J Log Lang Inf (2014) 23:219–247
DOI 10.1007/s10849-014-9198-8

A Formal Model of Communication and Context Awareness in Multiagent Systems

Julien Saunier · Flavien Balbo · Suzanne Pinson

Abstract Awareness is a concept that has been frequently studied in the context of Computer Supported Cooperative Work. However, other fields of computer science can benefit from this concept. Recent research in the multi-agent systems field has highlighted the relevance of complex interaction models such as multi-party communication and context awareness for simulation and adaptive systems. In this article, we present a generic interaction model that enables to use these different models in a standardized way. Emerging as a first-order abstraction, the environment, in the sense of a common medium for the agents, is a suitable paradigm to support the agents' awareness. We present an operational model, called Environment as Active Support of Interaction, to take into account all the agents that can be interested in a communication. This model is then extended for the regulation of multiagent systems interactions. Priority policies are given to manage the rules governing the context (un-)awareness of the agents. We also present a new AUML connector to create protocols that take into account the agent awareness to implement proactive behaviour, and several communication scenarios are proposed to show practical applications of this model.

J. Saunier (✉)
Computer Science, Information Processing and Systems Laboratory (LITIS), INSA-Rouen,
Avenue de l'Université - BP8, 76801 Saint-Étienne-du-Rouvray Cedex, France
e-mail: julien.saunier@insa-rouen.fr

F. Balbo
Henri Fayol Institute/ENS Mines Saint-Etienne, 158 Cours Fauriel, 42023 Saint-Étienne Cedex, France
e-mail: flavien.balbo@mines-stetienne.fr

S. Pinson
Laboratoire CNRS-Lamsade Place du Maréchal de Lattre de Tassigny, Université Paris Dauphine,
75775 Paris Cedex 16, France
e-mail: pinson@lamsade.dauphine.fr

Keywords Multi-party communication · Awareness · Environment · MultiAgent systems

1 Introduction

Awareness is a concept that has been frequently studied in the domains of Human-Computer Interaction and of Computer Supported Cooperative Work, see e.g. Markopoulos et al. (2009). Basically, it relates to the human ability to perceive and understand the activities of others. The multiagent systems (MAS) field, as a paradigm based on distributed artificial intelligence, often uses sociological metaphors and concepts because of its very nature : a society of virtual autonomous entities trying to accomplish their goals. However, the environment in which these agents evolve, the computer and network infrastructures, exhibits unique properties which do not allow unpredicted events such as co-presence to be easily perceived. For example, in many multiagent systems, the agents communicate via their addresses, which does not allow to replicate actions such as speaking to the company at large.

Awareness has been shown to be an important efficiency factor in team work (Dugdale et al. 2000). Furthermore, the simulation of human behaviour necessitates to introduce human-like awareness in virtual systems to become plausible. In this article, we focus on communication and context awareness. The basic idea is that in a MAS where agents communicate, these communications may be overheard -voluntarily or not- by other agents. This implies that a particular communication support is available in the virtual environment, which does not rely on point-to-point addressing.

The environment has been quite recently put forward as a first-order abstraction in Multi-Agent Systems (Weyns et al. 2004, 2007), thus introducing many new challenges in terms of modelling, methodology and engineering. Traditionally used in the Situated MAS paradigm, this abstraction is explicitly used to support indirect interactions based for example on stigmergy. For cognitive agents, “in the most part, the environment is an implicit part of the MAS that is often dealt with in an ad hoc way” (Weyns et al. 2004). In the cognitive agents community, few works (Ricci et al. 2007) explicitly present the environment as an interaction support. For direct interaction, the environment is often associated to an infrastructure that supports point to point communication. For indirect interaction, cognitive agents use specific services that are based on the management of a shared collection of data (Carriero et al. 1986; Julien and Roman 2006; Omicini and Zambonelli 1999) that may be understood as a part of the environment. There is therefore a separation between the solutions to realize direct and indirect interaction although the environment provides a suitable framework to unify them (Platon et al. 2005).

In order to enable the agents to be aware of the communications of the others, we have proposed (Saunier and Balbo 2009; Saunier et al. 2007) a unified model called Environment as Active Support of Interaction (EASI), in which the agents actively participate in the definition of their perception. The EASI model provides a support for multi-party communications; and is a suitable framework for the regulation of the interaction. In this article, we draw on this model (1) to show how the agent awareness self-management is improved by our model, (2) to define formally the communication policies in an extension of the EASI model called *Environment as Active Regulator of Interaction*, (3) to propose a new AUML connector to create protocols that take into account the agent awareness to implement proactive behaviour, and (4) to show the use of the model in a real transport application.

Multi-party communication is a communication model that takes into account all the agents that can be involved in a communication. The difficulty is that these participants do not necessarily have the knowledge of the others and/or of their interest for the communication. If the agents have to take into account the context of the MAS, the problem becomes yet harder. In our model, the context is the observable state of the MAS that includes the state of the agents and the current interactions. As in (Julien and Roman 2006), this approach implies that the agents share the ontology of the application domain.

Multi-party communication and context awareness models extend *overhearing*, which is the possibility for a communication to be "heard" by unpredicted agents (Gutnik and Kaminka 2006; Platon et al. 2007; Traum and Rickel 2002) and has been shown to be of importance for cooperative work (Busetta et al. 2002; Dugdale et al. 2000; Kaminka et al. 2002; Legras and Tessier 2004). Overhearing has been implemented either through massive broadcasting (Legras and Tessier 2004), or with the creation of a specific relation between the emitter and the recipients: subscription (Kaminka et al. 2002) and dedicated channels (Busetta et al. 2002). If all agents can participate to all the interactions according to their own interest and to the context, a solution has to be found to avoid an overloading of the system, especially in reliable networks. Within this framework, we propose to add to the environment an algorithm that takes into account the needs of the agents in order to deliver to them the right information at the right time.

This chapter is organized as follows. Section 2 describes the concepts of multi-party communication and context awareness. Section 3 introduces the environment model EASI, and a motivated example of digital city application is given. We extend this model in Sect. 4 to regulate the MAS interaction, with the Environment as Active Regulator of the Interaction (EARI) model. Section 5 integrates awareness in communication protocols, and Sect. 6 shows an application of the awareness model to the transportation domain.

2 Scope of the Interaction Environment

Awareness is about the participants' understanding of the activities of others. In multiagent systems, the agents have two kind of activities, cognition and actions, and only the latter may be directly observable. In our work, we focus on communication, which is a particular case of action.

Many definitions of awareness have been proposed, both general and specific (to tasks or fields of study). A selection of those is found in Drury et al. (2003). Concerning communication, Vertegaal et al. (1997) have proposed to define conversational awareness as knowing *who is communicating with whom* in the context of groupwares. This provides a first approach to communication awareness, but does not capture the possibility to overhear passing messages, and so to reason on their semantics. A more thorough definition of awareness comes from Endsley (1988): situation awareness is *the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future*. In this framework, our interaction environment model deals with the first level of awareness, i.e. providing to the agents the means to perceive the elements in the environment. Comprehension of the meaning of these elements and anticipation, which are parts of the cognitive process of the agent, cannot be attained without this first level.

In the MAS community as well as in the cognitive science community, the most studied type of communication is two-party dialogue. Although for multi-agent systems overhearing has drawn most of the attention, it is in fact a subcase of *multi-party communications* (Branigan 2006), sometimes referred to as *group communications* (Kumar et al. 2000). Enabling multi-party communications is a way to provide communication observability (and hence communication awareness). In this section, the characteristics of multi-party communications are given and the role of context awareness is highlighted.

2.1 Multi-party Communications

Traditionally, communication involves two interlocutors, the speaker that directs the interaction and the addressee. In such dialogues, the interlocutors know each other. Multi-party communication extends this two-party communication: several agents can hear the same message, and they may have different roles in the communication. For example, addressing a warning to a group of students can be heard by passers-by. In this case, the students and the passers-by do not have the same role in the communication act, and this difference affects their reactions. All the agents receiving a message are called *recipients*. A receiver may be *intended* or not, and if it is, it may be expected to take an *active* part in the conversation, or just to hear *passively* what is exchanged. The speaker can be anonymous, and does not necessarily know a priori who will overhear its message, nor their identity, e.g. a public announcement to “every art student”. Finally, hearing a communication may be the result of an initiative of the speaker, for example a multicast decided by the speaker, or of an initiative of the recipient, for example agents reading voluntarily a forum. In the following, we redefine group communications as being a multicast which is not based on the address of the recipients. The key notions to define the role of an agent [speaker and receiver(s)] in a multi-party communication are:

- *Intention*: is the agent expected to take part in the communication, and if so what is its intended role?
- *Knowledge of the identity*: is the agent known by its interlocutors as taking part in the communication?
- *Initiative*: is perception of the communication the result of an action of the speaker or of the receiver?

The different roles that an agent can play in a communication are defined according to these notions (Table 1). A message can be heard by some agents because of the initiative of the speaker, this is typically the case of direct interactions. In indirect

Table 1 Roles of the agents in a multi-party communication

Role	Intention (sender)	Known identity	Initiative
Speaker (sender)	Intended or not, Active	Known or not	Speaker
Addressee	Intended, Active	Known	Speaker
Auditor	Intended, Passive	Known	Speaker
Recipient group	Intended, Active or Passive	Members are not compulsorily known	Speaker
Overhearer	Unintended	Known or not	Overhearer
Eavesdropper	Unintended	Unknown	Eavesdropper

interactions, some other agents can receive the same message because of their own initiative. Here, a difficulty is to determine the actual recipients of a message. As the speaker does not necessarily know all the recipients, this means that it does not control the communication channel fully, as for instance in WiFi communication. For classical networks, this has to be achieved through the use of a particular middleware, or low-level multicast protocols.

