EVALUATING THE COST OF SUPPORTING INTERACTION AND SIMULATION THROUGH THE ENVIRONMENT

Flavien Balbo, Fabien Badeig
Université Paris-Dauphine, LAMSADE, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France
Université Paris Est, GRETIA, INRETS, 2 Rue de la Butte Verte, 93160 Noisy le Grand, France

Julien Saunier
Université Paris Est, LEPSiS, UMR INRETS LCPC, 58 Bld Lefèbvre, 75015 Paris, France

Keywords: Environment, Evaluation, Communication, Activation.

Abstract: The environment has emerged as a powerful first-order abstraction in Multi-Agent Systems (MAS), as well as a critical building block. One benefit is to reduce the complexity of the agents by delegating to the environment a part of the tasks of the system. This delegating process provides a flexible way to exchange information and to coordinate the agents thanks to the environment. The counterpart is a centralization of a part of the MAS processes inside the environment.
In this paper, we present the modeling of an environment for multi-agent communication and simulation. Our proposition enables the addition of advanced features to the MAS like multi-party communications (communication) and contextual activation (simulation). We evaluate the cost of this environment process and compare it to the execution of the same tasks in the agents for communication and simulation.

1 INTRODUCTION

In (Weyns et al., 2007), the authors give the following definition of a multi-agent system (MAS) environment: *"The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources"*. The environment is a critical building block in the multi-agent systems that encapsulates its own responsibilities. The agents interact in their environment and the effects of their actions are observed and evaluated through this environment. Thus, the environment provides observability and accessibility services.

The relation between the environment and the agents is traditionally based on a perception - decision - action cycle (Weiss, 1999) which is repeated by the agents during their life time. The three phases of the cycle are executed in each agent. The perception being the agent ability to observe its environment, its result is the computation of what can be called *contexts*. A context is defined as a set of information that can be used to characterize a situation. This set of information includes percepts and messages that are obtained by the agent and its own state. Thanks to the context computation, the agent decides to execute the suitable action. The relation between the environment and the agents based on this perception - decision - action cycle does not exploit the potential of the environment that remains a static entity. This relation can be enhanced by delegating the perception process to the environment (Weyns et al., 2004; Badeig et al., 2007; Saunier and Balbo, 2009; Kesaniemi et al., 2009). The objective is to support advanced features that are difficult to obtain without the environment.

We have proposed an environment modelling and the adequate platform that are based on the delegation of the perception process to the environment. Our proposition supports two advanced features: 1) Multi-Party Communications (MPC) (Branigan, 2006) takes into account dyadic interaction (one to one), group interaction (one to many) and overhearing (many to one/many) within the same interaction process; 2) Contextual activation (Laberge, 1975) applied to the simulation process consists in activating the agents directly according to their context.

Balbo F., Badeig F. and Saunier J..
EVALUATING THE COST OF SUPPORTING INTERACTION AND SIMULATION THROUGH THE ENVIRONMENT.
DOI: 10.5220/0003182001260135
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 126-135
ISBN: 978-989-8425-40-9

In our proposition, despite the delegation process to the environment, the agents keep their autonomy because the decision process and the action process remain in the agents. The perception phase of the cycle is modified to take into account the relation with the environment. That means that the environment computes the contexts for the agents, and that the agents can modify this computation process of the environment to suit their needs: the agents decides which contexts are important for them and act by modifying in consequence the computation process that is executed by the environment.

In our proposition, in the MPC case, the environment mediates the communication: the context is the information about the agent receiver, the message and any other environment information which is relevant to this communication. if the context is validated, the environment addresses the messages to the related agents. In the simulation case, the environment is the scheduler that manages the agent activation: the context is the subset of information accessible to the agents. The environment activates an agent in a specific context and according to this context, the agent performs the suitable action.

