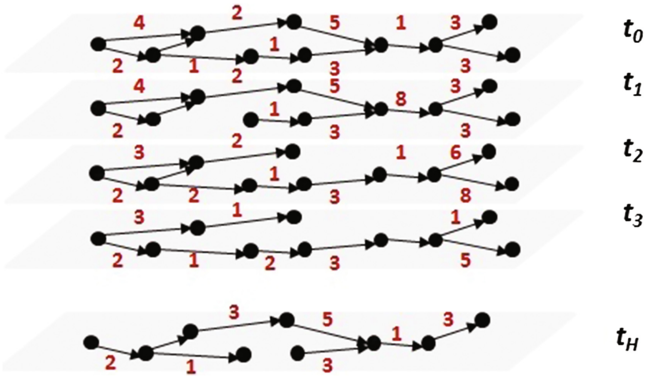

Fig. 2. Space-time network

3.2 Impact of Real-Time Information in Disturbed Transit Networks

Transportation systems are becoming progressively complex as they are increasingly composed of smart and mobile entities. Indeed, passengers mobile devices and connected vehicles allow passengers and vehicles to have up-to-date information and their behavior is now related to these information. However, without control, the massive spread of information via billboards, radio announcements and individual guidance may have perverse effects and create new traffic jams. Indeed, with this generalization of real-time traveler information, the behavior of modern transportation networks becomes harder to analyze and to predict. It is then important to observe these effects to consider the proper methods to deal with them. To this end, we have developed a multiagent simulation platform [14] that represents travelers, drivers and public transportation vehicles and make them evolve in a realistic way on a multimodal transportation network. To allow travelers to receive the only disturbance information that concerns them, we use a space-time network. In the next section, we briefly present the multiagent simulator before presenting our method for information dissemination to the relevant travelers with space-time graphs.

The multiagent simulation platform allows for the individual monitoring of travelers on a transport network. We enrich it with traveler information capabilities, both at the stops and with personal information. The simulator represents

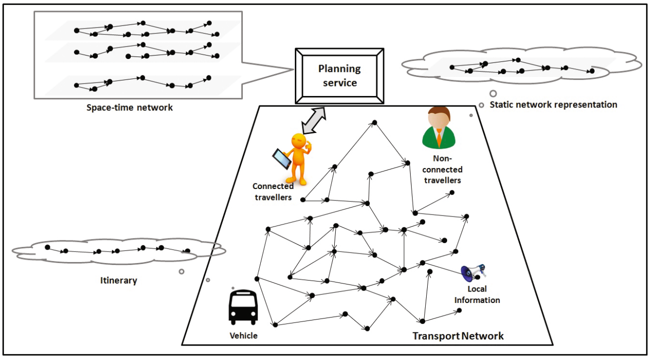

Fig. 3. Multiagent system

itinerary planners, passengers, public transportation vehicles and information means in a micro-level and simulate their dynamic movements (cf. Fig. 3).

The multiagent system of the simulator is composed of the following entities (cf. Fig. 3). The planning service has the responsibility of computing the best itinerary for the traveler agents. The planning service bases his computation on the latest status of the networks. Each active agent has a list of coordinates that he has to follow, resulting from the itinerary that he received from the planning service. At each simulation tick, the agent iteratively move from his current coordinate to the next in his list. The public transportation vehicle agents don't choose their origin and destination and obey to predefined timetables. When the vehicle reaches a stop, he looks in his onboard travelers who has to leave at this stop. Then the vehicle agent looks in the list of waiting travelers at the stop who has to take him. When they are not walking, traveler agents do not travel on their own, but take public transportation vehicles, which are responsible of their movements. The traveler agent alternates between walking and waiting for a vehicle. Local information agents represent devices that broadcast trafic information on screens or voice announcement at the stops. Every traveler, when he passes by this node at the planned time will get an update about the disturbances on the network.

Travelers that are not connected to a real-time information source base their calculation on a static view of the network. Once they have received an itinerary from the planning service, they are on their own. They will wait for vehicles at the planned stops and will not change their itinerary until they either get stuck in a disturbance (delay or line disconnection) or they receive a local information from the environment about a disturbance. When they receive the information, they will infer the new status of the network by applying the modifications to their mental - and static - view of the transportation network, and compute a shortest path based on that representation.

We model the public transportation network as a space-time environment, representing both the network topology and the vehicles timetables. An arc connects two nodes $n_1(t)$ and $n_2(t)$ in $G(t)$ when there is a vehicle departing from n_1 at t. Otherwise, the arc is absent. Space-time arcs are active in this application: they store listeners from the traveler agents and inform them when the departure time or the travel time changes or when the mission of the vehicle is deleted. It is necessary to circumscribe the broadcast of messages to nearby traveler agents. At each concerned node of the space-time network, an information device is associated. To be aware of the only information that concern their stop, local information agents subscribe to space-time edges connected to the concerned stop. To be aware of the only events that concerns him, the connected traveler agents subscribe to the only space-time arcs of the multiagent space-time environment that form his itinerary. When the travel time of an arc or the departure time of the vehicle changes, the information is broadcasted to the subscribing connected travelers. The planning process is then launched with the new status of the network.