Moreover, the initiatives can be expressed by conditions on properties of the environment of the agent (other agents, objects, etc.). For example, group communication implies that the agents can communicate using criteria other than their individual addresses, as when an agent is to be able to contact “the students taking the M201 course”, or “the second-year students in the gymnasium”. Hence, a degree of context awareness has to be supported.

2.2 Context Awareness

Although context awareness has long been considered as an undergone state, it has emerged recently that awareness is an active state, and not only the result of stimuli. Work in the fields of psychology and sociology has discussed whether or not there also has to be an active participation of the “perceiver”. For example, Heath et al. (2002) show that awareness is not only a state of availability to the environment, but that it is also an ability to “filter relevant information which is of particular significance”. Awareness is directed towards a subject via a receptor. Even at the physiological level, Warren (1999) highlights that hearing is a receptive activity that is the result of decisions of action.

Functionally, this means that stimuli coming from the environment can or cannot be ignored. For example, in a physical environment, someone shouting nearby is compulsorily heard. Then, depending on the will of the agent and its ability to do so, the stimuli are perceived or not. For example, someone speaking at a reasonable distance may be overheard if explicitly decided so, but it is impossible if they are too far away. Context awareness is generally studied from the point of view of the perceiver. However, the case of group communication illustrates the case of a speaker taking into account the context. Indeed, directing a message in the sense of choosing the recipients according to properties is a particular kind of focus, which shows an initiative of the speaker.

The control of the interactions in a multi-agent system cannot be delegated fully to the agents themselves: if the messages are processed by the agents, they have to voluntarily comply with the rules of the MAS. Hence, the solution is to design an active environment, which mediates the interactions (Esteva et al. 2004; Zheng et al. 2007). For such an interaction environment, it must be possible to carry out communication even against the will of an agent, and conversely it has to be able to prevent its perception of specific messages. It also has to be able to handle the perception initiatives of the agents. A particular role shown in Table 1 is that of eavesdroppers. Although it may be motivated in a MAS, e.g. for simulation purposes, this is generally an unwanted behaviour. The control of the stimuli by the environment, and especially the ability to prevent a perception, is a way to avoid eavesdropping.

2.3 Illustrative example

To illustrate this chapter, we propose an example of digital city (Benini et al. 2005; van den Besselaar and Beckers 2005; Camarinha-Matos and Afsarmanesh 2002; Hattori et al. 1999), a platform for community network and information space. The goal is to propose a communication infrastructure through a common portal by putting in common the information on the community and its context (a city, university, etc.). Therefore, in opposition to pervasive computing, there is a centralization of the information in the portal.

In a digital city, there are two kinds of participants: *organizations* and *users*. *Organizations* produce information on available sources (such as restaurants and museums) and on events (such as weather forecast and sports); and *users* search for particular data and produce some (such as comments or advertisement).

Information flow may be divided into three parts (Benini et al. 2005) (Fig. 1):

1. *Delivering information to the public.* The information flows one-way, it is produced by publishers (the organizations) and read by users. For example, the city administration publishes its decisions, and the associations publish their activities.
2. *Proposing services.* The information is distributed by publishers, but the objective is to get raw or aggregated data from users. For example, debates about urbanism decisions or polls fall into this category.
3. *Supplying a communication environment.* The users are in turn providing and consuming information. For example, they can announce events which will be commented by other users.

Multi-party communications are relevant for the community support domain because it implies the management of an important number of information sources, which can appear in and disappear from the community dynamically, and users with changing needs.

Currently, portals tend to be limited to only one type of service (Benini et al. 2005) (delivering information, services, or communication environment), although their integration would enable a greater flexibility and personalization. In order to achieve this goal, multi-party communications enable both the information providers and consumers to specify their needs by taking into account all available data.

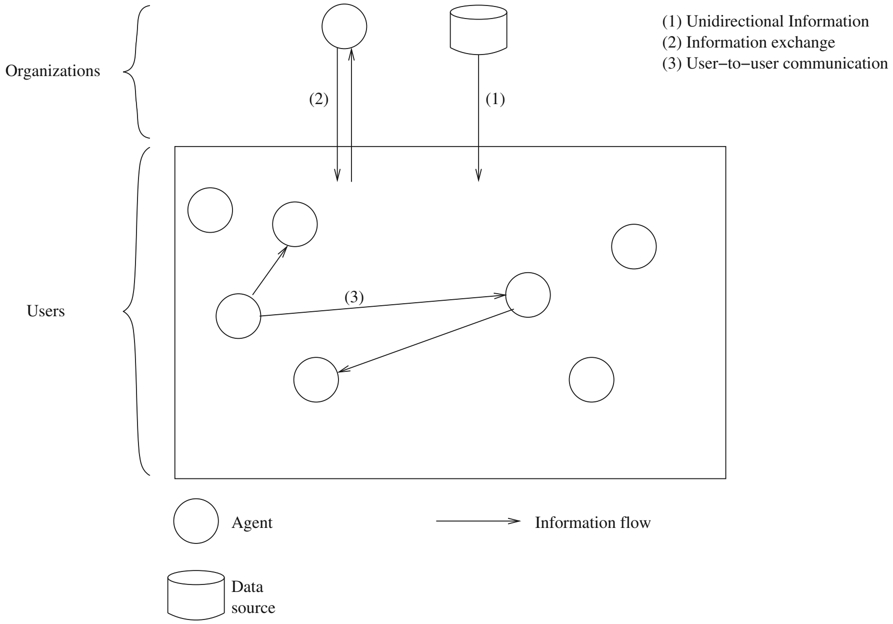


Fig. 1 Information flow in *digital cities*

In this portal example, each user has an agent representing him, and the digital city centralizes the information exchange in a common communication environment. The communications may be between users, with the mediation of their agents, or between agents, in order to carry out specific tasks (such as finding an appropriate schedule for a meeting).

An example of communication emphasizing the particularities of multi-party communications follows:

- The agent a_1 representing an organization needs to send a message to all the users currently available in a particular location, e.g. the library, to inform them of a sudden event.
- The agent a_2, representing a user which is not in this location, needs to get all available information about this place.

When the agent a_1 sends its message, it may not know all the users which fits the characteristics it needs (group communication). Furthermore, the agent a_2 does not belong to this group, but it should nonetheless receive this message because of its particular need.

3 EASI: Environment as Active Support of Interaction

The originality of EASI (Environment as Active Support of Interaction) is to model interaction as a whole and to consider the agents as a part of it. With EASI, communication processing or event/data processing are possible using the same medium, the environment. In both cases, the connection problem (Davis and Smith 1988), i.e. finding where the needed information is and which agents should receive it, is solved by the environment according to its description of the interaction components, agents or percepts. The reification of each connection problem is called a filter. In the following, a connection is the matching between a percept and its receivers.

Awareness is a part of the connection problem. We have seen that the agent situation awareness is driven by foci, which define towards which pieces of information the agent directs its attention. Hence, In EASI, the foci are filters which connect agents and percepts.

In our approach, the environment contains all the required information about the components of the MAS that are clustered in order to find the correct components for each connection efficiently. That is why the EASI formalism is based on the Symbolic Data Analysis (SDA) theory (Bock and Diday 2000). SDA is a clustering model for large qualitative and quantitative data sets. In EASI, modelling the MAS gives the interaction information clustering and the SDA formalization enables the reification of the connection components, i.e. the components of the MAS and the filters.

3.1 Overview

In EASI, the communication environment contains a mirror of the entities of the MAS, via the descriptions of these entities, and tools to manage the communications, called filters. In this section, we give an overview of the basic elements of the model.

Definition 1 (*EASI environment*) The communication environment $\mathcal{E}$ is a tuple $\langle \mathcal{D}, \mathcal{F}, IP \rangle$ where:

- $\mathcal{D}$ is the set of descriptions of the MAS entities,
- $\mathcal{F}$ is the set of communication filters
- IP is the filter priority range, with $IP \subset \mathbb{N}$

The communication environment model consists of a set of m descriptions of entities, $\mathcal{D} = \{d_{\omega_1}, \ldots, d_{\omega_m}\}$, and a set of k filters, $\mathcal{F} = \{f_1, \ldots, f_k\}$. An entity $\omega_l \in \Omega$ is a component of the MAS (agent, percept, object) and has a description d_{ω_l} given by observable properties. A filter f_j is the description of the constraints on the observable properties of the entities that are related to the connection problem j. The filters are defined formally in the next section.

The description of an entity is a set of property/value couples. Let $P = \{p_1, \ldots, p_n\}$ be the set of the n observable properties of the MAS. An observable property p_i is a function that gives for an entity ω_l a value that can be used for the connection, $\forall p_i \in P,\ p_i : \Omega \rightarrow D_i \cup \{unknown, null\}$, with D_i the description domain of p_i. D_i can be quantitative, qualitative or a finite data set. A data set example is given in Table 2. The description d_{ω_1} of the entity ω_1 contains seven property/value couples for p_1 to p_7.

The agents are responsible to update their own description in the environment, but are not allowed to modify the description of the other entities. The descriptions of the messages are generated by the environment when the message is sent by the agent. Thus, there is no concurrent access to properties.