To adapt the perception process to each agent, the environment needs 'tools' to compute their contexts. In our proposition, these tools are *filters* that reify the relation between an agent (described in the environment) and its contexts. A filter contains the constraints on the context related to an action of an agent. This action is the reception of a message for communication filters and a specific agent action to perform after validation for activation filters. The agents modify their relation to the environment thanks to the addition and removal of their filters in the environment. This process is dynamic and the environment activates only its current filters. These choices (addition/removal) belong to the decision process that is managed by each agent.

The externalization of the computation of the perception process improves qualitatively the design of a MAS. The dynamic filter management by the agents gives more flexibility to the MAS design. More details on qualitative improvement are given in (Saunier and Balbo, 2009) for communication and in (Badeig et al., 2007) for simulation. The counterpart is a centralization of a part of the MAS processes inside the environment. In this paper, we evaluate the cost of this centralization for computing agent contexts, in comparison to the execution of equivalent solutions for communication and simulation.

Section 2 motivates the delegating process of tasks to the environment for communication and simulation and introduces an illustrative example with different examples issued of a crisis management application. Section 3 presents the environment model. Section 4 provides a theoretical evaluation. In section 5, we provide an empirical evaluation, and we conclude in section 6.

2 MOTIVATIONS

2.1 Communication

Recent research on multi-party communications (Branigan, 2006; Saunier and Balbo, 2009) shows how multi-agent communications can take advantage of the complexity of the human communication process. The main issue in supporting MPC is to take into account dyadic interaction (one to one), group interaction (one to many) and overhearing (many to one/many) within the same interaction process. The sender does not know all the agents that might be interested in its messages. For example, an agent can listen to messages without the agreement/knowledge of the sender through overhearing. For a recipient, the usefulness of a message may depend on the context of the sender, the context of the message, and the context of the recipient itself.

These challenges are related to the way the recipients are chosen. MPC requires knowledge of the needs of both the sender and the recipients. From the sender viewpoint (direct interaction), it is a connection problem: which agents are related to my message? The problem is to map the senders needs (information, capabilities, resources, ...) to the address of related agents. From the recipient viewpoint (indirect interaction), it is a data extraction problem: which messages are related to me? The problem is to map the recipients needs to the content of the messages. For each message, these problems have to be simultaneously solved by the communication infrastructure. The environment is able to solve these problems by mediating the communication in order to find all the receivers of each message. Nevertheless, in the cognitive agents community, few works 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 (*e.g.* (Picco and Buschini, 2002)) 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).

If the computation of the context does not take into account ambient conditions, it can classically be done inside the agents. This solution implies that the agents receive all messages and filter them. An evaluation of the environment support consists in the cost comparison of the context filtering either in each agent after the reception or in the environment during the transmission process.

2.2 Agent-based Simulation

In a simulation, the scheduling policy defines the activation order of the agents. Once activated, the agents behave according to their context. How the agents are activated and what information is available to compute the agent context depend on the agent-based simulation (ABS) framework.

In most ABS frameworks, a scheduler activates the agents one after the other and each activated agent computes its context to choose one action to execute at each simulation step. The classical ABS frameworks are designed to support this activation process. For example, in the platforms CORMAS (Bousquet et al., 1998) and MASON (Luke et al., 2005), the scheduler activates a standard method for each agent. This method is specialized by the designer to adapt the agent behavior. In the Logo-based multi-agent platforms such as *TurtleKit* (Gutknecht and Ferber, 2001) or the STARLOGO system[1], an agent has an automaton that determines the next action to perform.

The choice of these platforms to be agent-oriented, with a light environment support, implies that the computation of the context is repetitive because it is computed in each agent at each time cycle during the simulation execution. This computation is done in each agent even if the agent context does not change between two time cycles and/or if several agents share the same context during a time cycle. Here, the evaluation consists in the cost comparison of the context filtering either in each agent after the activation or in the environment during the scheduling process.