Disturbances are modeled exclusively by modifying space-time arcs (that correspond to vehicles timetables). Indeed, delays are injected in the model by dynamically modifying the timetables of the vehicles, adding some time to arrival times. Breakdowns are modeled also by deleting a part of the mission of a vehicle. To model the breakdown of an entire line, the timetables of the remaining vehicles of the line are all deleted. As soon as a timetable is modified, based on the space-time network, the information is immediately detected by the concerned local information devices at stops. The concerned connected travelers will also receive the information immediately. Hence, when a timetable is modified, the information about the delay or the breakdown is sent to the only connected travelers that are interested by these vehicles missions.

4 Conclusion

This paper is based on the conviction that multiagent systems are a suitable paradigm for modeling, simulating and optimizing dynamic transportation applications. It investigates one research question: the explicit modeling of the multiagent environment is it a good choice for these applications? The answer suggested by our work is yes, and we propose two classes of environment models that are interesting for the design of transportation applications. The first class concerns generic environment models for interaction. The second class proposes a space-time model for interaction and is supported by a space-time representation of the environment.

The design of the multiagent environment as an explicit entity is often criticized because it introduces centrality in systems that are supposed to be completely distributed. Following these arguments, centrality could lead to communication bottlenecks, to weak fault tolerance and to poor scalability [15]. However, as we can see it in the models and applications presented in this paper, this architecture has several benefits, and we believe that there is a compromise between the two visions. In our ongoing work, we develop the idea that we still can benefit from an explicit representation of the multiagent environment without loosing the benefits of distribution, namely fault tolerance and scalability.

This is done by splitting the design process in two phases. During the first phase, the system is designed with a conceptually centralized environment. During the second phase, the multiagent environment is distributed. We are working on environment distributions for each type of environments presented in this paper.

References

1. Bazzan, A.L., Klügl, F.: A review on agent-based technology for traffic and transportation. Knowl. Eng. Rev. **29**(03), 375–403 (2014)
2. Bessghaier, N., Zargayouna, M., Balbo, F.: Management of urban parking: an agent-based approach. In: Ramsay, A., Agre, G. (eds.) AIMSA 2012. LNCS, vol. 7557, pp. 276–285. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33185-5_31
3. Bessghaier, N., Zargayouna, M., Balbo, F.: An agent-based community to manage urban parking. Adv. Intell. Soft Comput. **155**, 17–22 (2012)
4. Parunak, H.V.D., Savit, R., Riolo, R.L.: Agent-based modelling vs. equation-based modelling: a case study and users' guide. In: Proceedings of Workshop on Modelling Agent Based Systems (MABS 1998), Paris (1998)
5. Weyns, D., Michel, F. (eds.): E4MAS 2014. LNCS (LNAI), vol. 9068. Springer, Cham (2015)
6. Sycara, K., Wong, H.: A taxonomy of middle-agents for the internet. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000), Washington, DC, USA. IEEE Computer Society, pp. 465–466 (2000)
7. Vercouter, L.: Conception et mise en oeuvre de systèmes multi-agents ouverts et distribués. Ph.D. thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, Université Jean Monnet-Saint-Etienne (2000)
8. Zargayouna, M., Trassy, J.S., Balbo, F.: Property based coordination. In: Euzenat, J., Domingue, J. (eds.) AIMSA 2006. LNCS, vol. 4183, pp. 3–12. Springer, Heidelberg (2006). doi:10.1007/11861461_3
9. Gelernter, D.: Generative communication in linda. ACM Trans. Program. Lang. Syst. **7**(1), 80–112 (1985)
10. Milner, R.: Communication and Concurrency. Prentice-Hall, Upper Saddle River (1989)
11. Zargayouna, M.: Une représentation spatio-temporelle de l'environnement pour le transport à la demande. In: Représentation et raisonnement sur le temps et l'espace, Nice, France (2005)
12. Zargayouna, M., Zeddini, B.: Fleet organization models for online vehicle routing problems. In: Nguyen, N.T. (ed.) Transactions on Computational Collective Intelligence VII. LNCS, vol. 7270, pp. 82–102. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32066-8_4
13. Zargayouna, M., Balbo, F., Scemama, G.: A multi-agent approach for the dynamic VRPTW. In: Proceedings of the International Workshop on Engineering Societies in the Agents World (ESAW 2008), Saint-Etienne. Springer (2008)
14. Zargayouna, M., Zeddini, B., Scemama, G., Othman, A.: Agent-based simulator for travelers multimodal mobility. Front. Artif. Intell. Appl. **252**, 81–90 (2013)
15. Billhardt, H., Fernández, A., Lujak, M., Ossowski, S., Julián, V., Paz, J.F., Hernández, J.Z.: Towards smart open dynamic fleets. In: Rovatsos, M., Vouros, G., Julian, V. (eds.) EUMAS/AT -2015. LNCS, vol. 9571, pp. 410–424. Springer, Cham (2016). doi:10.1007/978-3-319-33509-4_32

http://www.springer.com/978-3-319-59393-7

Agent and Multi-Agent Systems: Technology and Applications
11th KES International Conference, KES-AMSTA 2017 Vilamoura, Algarve, Portugal, June 2017 Proceedings

Jezic, G.; Kusek, M.; Chen-Burger, Y.-H.J.; Howlett, R.J.; Jain, L.C. (Eds.)

2017, XIII, 246 p. 68 illus., Hardcover
ISBN: 978-3-319-59393-7