Table 2 Symbolic data table: n individuals and r properties

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	$\dots$	p_r
ω_1	"a1"	"Saunier"	True	30	"M"	"Vincennes"		$\dots$	$p_r(\omega_1)$
ω_2	"a2"	"Balbo"	False	Unknown	"M"	"Arcueil"		$\dots$	$p_r(\omega_2)$
ω_3	"a3"	"Lamsade"					"Laboratory"	$\dots$	$p_r(\omega_3)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
ω_n	$p_1(\omega_n)$	$p_2(\omega_n)$	$p_3(\omega_n)$	$p_4(\omega_n)$	$p_5(\omega_n)$	$p_6(\omega_n)$	$p_7(\omega_n)$	$\dots$	$p_r(\omega_n)$

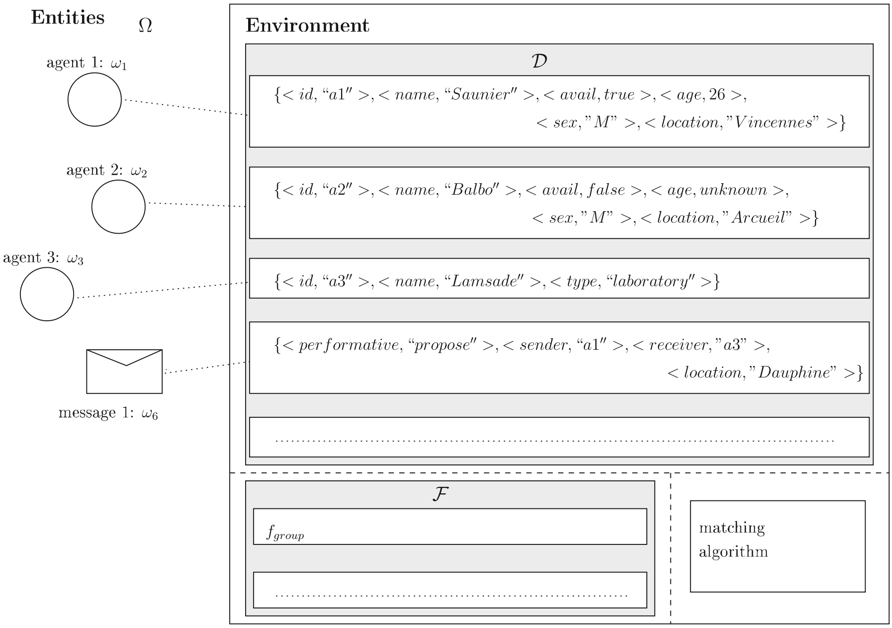


Fig. 2 EASI interaction modelling example

Figure 2 shows an instantiation of our environment modelling for the digital city example. Here there are four entities, $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ that are respectively the description of the *user* agents ω_1 and ω_2, the description of the organization agent ω_3 and the description of the message ω_4. Empty entries are *null* values. The agents have a property called *availability* (avail) and $D_{availability}$ is $\{true, false\}$. The value $location(\omega_i)$ is an area of the digital city; the value $age(\omega_2)$ is $unknown$ because the value has not been given; the value $age(\omega_3)$ is $null$ because ω_3 does not have this property in its description. The value of a property can be modified at run-time by the agents, except if it is $null$. A $null$ value expresses the absence of a property.

An agent that has to solve a connection problem puts in the environment a *filter* that describes this connection problem. For example, the agent a_1 needs to send a message to all the users currently available in the location "Dauphine", to inform them of a sudden event. It puts a filter f_{group} to manage its future message, and then the description $d\omega_4$ of the message m_1 (the message itself is stored in another part of the system). Using the set of descriptions of the components of the MAS (D) and $\mathcal{F}$, the matching process described below finds all the recipients of ω_4 .

In the EASI model, only what is useful for interaction—from the viewpoint of the environment—is taken into account. This can be seen as the public part of the agents, which is independent of their private part (such as their knowledge or their internal architecture). This choice implies that agents having identical descriptions can receive the same kind of messages. Furthermore, there is no conceptual difference between different kind of interaction objects (messages; or objects, such as a trace), as each of them is known only according to its description set. Thus, an interaction is defined in a generic way.

3.2 Transmission Constraints

A category of entities is a set of entities that are described by the same set of properties. Let A be the set of agents that are described by the subset of properties $P_A \subset P$ and let M be the set of the interaction objects (messages, signals…) that are described by $P_M \subset P$. The entities are clustered using the existence condition of required properties. The agent that initiates a connection is looking for particular recipients in a particular context. That means that entities related to this connection have to respect constraints, and that this set of entities can be computed. The function e is an elementary test:

$$\omega_l \in \Omega,\ p_i \in P,\ e : \Omega \rightarrow \{true, false\},\ e(\omega_l) = [p_i(\omega_l) R_e v_e] \qquad (1)$$

R_e may be any binary relational operator,[1] p_i is a property and v_e is a value that belongs to the description domain D_i of p_i. $e(\omega)$ is *true* if the test holds and *false* if not. For example, $\omega_l \in \Omega, e_{loc}(\omega_l) = [location(\omega_l) \in labs]$ (with $labs$ the set of laboratories) is a test related to the location of an entity. For $\omega_l \in A$, the test is *true* if ω_l is in a laboratory and *false* if not. The $null$ value is not possible because the tests are done on entities having the required properties (Definition 2).

The constraints on the description of an entity are called an assertion. An assertion is a conjunction of elementary tests:

$$\forall \omega_l \in \Omega \quad as(\omega_l) = \wedge_{i=1,\ldots,n} [p_i(\omega_l) R_i v_i] \qquad (2)$$

Let us note that v_i is either a value (belonging to the description domain of p_i) or a variable used for a matching between the values of two properties. The extent of an assertion in Ω contains all the entities whose description satisfy all the elementary tests.

Remember that a filter is a reification of a connection, this is why it is a symbolic object which describes the entities that are related to a connection problem, which means at least one agent and one percept.

---

[1] Such as =, > or ⊂, more details on symbolic data operators can be found in Bock and Diday (2000).

Definition 2 (*Filter*) A filter $f \in \mathcal{F}$ is a tuple $\langle f_a, f_m, [f_c], n, [priority], initiator\rangle$ where:

- f_a is the description of the receiver agent, such that: $a \in A,\ f_a(a) = \wedge_{p_i \in P_{f_a}} [p_i(a) R^a_{p_i} d^a_{p_i}]$.
- f_m is the description of the percept, such that: $\omega \in \Omega,\ f_m(\omega) = \wedge_{p_i \in P_{f_m}} [p_i(\omega) R^m_{p_i} d^m_{p_i}]$.
- f_c is the optional description of the context, such that: $C \subset \Omega, f_c(C) = \wedge_{c \in C}\ as_c(c)$, with $as_c(c) = \wedge_{p_i \in P_c} [p_i(c)\ R^c_{p_i}\ d^c_{p_i}]$.
- n is the name of the filter.
- $priority$ is the priority of the filter.
- $initiator$ is the initiator of the filter.

In Definition 2, the receiver description is based on a subset of properties $P_{f_a} \subset P_A$ which contains the properties that an agent must have to be a potential receiver. Using R^a, which is a tuple of comparison operators, and d^a, which is a tuple of reference values and variables, the assertion f_a describes the conditions to become a receiver. In the same way, the description of the percept is given by the assertion f_m. The context of the interaction is given by the symbolic object[2] f_c. From this viewpoint, a context is a subset of Ω and therefore a part of the observable state of the MAS. In order to describe an interaction, the first two assertions f_a and f_m are compulsory, since there must be a description of at least which message(s) and which agent(s) are concerned by the filter.

Let $priority_f \in \mathbb{N}$ be the priority value given by the initiator agent to the filter $f \in \mathcal{F}$. This value determines the order of the evaluation of the filters in the environment. Section 4.4 shows how this information is used to solve conflicts between filters.

Let $initiator_f$ be the identifier of the entity that puts the filter $f \in \mathcal{F}$ in the environment. In EASI, the filters dedicated to the management of the MAS belong to the environment. These filters are put either by a group of system agents, or by a mechanism internal to the environment. Thus, the filters are partitioned in two categories, depending on their initiator: $\mathcal{F} = \mathcal{F}_E \cup \mathcal{F}_A$, with $\mathcal{F}_E$ the set of filters put by or on behalf of the environment and $\mathcal{F}_A$ the set of filters put by the agents.

In this way, the environment can add percepts to the agents, i.e. they will receive percepts that would not have been received otherwise. It is thus possible to distinguish a new kind of auditor in a multi-party communication, for the case of an agent that perceives a communication because of an "initiative" (rule) of the environment. $\mathcal{F}_E$ contains the filters that are the rules of the environment, for example standard transmission behaviours for certain kinds of messages. The advantages of this approach are (i) to enable the existence of the rules inside the environment itself, and not external to it (by monitoring, for example), which enables the regulation of the MAS by controlling actions in real-time instead of doing it a posteriori and (ii) to unburden the agents of this task.

In the digital city, an example is the filter for group communication, which connects a message to all available agents in an area:

---

[2] Formally, f_c is a horde, i.e. a conjunction of assertions containing tests on several entities descriptions (Diday 1991).

$a \in A, m \in M, f_{group} = \langle [avail(a) = true] \wedge [location(a) =?x], [location(m) =?x], \emptyset, \text{“}group'', 0, environment\rangle$

This filter describes the condition related to the agents ($[avail(a) = true] \wedge [location(a) =?x]$), and to the percept ($[location(m) =?x]$). $?x$ is a variable. The value of the constraints on the context f_c is $\emptyset$ (default value) because this filter does not depend on a specific context, hence its extension contains all possible contexts. The default value of the priority is 0. In the following, default values will be omitted in order to simplify the syntax of the filters. f_{group} is a filter of the environment, it defines the way a message sent to users according to their location is handled. It is a group communication because the speaker does not need to know the addresses of the agents, and this example shows basic context awareness in the sense that the perception depends on the value of the properties $avail$ and $location$ of the agent, which can change dynamically.