Example. In this paper, we consider a crisis management application where several emergency services must be coordinated in order to reduce the crisis effects. A crisis situation is a dynamic phenomenon defined by the initial situation, which depends on place and time, and by the impact on population and infrastructure. We focus our example on a specific point which is the victim evacuation. This task consists in coordinating two agents playing the role *medical porter*. The goal of the medical porters is to shift a victim to an emergency vehicle. This action requires one medical porter with the skill *medical monitoring* and one medical porter with the skill *victim handling*. The first skill allows the medical porter to monitor the victim and to inform the hospital of the evolution of the victim health. The second skill is necessary to handle correctly the victim.

Each simulation component (agents, messages and objects) is situated on a grid, *i.e.* a two-dimensional space. In this example, we consider that the victims belong to the set of the objects because they are not autonomous. The agents (medical porter) act in this environment and cooperate in order to evacuate victims. A medical porter can move *randomly* or towards *a given direction*, and it possesses only one skill. It can either *monitor* or *handle* a victim, but the two skills are required to evacuate the victim. The agents can communicate in order to find a partner with the complementary skill. Each agent has a field of perception that limits its perception of the environment. A medical porter is able to perform one of the following actions in a time cycle of simulation: 1) the action *move randomly*, 2) the action *move towards a location*, 3) the action *wait*, and 4) the action *evacuate a victim*.

We use this application along the paper to illustrate our study of the cost of the environment supporting interaction and simulation.

3 ENVIRONMENT MODELING

To support multi-party communications and contextual activation, we propose to use the environment as a privileged intermediary to control interaction (EASI, Environment as Active Support for Interaction) and activation (EASS, Environment as Active Support for Simulation). The EASS model (Badeig et al., 2007) embeds the EASI model (Saunier and Balbo, 2009). The environment manages meta-informations on the MAS (agents, messages, context) and uses them to compute the agent context(s). In this section, we give the background elements to understand the assessment of the cost of the environment. More details about the models can be found in [?][3].

The environment model EASI is thus defined by $\langle \Omega, \mathcal{D}, P, \mathcal{F} \rangle$ with:

- $\Omega = \{\mathcal{A} \cup \mathcal{MSG} \cup O\} = \{\omega_1, ..., \omega_m\}$ the set of entities (with $\mathcal{A}$ the set of agents, $\mathcal{MSG}$ the set of messages and O the set of objects, *i.e.* all entities that are not agents or messages),
- $\mathcal{D} = \{d_1, ..., d_m\}$ the set of domain descriptions of the properties,

[1] http://education.mit.edu/starlogo/

- $P = \{p_1, ..., p_n\}$ the set of properties,
- $\mathcal{F} = \{f_1, ..., f_k\}$ the set of filters.

Entity: An *entity* $\omega_i \in \Omega$ is defined by $\langle e_r, e_d \rangle$ where e_r is a reference to an element of the MAS and e_d is the description of that element. An element of the MAS can be an agent, a message or an object. A reference is its physical address on the platform or other objects (such as URL, mailbox, *etc.*). The description e_d is a set of couples $\langle p_i, v_j \rangle$ where $p_i \in P$ and v_j is the value of the property for the entity. Any agent of the MAS has its own processing and knowledge settings. It is connected to the environment thanks to its description that the environment stores and updates. This description e_d is used for the routing of the informations to the agent with the reference e_r (EASI) or its activation (EASS). The agents can modify their description dynamically.

Property: A property $p_i \in P : \Omega \rightarrow d_j \cup \{unknown, null\}$ is a function whose description domain $d_j \in \mathcal{D}$ can be quantitative, qualitative or a finite set of data. The *unknown* value is used when the value of the property cannot be set, and *null* is used to state that the property is undefined in the given description. In order to simplify the notation, only the value of the description domain d_j is given to specify a property.