Another example of filter is $f_{indirect}$. The transmission of the percept is data-driven if the speaker does not choose a recipient:

$f_{indirect} = \langle [id(a) = a_x], [type(m) = \text{“}concert''] \wedge [location(m) =?x], [location(c) =?x] \wedge [type(c) = \text{“}restaurant''], \text{“}indirect'', 0, a_x\rangle$

In this case, an agent a_x will receive all the messages related to concerts announcements ($[type(m) = \text{“}concert'']$), if and only if these concerts happen in a restaurant ($[location(m) =?x]$, $[location(c) =?x] \wedge [type(c) = \text{“}restaurant'']$). It takes into account the description of an organization in the environment as the context of this transmission.

3.3 Management of Communications

The matching algorithm is based on the following relation:

$$V : A \times M \times P(\Omega) \times \mathcal{F} \to \{true, \; false\}, V(a, m, C, f) = f_a(a) \wedge f_m(m) \wedge f_c(C) \tag{3}$$

A filter $f \in \mathcal{F}$ is valid for an agent $a \in A$, a percept $m \in \Omega$, and in the context $C \subset \Omega$ if the value of $V(a, m, C, f)$ is *true*.

Each time there is a connection, a recipient receives a percept. If $V(a, m, C, f)$ is valid, the message m and possibly context information are transmitted to a. This set called *perception set* is noted Per_f:

$$C' \subset C, Per_f = \{m, C', n_f\} \tag{4}$$

Allowing the agents to perceive a part of the transmission context implies that they do not have to reconstruct it a posteriori.

Algorithmically, the perception act is realized thanks to the primitive $receive$ (a, Per_f), which means the perception of Per_f by the agent a. We do not make any assumption on the internal architecture of the agents. We only consider the agent a has a Private Knowledge noted PK_a. The primitive is then:

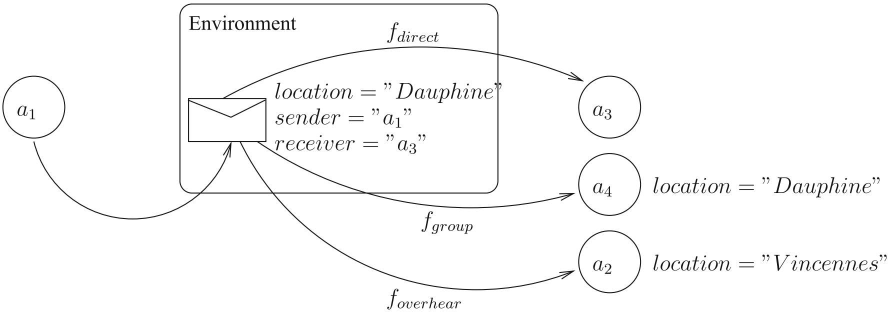


Fig. 3 Example of multi-party communication

$$receive(a, Per_f) \Leftrightarrow PK_a \leftarrow PK_a \cup Per_f \tag{5}$$

We have seen that multi-party communication involve initiatives of the sender (also called speaker, see Table 1), of the receivers and of the environment. Here, each initiative is represented by a filter, whatever the role of their initiator in the communication (Table 1) is. Hence, once a message has been matched with all the filters, all the initiatives have been taken into account, which enables EASI to be generic towards the interaction models, in a standardized way.

The EASI model enables the features highlighted in Sect. 2. In our scenario, we have seen that some agents can send a message to users situated in a location; and that these messages may interest other agents. Furthermore, direct communications should be allowed. To achieve this, let us detail the three corresponding filters, f_{direct}, f_{group} and f_{aware} (Fig. 3). The message description is the following:

$\{< location, “Dauphine'' >, < sender, “a1'' >, < receiver, “a3'' >\}$

The message is sent by the agent a_1 to the agent a_3, thanks to the filter f_{direct}:

$f_{direct} = \langle [id(a) = ?x], [receiver(m) = ?x], “direct'', 20, environment\rangle$

This filter is a matching between the property id of the agents and the property $receiver$ of the messages. Hence, the agent a_3 is an addressee: this is a direct interaction. However, the message also has a $location$ property, which means that it should be sent to all the available agents in this location, through the filter f_{group} detailed above. These agents (in the figure a_4) are in this case a group of auditors: the initiative is that of the speaker and the recipients are intended, but they do not take an active part in the current dialogue.

Finally, the agent a_2 is not situated in the location, but wants to receive all the messages related to this location. Its property $location$ being not set to “Dauphine”, the filter f_{group} does not transmit the message to it. Hence, it will have to put a new filter using the $location$ property of the message:

$f_{aware} = \langle [id(a) = “a2''], [location(m) = “Dauphine''], “aware'', 0, a_2\rangle$

Hence, every message sent to the agents situated in “Dauphine” is received by a_2, which is an overhearer. As soon as a_2 is occupied, it can remove the filter from the environment to stop these messages reception.

Thanks to these filters, the multi-party communication has been enabled by the conjunction of the initiatives of the speaker and of the recipients, allowing several roles to exist for the same interaction object. A filter added by an agent is an instantiation of a focus, which enables the agent to be aware of ongoing activities in the MAS. In this way, the set of filters that concern an agent defines its context awareness.

4 Environment and Rules

The model we have described so far enables agents to act on their own perception channels to match their needs. The availability of all the information is necessary to support mutual awareness and is acceptable for cooperative agents, but compliance with the potential rules of the environment is entrusted to the agents. This is not sufficient for systems whose designers do not control every component. In the digital city example, every agent can hear every message, although it is not acceptable in every situation: users may want to exchange private messages, for example. In this section, we complete the model in order to deal with the issue of regulation.

The concepts were first developed in Saunier and Balbo (2009), and we propose here a formalization of this model extension.

4.1 EARI: Environment as Active Regulator of Interaction

The extension of the EASI model to regulate the interaction is called EARI, *Environment as Active Regulator of Interaction*. The new communication environment definition is:

Definition 3 (*EARI environment*) The communication environment $\mathcal{E}$ is a tuple $\langle \mathcal{D}, \mathcal{F}, IP, \mathcal{R} \rangle$ where:

- $\mathcal{D}$ is the set of descriptions of the MAS entities,
- $\mathcal{F}$ is the set of communication filters
- IP is the filter priority range, with $IP \subset \mathbb{N}$
- $\mathcal{R}$ is the priority policy

$\mathcal{D}$, $\mathcal{F}$ and IP are the same as in EASI, $\mathcal{R}$ is an application which defines the priority policy of the MAS environment. We define $\mathcal{R}$ formally further in the section.

In EARI, we also introduce two new elements: negative filters manage perception blocking, i.e. explicitly preventing message transmission; and a filter triggering algorithm manages concurrent "traditional" and negative filters.

4.2 Negative Filters

The filters put by the environment trigger perceptions, even if the recipient agents do not intend to. To support the opposite case – the environment blocking perceptions –, we must be able to put *negative filters*, noted f_N: when a negative filter is valid, the related percepts cannot be perceived by the related agent. The set of negative filters is noted $\mathcal{F}_N \subset \mathcal{F}$. Negative filters can be put by the environment and by the agents.

Algorithm 1 Matching Algorithm

$Filters(a, io, C)$: list of the elements f of $FPer_a \cap Channel_m \cap FContext_C$, ordered by decreasing priority
$next$: boolean

```
ForEach ($m \in M$)
  ForEach ($a \in Receiver_m$)
    $next \leftarrow true$
    ForEach ($C \in Context_m$), Additional loop condition: $next = true$
      For $f$ in $Filters(a, m, C)$, Additional loop condition: $next = true$
        If ($V(a, io, C, f)$) Then
          If ($f \in \mathcal{F}_N$) Then
            $next \leftarrow false$
          Else
            $receive(a, Per_f)$
          End If
        End If
      End For
    End For
  End For
End For
```

The environment, for its regulation task, has to be able to prevent the perception of interaction objects, *e.g.* for security reasons or to ensure a ban. For the agents, the negative filters enable them to control and limit voluntarily their awareness according to their availability, will or needs. We note the kind of filter with a superscript, *e.g.* f^{+}_{direct}.

The effects of negative and positive filters are contradictory. This problem is solved thanks to the priority of the filters. The filters are processed in decreasing priority order, and once a negative filter is valid for a couple agent/percept, the matching algorithm (Algorithm 1) stops the processing for this couple and tries the next agent. This algorithm handles the percepts each time one is put in the environment.

The sets of agents, percepts, contexts and filters used in the algorithm are based on the extent of the filters. The extent of a filter f is the following tuple of sets $\langle E(P_{f_a}), E(P_{f_m}), E(P_{f_c})\rangle$. Each of these sets contains the entities that have a value for the properties that are required for the evaluation of the related part of the filter. For example, $E(P_{f_a}) = \{a \in A | \forall p_i \in P_{f_a}, p_i(a) \neq null\}$. These sets are computable for each MAS description. The perception domain of a filter indicates whether or not the agents and the messages have the observable properties required by the filter definition, but it does not verify the properties values.

A percept may be received by multiple agents using the same filter, and a percept may be received by multiple agents using multiple filters. The difficulty is to find for a percept m the smallest set containing all the potential receivers, by using the set of filters that are valid for this percept. Scanning smaller sets reduces the algorithm execution time.

The sets used in the algorithm are the following: $Channel_m$ is the set of filters that are related to the message m such as $Channel_m = \{f \in \mathcal{F} | m \in E(P_{f_m})\}$. For each filter f in $Channel_m$ the set of potential receivers and the set of contexts are computable. $Receiver_m$ is the set of the potential receivers of the percept io such as $Receiver_m = \{a \in A | \exists f \in Channel_m,\ a \in E(P_{f_a})\}$. $FPer_a$ (for filter of Perception) is the set of filters that are related to the agent a such as $FPer_a = \{f \in \mathcal{F} | a \in E(P_{f_a})\}$. In the same way, $Context_m$ is the set of contexts in which the percept can potentially be transmitted such as $Context_m = \{C \subset \Omega | \exists f \in Channel_m,\ C \in E(P_{f_c})\} \cup \{\emptyset\}$, and $FContext_C$ is the set of filters related to a context C such as $FContext_C = \{f \in \mathcal{F} | C \in E(P_{f_c})\}$. More information on the rationale and use of these sets is found in Saunier and Balbo (2007).

4.3 Policies Definition

We call *policy* the management of the priorities. A policy determines the priority interval which can be given to the filters, and thus the order in which the filters are treated by the algorithm. Policies are defined in the environment with the function:

Definition 4 $\mathcal{R}$ is a function from $\mathcal{F}$ to $I(IP)$, with $I(IP)$ the set of intervals in IP, such as $\mathcal{R}(f) = [min, max]$

The function $\mathcal{R}$ associates to each filter f a priority interval $[min, max]$ in the set of accepted priorities in the environment definition IP. Any priority in $[min, max]$ is considered valid. The policy is verified at run-time by the environment. When an agents tries to add a new filter with a given priority, the environment verifies its validity thanks to $\mathcal{R}$. In the case of a non valid priority, the filter is not added to the filter set.

Let us note that the policies are decided by the environment designer, and should be public.

4.4 Environment Policies

The choice of relative priorities will impact the behaviour of the MAS. Let a priority rule be called a *policy*. The modelling enables two policies related to the environment filters: either they are constraints, which cannot be violated, or they are "default" rules, which can be overruled. Indeed, both constraints and rules can coexist in the same MAS.

In order to allow a regulation by the environment to ensure the compliance with the interaction constraints of the MAS, the corresponding policy is that the filters put by the environment are stronger than those put by the agents. With this policy, the filters of the environment cannot be overruled by any of the filters of the agents:

Policy 1 (Environment Precedence) $\mathcal{R} : \mathcal{F} \mapsto I(IP)$

$$\mathcal{R}(f) = \left\{ \begin{array}{l} [min_e, max_e]\ if\ initiator = environment \\ [min_a, max_a]\ if\ initiator \in \mathcal{A} \end{array} \right| with \quad max_a < min_e$$

To solve the case of a conflict between positive and negative filters which have the same priority, a policy has to be chosen to order them according to their type. A restricting policy is to favour the negative filters.

In the digital city example, adding the support for private messaging means blocking the perception of these specific messages by other agents than the one explicitly addressed. A solution is to identify the messages using a boolean property "private":

$$f^{-}_{private} = \langle [id(a) = ?x], [receiver(m) \, ! = ?x] \wedge [private = true], \text{``}private'', \\ 100, environment \rangle$$

These messages will be handled for the receiver thanks to the filter f^{+}_{direct}, but no other agent is able to hear them, since the filter $f^{-}_{private}$ is treated before any agent filter.

For example, if an agent a_5 wants to overhear every private message, it will put a filter such as:

$$f^{+}_{intercept} = \langle [id(a) = \text{``}a5''], [private = true], \text{``}intercept'', 0, a_5 \rangle$$

This filter matches every message which has a $private$ property set to *true*, and identifies a_5 as the receiver. However, since $max_a < min_e$ due to the policy, $f^{-}_{private}$ is activated for all the agents except the intended receiver, which means that $f^{+}_{intercept}$ cannot be activated.

Apart from these constraints, the agents can put filters concerning the situations which are not described in the environment negative filters, and thus focus their attention toward particular interactions of interest.

The second policy enables some filters of the agents to overrule those of the environment.

Policy 2 (Facilitator Environment) $\mathcal{R} : \mathcal{F} \mapsto I(IP)$

$$\mathcal{R}(f) = \left\{ \begin{array}{l} [min_e, max_e] \; if \; initiator = environment \\ [min_a, max_a] \; if \; initiator \in \mathcal{A} \end{array} \right| with \quad max_a > max_e$$

It enables the environment to manage a standard interaction processing which can be adapted by the filters of the agents. We call this policy *facilitator environment*, because although they may be overruled, the presence of environment rules indicates the presence of a normative way to manage the messages.

In our digital city example, let IP be $IP = [-128, 127]$, a facilitator environment is defined as:

$$\mathcal{R} : \mathcal{F} \mapsto I(IP), \;\; \mathcal{R}(f) = \begin{cases} [-128, 0] & if \; initiator = environment \\ [-128, 127] & if \; initiator \in \mathcal{A} \end{cases}$$

An example of communication service is to propose group communications to contact the *organization* agents in function of their type. The condition on the message is, similarly to $f_{private}$ and f_{group}, related to a new property $receiver - type$, which is matched to the $type$ property of the agents:

$$f^{+}_{type}(a, m, C) = \langle [type(a) = ?x], [receiver - type(m) = ?x], \emptyset, \text{``}type'', 0, \\ environment \rangle$$

Thanks to this filter, all the *organization* agents receive the messages addressed to their type of organization. It is a default behaviour of the environment, that the agents may override.

While it is occupied, an *organization* agent a_x that does not want to receive these messages has to adapt its awareness

$$f^-_{unaware}(a, m, C) = \langle [id(a) = id(a_x)] \wedge [avail(a) = false], \emptyset, \emptyset, \text{``}unaware'', \\ priority, a_x \rangle$$

The agent then has two choices:

- Either it decides to give this filter a positive priority ($priority(f^-_{unaware}) > 0$), in which case it does not follow the standard environment filter f^+_{type} when the agent is not available.
- Either it decides to give this filter a negative priority ($priority(f^-_{unaware}) < 0$), in which case it follows the environment filter. The group messages are transmitted to it, but if filters of lower priority than $priority(f^-_{unaware})$ exist, these are not treated.

4.5 Individual Behaviour

Finally, we study the case of conflicts between the agents' filters. This may result from undesirable behaviour, or from poor design of the filters. If all the agents have access to the same priority intervals, an agent can block all filters of the other agents by putting a negative filter generic enough to cover every agent and every message, or inversely "flood" them with messages. In order to tackle this problem, we have introduced in Saunier et al. (2007) a particular kind of filter: the personal filters. The *personal filters* are the filters whose initiator is the only potential recipient.

Definition 5 (*Personal Filter*) f is a personal filter of a_x iff $initiator_f = a_x$ and $E(f_a) = \{a_x\}$

Personal filters trigger or block perceptions only for their initiator. We note $\mathcal{F}_{AP} \subset \mathcal{F}_A$ the set of personal filters in the environment. In this way, we distinguish filters that implicate only their owner from filters that affect – exclusively or not – other agents. Personal filters are typically used by an agent to adapt its awareness of its context in function of its current state or task.

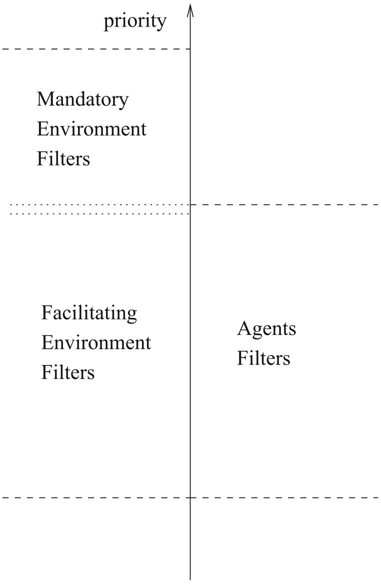
In order to counter the threat of an agent putting filters to the detriment of other agents, the personal filters have to comply with the following policy:

Policy 3 (Personal Filters Precedence) $\mathcal{R} : \mathcal{F} \mapsto I(IP)$

$$\mathcal{R}(f) = \left\{ \begin{array}{ll} [min_e, max_e] & if\ initiator = environment \\ [min_{ap}, max_{ap}] & if\ initiator \in \mathcal{A}\ and f \in \mathcal{F}_{AP} \\ [min_a, max_a] & if\ initiator \in \mathcal{A}\ and f \notin \mathcal{F}_{AP} \end{array} \right| with\ max_{ap} > max_a$$

This means that the agents personal filters overrule the filters they did not put personally in the environment, except of course for the constraints of the environment.

Fig. 4 Priority and filter types



This policy allows the agents to be less vulnerable to other agents' misbehaviours, intentional or not. When the filters are put, the environment checks that the priority given to the filter by the agent is compliant with the policies, by detecting the initiator of the filter and whether it is personal or not. Depending on the MAS designer choice the filters which are not compliant with the priority level policies are either refused or corrected.

The priority levels of the different kinds of filters, according to their initiator and nature, allow the implementation of a "natural" order of precedence (Fig. 4): all the agents must comply with the constraints of the environment, then under these mandatory rules, they define their own interactions and perceptions, and finally they may perceive solicitations of others. We emphasized in the introduction that the awareness is the result of external stimuli, which are not chosen, and of decisions of perceptions. In our model, the stimuli the agents cannot choose are the perceptions caused by the filters of the environment, while agents can decide to focus their attention by putting filters for themselves.