Filter: A filter identifies the entities according to their description (e_d) and realizes the interaction between the concrete objects (e_r). A filter $f_j \in \mathcal{F}$ is a tuple $f_j = \langle f_a, [f_m], [f_C], n_f \rangle$ with n_f the filter name. The assertion $f_a : \mathcal{A} \rightarrow \{true, false\}$ identifies the related agents, the assertion $f_m : \mathcal{MSG} \rightarrow \{true, false\}$ identifies the related messages, and $f_C : \mathcal{P}(\Omega) \rightarrow \{true, false\}$ is an optional set of assertions identifying other entities of the context. A filter is considered valid for any tuple $\langle agent, [message], [context] \rangle$ such as $f_a(agent)[\wedge f_m(message)][\wedge f_C(context)]$ is true.

The choice of the action specializes the filter. Every agent a whose description validates f_a is activated if there exists a set of entities in the *context* such that f_C is true. After its activation, the agent validates the action related to this context and executes it (or not). It is a communication filter if the action is to add the message m that satisfies f_m in the message box of the agent a. The agents can add and remove dynamically the filters. Thus, they adapt their relation with the environment according to their needs.

In a crisis simulation, the agents have observable properties: their identifier, which is unique, their location, their availability (true/false values) which specifies if an agent rescues a victim, their skill (*position/monitoring*), their field of vision, and their internal time. The messages have the following properties: their identifier that is unique, their sender identifier, the message type ('request'/'accept'), their location which is the location of the sender when the message is put in the environment, and the skill required by the sender in the case of communication. The victims have three properties: their identifier, their location and their status (*stretcher/vehicle*). The victims to be evacuated must be diagnosed; this is expressed by the property *diagnosed* (true/false values). The following table sums up the properties of the MAS components descriptions:

property	Medical porter	Message	Victim
id	x	x	x
location	x	x	x
availability	x		
vision	x		
time	x	x	
skill	x	x	
id_{sender}		x	
type		x	
id_v		x	
status			x
diagnosed			x

In the simulation example, let f^e be a communication filter related to the following context: an agent is interested by a message if it is available (f_a), if the request message is close to the agent a and a has the requested skill (f_m) and if the victim has been diagnosed (f_c). In this example, f_c contains one entity which is a victim identified by the property id_v of the message. The request messages have information about the victim location and the skill of the medical porter.

The processing of the context is traditionally done by the agents, but in EASI and its extension EASS it is done by the environment. In this way, the computation can be done with information that is shared by the agents (the entities descriptions) and with a degree of mutualization. However, it implies the centralization of the computation and the management of the information update. Now, we propose to evaluate the cost of this centralization.

4 THEORETICAL ASSESSMENT

The theoretical assessment is the comparison between the processing by the environment of a communication filter for n_a agents and the processing by n_a agents of all the messages. In order to take into account the context dynamics, this theoretical assessment also studies the update process of the MAS. The theoretical assessment is based on two criteria. The first, noted $Cost_T$, is the number of tests per-

formed during the filtering process and the second, noted $Cost_M$, is the number of resulting messages. The objective is to identify in which cases it is better to mediate the communication through the environment than to manage it in the agents.

Following the definition given in section 3, a filter f has tests related to one agent description (f_a), one message description (f_m) and a subset of entities (f_c). To simplify the explanation, we assume there is only one entity to match f_c and this entity cannot be an agent. This last assumption implies that there is no additional cost for the update process if the message processing is executed by the agents. The following generic filter illustrates our assessment, it is shown in a classical "if" structure:

Tests	N
if $\wedge_{i=1..n} T_i^m(?m)$	(1)
and $\wedge_{i=1..n} T_i^a(?a)$	(2)
and $\wedge_{i=1..n} T_i^{a,m}(?a,?m)$	(3)
and $\wedge_{i=1..n} T_i^c(?c)$	(4)
and $\wedge_{i=1..n} T_i^{m,c}(?m,?c)$	(5)
and $\wedge_{i=1..n} T_i^{a,c}(?a,?c)$	(6)