In the digital city example, the filters related to private messaging and direct interaction are imperative filters, in order to ensure that addressed messages are transmitted and private messages are not overheard. However, both group communication filters are facilitating filters whose effect may be altered by the agents.

5 Integrating Awareness into Communication Protocols

We have said in Sect. 2 that overhearing and multi-party communications enable to design opportunist behaviour in agents (Kaminka et al. 2002; Legras and Tessier 2004; Platon et al. 2005). If an agent has access to surrounding information, it can use this information to enhance the functioning of the system and reach its goals. However, previous works did not propose a priori design tools to take into account such hybrid interactions, with the exception of Platon et al. (2004) which proposes a visual expression of overhearing through design patterns.

5.1 A Contextual Interaction Connector

We have chosen to formalize multi-party communications in the Agent UML (Odell et al. 2000) framework, which is the result of a standardization effort of multi-agent modelling. More precisely, the multi-party communications impact the protocol specification, and therefore the interaction diagrams. The choice of a new interaction support implies that current AUML connectors are not sufficient to model its possibilities, since they are based on point-to-point communications chosen by the sender, and do not cover the cases where the initiative of the reception comes from the environment or from the receiver itself.

The proposed AUML connector is shown in Fig. 5. Direct (addressed) messages are drawn in the same way as classical communications through a straight arrow [line (1)], for *n* receivers, while indirect (overheard) messages are drawn with a perpendicular polyline [line (2)] to represent *m Listener* agents, which receive the message through another set of transmission constraints. The filter name, or set of constraints, can be added on the arrow. The original AUML role management (possibility to change the role dynamically, multiple roles…) are conserved, e.g. an agent can be a Listener and then a participant.

With this new connector, it is possible to revisit existing protocols by introducing agents using the multi-party communication properties explicitly.

5.2 Protocol Composition

In the illustrative example shown in Fig. 6, we compare two request protocols, the second being derived from the first thanks to the addition of listening capabilities.

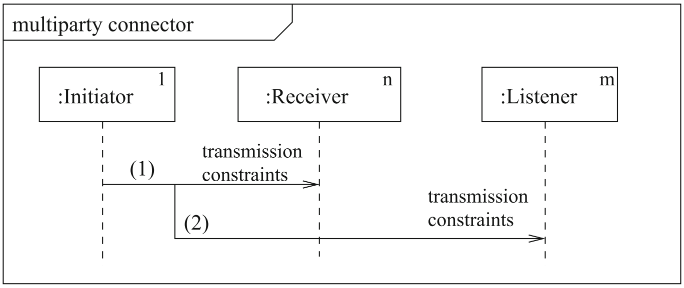


Fig. 5 AUML multi-party connector

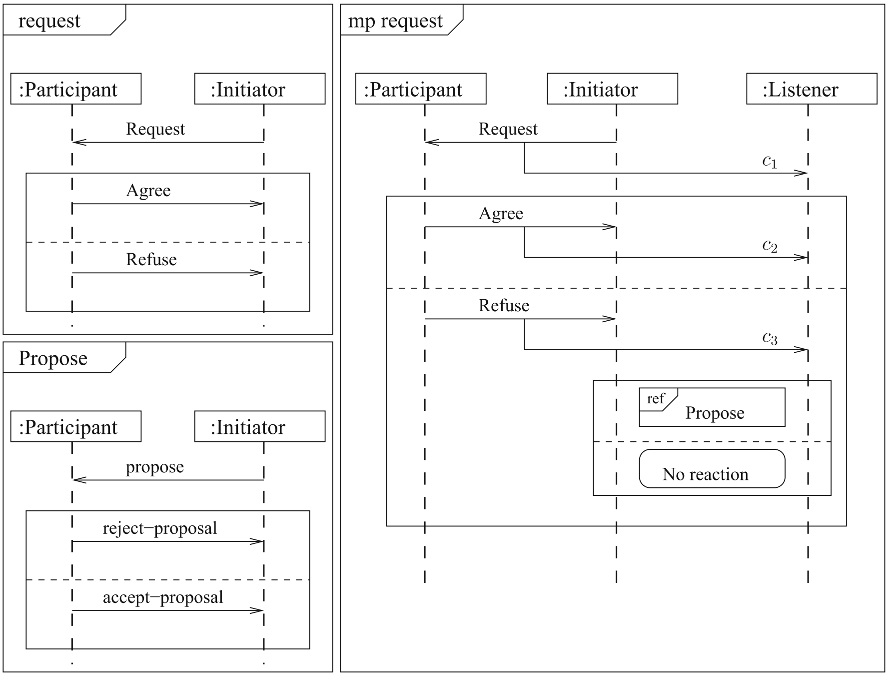


Fig. 6 Multi-party connector: simplified **request** protocol (*top left*). FIPA **propose** protocol (*bottom left*). Extended **mp request** protocol (*right*)

To improve the readability, we simplified the **request** by withdrawing the error and time-out cases. The protocol consists in the sending of a request, for example to perform a task, by the initiator agent, and in the reply *agree* or *refuse* of the participant.

The **mp request** (mp stands for multi-party) is an extension of the request protocol. It adds the possibility that the messages from the original protocol are overheard by one or more *listener* agent(s), thanks to the c_1 (for the *request* message), c_2 (for the *Agree* messages) and c_3 (for the *Refuse* messages) transmission constraints.

In the case where the participant refuses to execute the request-situation detected when the reply message is overheard-, the listener agents have the opportunity to initiate a FIPA[3] propose protocol (also shown in Fig. 6) with the initiator of the request protocol.

In the classical protocol, when the protocol fails (because of a *refuse* or *failure*), the initiator has to find by itself other agents to contact and start new protocols with these agents, each of these protocols having a cost in terms of bandwidth and computation. With the *mp request*, in the case where a listener agent can satisfy the request, it can spontaneously start a *propose* protocol to satisfy the failed request of the initiator, thus saving as many request protocols as would have been needed to find a satisfactory agent.

---

[3] The Foundation for Intelligent Physical Agents (FIPA) is the main standards organization for agents and multi-agent systems: http://www.fipa.org.

It is an *opportunistic* behaviour, since the agent bases its action on a message it was not meant by the sender to receive.

This example of extension of an existing protocol also shows how new behaviours enabled by agents aware of the surrounding communications can fit within existing multiagent systems. In this way, *opportunistic agents* using these new protocols can co-exist with traditional agents.

6 Application

In this section we present the use of our awareness model in the transportation domain. Our proposition, called SATIR (Automatic System for Network Incident Processing) has been developed for the Brussels STIB bus network operator. SATIR is a Transportation Regulation Support System that reports the network activity in real-time and thus assists the bus network regulators (Balbo and Pinson 2010). The objective is to combine the functionalities of an existing information system with the functionalities of a decision support system in order to propose a generic model of traffic regulation support system.

In urban transportation control domain, human regulators are located in a control center. They have to manage the transportation network under normal operating conditions (monitoring buses location) and also under disturbed conditions (monitoring disturbances - bus delays, bus advances -, choosing and planning actions to be taken to solve the problem).

In the majority of networks, the vehicles are located through sensors which provide real-time information. This information represents a huge amount of data (for example data arrives every 40 s in the STIB network). Furthermore it may be incomplete (a sensor may break down) or uncertain (the quality of the data may be poor). This data is collected through the Automatic Vehicle Monitoring system (AVM). The AVM system compares the actual positions of the vehicles (captured by the sensors) with their theoretical locations given by pre-registered timetables in order to detect disturbances represented by alarms on the control screen.

In SATIR, the agents use mutual awareness to tackle two issues. Dugdale shows in (Dugdale et al. 2000) that the efficiency of a regulation center is directly impacted by the ability of the agents to be aware of the interactions of the other agents, according to multiple criteria. The first issue is therefore to reproduce the same dynamics between the agents that propose solutions to a disturbance as between humans in a control center. The objective is to enable the receivers to participate actively in the interaction without overloading the agents by broadcasts. The second issue is to take into account the quality of the information for the timetable management. This second issue is presented in the following.

In our generic urban network model, we propose two categories of agents:

– STOP agents that represent the theoretical structure of the network (organized in lines and routes). They encompass the knowledge of the graph makers: passenger flows and traffic problems used to make up the theoretical timetables. A STOP

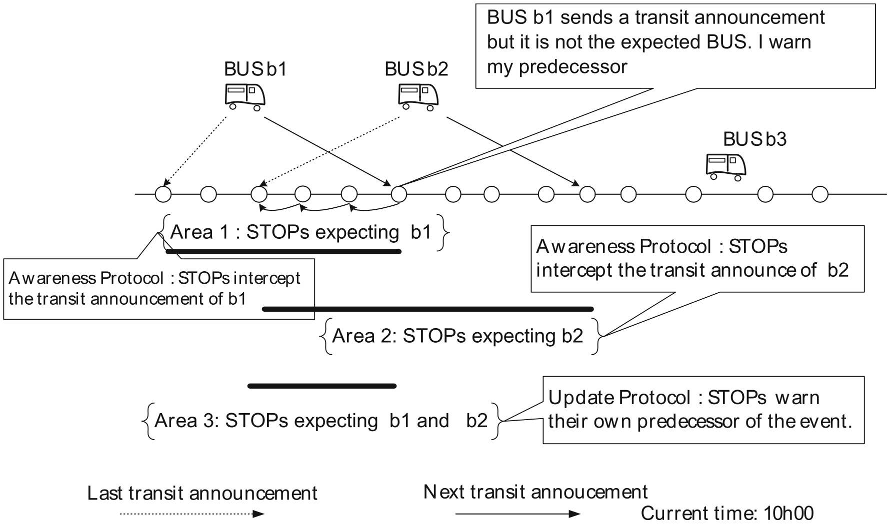


Fig. 7 Inconsistencies management by a mutual awareness approach

knows the identifiers of the STOP agents situated just before and after it on the same line(s) and route(s).

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

When a vehicle passes a stop on the real network, a transit announcement message is sent from the BUS agent to the STOP agent concerned. This message is addressed in function of the location of the stop where the bus agent is. For example, in Fig. 7 the bus agent b_1 addresses its transit announcement message to the STOP agent that is at the location 6, and the BUS agent b_2 sends its message to the stop agent situated at the location 10. The STOP agent updates its timetable by removing this vehicle from the list of vehicles due. A STOP agent which does not receive any message detects an anomaly and triggers the disturbance processing. To do so, we added in our awareness model a filter to compare the location of the STOP with the required location. This location is stored in the message by the BUS agent. The use of the location of the agents to address the communication enables each agent to define which messages it is waiting for.

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

- Awareness Protocol: when a vehicle is no longer located, the STOP agents on the bus route have not been warned about the passage of the bus, but they will be aware of all new transit announcement sent by vehicles not running to timetable. A filter is triggered when a transit announcement is sent by the BUS agents with a required location superior to the position of the STOP agent waiting for it. In Fig. 7, this filter is triggered for the STOP agents in the area 1 and 2 when they respectively receive the transit announcements sent by the BUS agents b_1 and b_2. The interceptor agents receive the message and update their timetable.
- Update Protocol: in the case of "virtual overtakings" a STOP agent receives a transit announcement of a vehicle which is not the bus it is expecting. The receiver of the transit announcement detects an anomaly and sends a message to its STOP predecessor on the route in order to announce this event. In Fig. 7, the STOP agents 6 in the area 3 is waiting for the BUS agent b_2 and it receives a transit announcement from the BUS agent b_1. Therefore it forwards this message to its predecessors that are waiting for b_1 and are not aware of this event. If the STOP agent is expecting this BUS, it updates its timetable and forwards the message to its own predecessor. If the receiver is not expecting the BUS, it does not forward the message.

7 Discussion and Conclusions

In this article, we have proposed a formal model of context awareness for communications in multiagent systems. Emerging as a first-order abstraction, the environment, in the sense of a common medium for the agents, is a suitable paradigm to support the agents' awareness. We have presented an operational model called Environment as Active Support of Interaction (EASI) that enables each agent to actively modify the environment according to its interaction needs. This model provides a suitable framework for the regulation of MAS interactions, and we have given an example of digital city application. Then priority policies were added to manage the rules governing its (un-)awareness and we have integrated awareness into communication protocols. The last section has presented an application of the awareness model to the transportation domain.

We have introduced EASI and sketched its main features, such as flexible management and regulation of the MAS interactions. In this proposal, an agent states its foci declaratively and the connection is established by the environment. The use of symbolic objects enables to take into account, in a standardized way, the needs of the agents, the context of the interaction and the rules of the MAS. Due to these features, EASI allows to enhance the management of the agent awareness in the context of multi-party communications. The extension Environment as Active Regulator of Interaction (EARI) enables different types of perception rules in a unified manner: imperative rules from the environment, which represent the signals that cannot be ignored by the agents (or reversely cannot be perceived); norms from the environment, which represent standard ways to handle the message reception; foci from the agents, which can overrule the norms from the environment and initiatives from the other agents, depending on the need of their initiator.

The EASI/EARI models only tackle the first issue in implementing awareness, i.e. the definition of the agent perception of its environment. The agent perception is an aggregate of environment rules, agent capabilities and foci, which is in fact an embodied agent approach. Once the perception is defined, the agent still has to understand its percepts, and reason on it. An alternative is for the MAS designer to set up pre-defined scenarios using this flexible perception, as shown in the protocol and bus examples.

In the following, we compare our model with the approaches found in the literature. A well-known solution to the connection problem for open systems, that matches the needs of the agents to the available services, is that of middle-agents (Sycara and Wong 2000). This solution relies on a specialized agent that is responsible for the connection. If this approach is suitable for a huge open network such as Internet, it presents several drawbacks: (1) it limits the criteria that link the connection to capabilities and preferences, (2) middle-agents act either as screens between the interlocutors (broker agents), or as advanced directories (matchmakers). Hence, dynamic awareness of the activity of the multi-agent system is not ensured. By contrast, on one hand, our EASI model extends the transmission criteria, which can be freely defined by the agents, and on the other hand allows opportunistic and dynamic behaviour.

Recently, the environment has been promoted as a common shared medium in which the agents evolve. A thorough state-of-the-art can be found in Weyns et al. (2007) and Weyns et al. (2004). Regarding the agent interaction, the basic concerns of the proposed models in the literature are (i) how an agent can evolve in systems where the configuration might change dynamically, and how this can be done without changing the agents' configuration; (ii) how an agent can evolve in a highly dynamic system (high number of exchanged messages); (iii) how two agents can influence each other without direct communication. In these works, direct and indirect interaction are generally modelled separately, for example in the reference architecture model proposed in Weyns et al. (2004). These limitations are tackled by our EASI model that standardizes the interactions through an unified environment enabling the model to support multi-party communication.

Several other frameworks for active perception are proposed in the literature. Among the most well-known are Weyns et al. (2004), Platon et al. (2007) model, Behavioural Implicit Communication (Tummolini et al. 2004), MIC* (Gouaïch et al. 2005) and the tuplespaces models (Julien and Roman 2006; Omicini and Zambonelli 1999). The active perception model (Weyns et al. 2004) is developed from the viewpoint of the agents, and not of the environment. In this work, the filtering is realized by the agents. The advantage is an increased autonomy. However, delegating this task to the environment as it is the case in EASI decreases the cost of context management and percepts processing for the agents. Platon et al. (2007) have introduced the concept of over-sensing: agents have soft-bodies that have public states, which are mediated (both for visibility and accessibility) by the environment. The information on modifications to the public states is spread throughout the environment. Nevertheless, the model does not explicitly take message management into account, and even if the agents are observable, the model does not address the questions of the regulation and of the use of the environment by the agents. Behavioural Implicit Communication (BIC) Tummolini et al. (2004), within the framework of cooperative systems for task management, is the set of interactions that can be observed indirectly, i.e. information conveyed by the actions or communications of the other agents. The properties that are required to fulfil BICs, like the observability of the actions and their results, are environment services. The EASI model is a way to instantiate an environment that support BICs for the communication part: the formulas describing the MAS are either filters or information on filters. Although it has not yet been proposed, the EASI model could be improved to support actions or objects observability by adding new primitives to the one managing messages ($receive$). However, we do not believe that the environment should have access to the motivational state of the agents to automatically generate observation rules. Instead, we have chosen to let the agents declare these rules to the environment, separating clearly the responsibilities between the agents and the environment. MIC* (Gouaïch et al. 2005) is a formal deployment environment independent of its implementation, which represents interactions as Interaction Objects (IOs). These IOs, once produced, are separated from the agents and managed by the environment. The tuplespaces models (Julien and Roman 2006; Omicini and Zambonelli 1999), which were initiated by Linda (Carriero et al. 1986), fall within the environment category even though it does not specifically address multi-agent systems. The difference with our work is that in Linda-like models agents (processes) are not necessarily represented by data inside the environment, while they are in EASI. In addition, our use of symbolic objects allows a richer expressiveness than the templates used in these models. A template tests whether the given data matches the template, whereas with a filter, the matching is conditioned by a context, i.e. environmental criteria. Standard tuple-spaces are passive, i.e. do not allow to actively dispatch messages; however programmable tuple-spaces such as TuCSoN (Omicini and Zambonelli 1999), or artifacts (Omicini et al. 2008) could be a way to implement the EASI model, by programming the filters as reactions.

Concerning the control and norms, the electronic institutions (Esteva et al. 2004) propose run-time verification of the interaction protocols. This ensures a strict compliance with the specifications of the MAS. Although the "control" part is taken into account to mediate the interaction, the infrastructure is not intended to also facilitate the interaction. Zheng et al. (2007) have proposed to enforce the control in a decentralized way, while proving global properties. This raises an important issue about the locality of the laws, the tradeoff being that taking into account the context implies requesting the information each time this one is needed. A solution, inspired from several tuple-spaces infrastructures, could be a binary range of the objects, local and global.

Several other open issues remain. Until now, we have chosen a full availability of the information to the environment, in order to ensure this support of multi-party communications, context awareness and regulation. The counterpart is a centralization of the information in the environment. The agents have the modification rights on the objects and filters they put. Allowing several levels of rights could add some flexibility to the environment.

We stated earlier that the awareness is the ability to perceive and understand the ongoing activity of others. Our work addresses the first part of this ability, but we have mentioned that we assume that some meta-knowledge about the representation of the context is shared among agents. This enables the model to be flexible in allowing any property type to be used in the filters, but implies the definition a priori of a property scheme. How to publish and reason about the rules existing in the environment is still an open question.

References

Balbo, F., & Pinson, S. (2010). Using intelligent agents for transportation regulation support system design. *Transportation Research Part C: Emerging Technologies*, *18*(1), 140–156.

Benini, M., Cindio, F. D., & Sonnante, L. (2005). Virtuose, a virtual community open source engine for integrating civic networks and digital cities. In *Digital Cities III*, Lecture Notes in Computer Science (Vol. 3081, pp. 217–232). Springer, Berlin

Bock, H., & Diday, E. (2000). Analysis of symbolic data, exploratory methods for extracting statistical information from complex data. In *Studies in classification, data analysis, and knowledge organisation* (Vol. 15). Berlin: Springer .

Branigan, H. (2006). Perspectives on multi-party dialogue. *Research on Language & Computation*, *4*(2–3), 153–177.

Busetta, P., Donà, A., & Nori, M. (2002). Channeled multicast for group communications. In: *AAMAS '02: proceedings of the first international joint conference on autonomous agents and multi-agent systems* (pp. 1280–1287). New York: ACM Press.

Camarinha-Matos, L. M., & Afsarmanesh, H. (2002). Design of a virtual community infrastructure for elderly care. In: *PRO-VE '02: Proceedings of the IFIP TC5/WG5.5 third working conference on infrastructures for virtual enterprises*, The Netherlands: Kluwer, B.V. Deventer.

Carriero, N., Gelernter, D., & Leichter, J. (1986). Distributed data structures in linda. In: *POPL'86: Proceedings of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages* (pp. 236–242). ACM Press.

Davis, R., & Smith, R. G. (1988). Negotiation as a metaphor for distributed problem solving. *Distributed Artificial Intelligence*, *20*(1), 333–356.

Diday, E. (1991). From data analysis to uncertainty knowledge analysis. In *Symbolic and quantitative approaches to uncertainty* (pp. 153–160).

Drury, J., Scholtz, J., & Yanco, H. (2003). Awareness in human-robot interactions. In *IEEE international conference on systems, man and cybernetics* 2003. (Vol. 1, pp. 912–918).

Dugdale, J., Pavard, J., & Soubie, B. (2000). A pragmatic development of a computer simulation of an emergency call center. In *Designing cooperative systems: the use of theories and models* (pp. 241–256). Amsterdam: IOS Press.

Endsley, M. (1988). Design and evaluation for situation awareness enhancement. In *Human factors society, annual meeting, 32nd* (pp. 97–101). CA: Anaheim.

Esteva, M., Rodriguez-Aguilar, J., Rosell, B., & Arcos, J. (2004). Ameli: An agent-based middleware for electronic institutions. In: *Proceedings of the 3rd international joint conference on autonomous agents and multi-agent systems (AAMAS 2004)* (pp. 236–243). New York: ACM Press.

Gouaïch, A., Michel, F., & Guiraud, Y. (2005). Mic*: A deployment environment for autonomous agents. In: Weyns, D., Parunak, H. V. D., & Michel, F. (eds.) *Proceedings of environment for multi-agent systems, workshop held at the third AAMAS conference* (Vol. 3374, pp. 109–126) Springer, LNAI.

Gutnik, G., & Kaminka, G. A. (2006). Representing conversations for scalable overhearing. *Journal of Artificial Intelligence Research*, *25*, 349–387.

Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S., & Yoshida, S. (1999). Socialware: Multiagent systems for supporting network communities. *Communications of the ACM*, *42*(3), 55. doi:10.1145/295685.295701.

Heath, C., Svensson, M. S., Hindmarsh, J., Luff, P., & vom Lehn, D., (2002). Configuring awareness. *Computer Supported Cooperative Work*, *11*(3), 317–347. doi:10.1023/A:1021247413718.

Julien, C., & Roman, G. C. (2006). Egospaces: Facilitating rapid development of context-aware mobile applications. *IEEE Transactions on Software Engineering*, *32*(5), 281–298. doi:10.1109/TSE.2006.47.

Kaminka, G., Pynadath, C., & Tambe, M. (2002). Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, *17*, 83–135.

Kumar, S., Huber, M. J., McGee, D., Cohen, P. R., & Levesque, H. J. (2000). Semantics of agent communication languages for group interaction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (pp. 42–47). AAAI Press/The MIT Press.

Legras, F., & Tessier, C. (2004). Lotto: Group formation by overhearing in large teams. In *Advances in Agent Communication, LNAI* (Vol. 2922, pp. 254–270) Berlin: Springer.

Markopoulos, P., De Ruyter, B., & MacKay, W. (2009). *Awareness systems: Advances in theory, methodology and design*. Berlin: Springer.

Odell, J., Parunak, H. V. D., & Bauer, B. (2000). Extending uml for agents. In *Proceedings of the agent-oriented information systems workshop at the 17th conference on artificial intelligence*.

Omicini, A., & Zambonelli, F. (1999). Tuple centres for the coordination of internet agents. In: *SAC'99: Proceedings of the 1999 ACM symposium on Applied computing* (pp. 183–190). New York: ACM Press. doi:10.1145/298151.298231.

Omicini, A., Ricci, A., & Viroli, M. (2008). Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *17*(3), 432–456.

Platon, E., Sabouret, N., & Honiden, S. (2004). T-compound: An agent-specific design pattern and its environment. In *Proceeding of the third international workshop on Agent Oriented Methodologies at OOPSLA* (pp 63–74).

Platon, E., Sabouret, N., & Honiden, S. (2005). Overhearing and direct interactions: Point of view of an active environment. *Proceedings of environment for multi-agent systems, workshop held at the fourth AAMAS conference, Springer Verlag, LNAI* (vol. 3830, pp. 121–138).

Platon, E., Sabouret, N., & Honiden, S. (2007). Tag interactions in multiagent systems: Environment support. In Weyns, D., Parunak, H. V. D., Michel, F. (eds.), *Proceedings of Environment for Multi-Agent Systems, Workshop held at the fifth AAMAS conference, LNAI* (Vol. 4389, pp. 106–123) Berlin: Springer.

Ricci, A., Omicini, A., Viroli, M., Gardelli, L., & Oliva, E. (2007). Cognitive stigmergy: Towards a framework based on agents and artifacts. In: Weyns, D., Parunak, H. V. D., Michel, F. (eds.). *Proceedings of Environment for multi-agent systems, workshop held at the fifth AAMAS conference, LNAI*, (Vol. 4389, pp. 124–140). Berlin: Springer.

Saunier, J., & Balbo, F. (2007). An environment to support multi-party communications in multi-agent systems. *Multi-agent systems and applications V, 5th international central and eastern European conference on multi-agent systems, CEEMAS 2007* (pp. 52–61). Leipzig, Germany: Springer.

Saunier, J., & Balbo, F. (2009). Regulated multi-party communications and context awareness through the environment. *International Journal on Multi-agent and Grid Systems*, *5*(1), 75–91.

Saunier, J., Balbo, F., & Badeig, F. (2007). Environment as active support of interaction. In: Weyns, D., Parunak, H. V. D., Michel, F. (eds.) *Proceedings of environment for multi-agent systems, workshop held at the fifth AAMAS conference, LNAI* (vol. 4389, pp. 87–105). Berlin: Springer.

Sycara, K., & Wong, H. (2000). A taxonomy of middle-agents for the internet. In *ICMAS '00: Proceedings of the fourth international conference on multiagent systems (ICMAS-2000), IEEE computer society* pp. 465–466.

Traum, D., & Rickel, J. (2002). Embodied agents for multi-party dialogue in immersive virtual worlds. In *AAMAS'02: Proceedings of the first international joint conference on autonomous agents and multiagent systems* (pp. 766–773) New York: ACM Press.

Tummolini, L., Castelfranchi, C., Ricci, A., Viroli, M., & Omicini, A. (2004). "Exhibitionists" and "voyeurs" do it better: A shared environment approach for flexible coordination with tacit messages. In: Weyns, D., Parunak, H. V. D., Michel, F. (eds.), *Proceedings of environment for multi-agent systems, workshop held at the third AAMAS conference, LNAI*, (Vol. 3374, pp. 215–231) Berlin: Springer.

van den Besselaar, P., & Beckers, D. (2005). The life and death of the great amsterdam digital city. In *Digital Cities III*, Lecture Notes in Computer Science (Vol. 3081, pp. 66–96) Berlin: Springer.

Vertegaal, R., Velichkovsky, B., & Van Der Veer, G. (1997). Catching the eye: Management of joint attention in cooperative work. *SIGCHI Bulletin*, *29*, 87–92.

Warren, R. (1999). *Auditory perception. A new analysis and synthesis*. Cambridge: Cambridge University Press.

Weyns, D., Parunak, H. V. D., Michel, F., Holvoet, T., & Ferber, J. (2004). Environments for multiagent systems, state-of-the-art and research challenges. In: Weyns, D., Parunak, H. V. D., Michel, F. (eds.), *Proceedings of Environment for Multi-Agent Systems, Workshop held at the third AAMAS conference, LNCS* (Vol. 3374, pp. 2–52).

Weyns, D., Steegmans, E., & Holvoet, T. (2004). Towards active perception in situated multi-agent systems. *Journal on Applied Artificial Intelligence*, *18*(9–10), 867–883.

Weyns, D., Omicini, A., & Odell, J. (2007). Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *14*(1), 5–30. Special issue on Environments for Multi-agent Systems.

Zheng, W., Serban, C., & Minsky, N. (2007). Establishing global properties of multi-agent systems via local laws. In: Weyns, D., Parunak, H. V. D., Michel, F. (eds.), *Proceedings of Environment for multi-agent systems, workshop held at the fifth AAMAS conference, LNAI* (Vol. 4389, pp. 170–183). Berlin: Springer.