The elements $?a, ?m$ and $?c$ are variables that respectively match an agent, a message and an entity. Let T_i^x be a test on a x entity and $T_i^{x,y}$ a test between two entities. For the filter example f^e: an agent ($?a$) is interested by a request message ($?m$) (1) if it is available (2), if this request message is close to the agent and the agent $?a$ has the skill specified in the message (3) and if the victim to evacuate ($?c$) (5) is diagnosed (4). $T_1^a(?a)$ is the result of the first test on an agent related to its availability and $\wedge_{i=1..n} T_i^a(?a)$ are all the tests on the agent receiver. $T_1^{a,m}(?a,?m)$ is the result of the first test on an agent and a message related to the requested skill and $\wedge_{i=1..n} T_i^a(?a,?m)$ are all the tests on the receiver and the message.

Evaluation of the Number of Tests. A filter is composed of two types of tests. The first type is a comparison between the value of a property and a constant: the tests (1)(2)(4). The second type of test implies a matching process: the tests (3)(5)(6). For each of these types the cost in number of tests is related to the number of entities that have to be tested and on the result of the previous tests on the same set of entities. For example, the number of agents that is tested by $T_2^a(?a)$ is related to the number of agents that have validated the test $T_1^a(?a)$. Let t_i be the percentage of entities that validates the test i.

Let $|C_i|$ be the number of tests and C_i be the number of evaluated tests of (i), the value of C_i is:

$$C_i = \begin{cases} 1 + \sum_{j=1}^{|C_i|} \prod_{k=1}^{j} t_k & \text{if } |C_i| > 1 \\ 1 & \text{if } |C_i| = 1 \\ 0 & \text{if } |C_i| = 0 \end{cases}$$

The cost of a filter is the sum of the costs of its tests (sequentially validated). The following table gives the cost of the generic filter if one message has to be processed by the environment (column E) or by the n_a agents (column A) .

N	E	A	tested entities
(1)	C_1	$n_a * C_1$	$n_m = 1$
(2)	$n_a * C_2$	$n_a * C_2$	all the agents in E and each agent individually (A)
(3)	$p_2 * n_a * C_3$	$p_2 * n_a * C_3$	p_2 = percentage of agents that have the requested state
(4)	$n_c * C_4$	$(p_2 * p_3 * n_a) * n_c * C_4$	n_c =number of entities that are tested in (4). p_3 = percentage of agents that are related to the message.
(5)	$p_4 * n_c * C_5$	$(p_2 * p_3 * n_a) * (p_4 * n_c) * C_5$	p_4 =percentage of entities related to the context (4)
(6)	$(p_2 * p_3 * n_a) * (p_5 * p_4 * n_c) * C_5$	$(p_2 * p_3 * n_a) * (p_5 * p_4 * n_c) * C_5$	p_5 =percentage of entities validating (5)

There are two differences between these two processing: 1) the factorization of the tests if the processing is executed by E; 2) the implicit elimination of the agents if the processing is executed by the agents. If the processing is executed by E the tests are factorized, for example (1) is executed only once and is repeated for n_a agents. Each test (i) that is false implies an elimination of the agent that executes the the processing. For example in (5) only the subset of agents where the test (4) is validated continue the processing.

Following the hypothesis that the filtering and the shared information are the same, the matching done by the agents is more costly than by the environment for the tests (1)(4)(5) and are the same for (2)(3)(6). We conclude that the mediation by environment is always better than the broadcast when at least one agent is interested in the message. Nevertheless the implicit agent elimination implies that if the agents do not share all information some part of the filter evaluation can be less costly for the agents than for the environment. This parameter will be empirically tested in section 5.

Moreover, the environment and the agents need the same information to process the filters and the result has to be balanced by the cost of the access to this information.

Evaluation of the Number of Messages. If the exchange of information is done via messages, there are two types of messages to take into account: 1) the messages related to the interactions in the MAS such as the request messages in our example f^e; 2) the messages related to the update process.

Firstly we evaluate the number of messages that are the result of the interaction in the MAS. Let n_m be the number of messages sent by the agents, let p_r be the average percentage of agents that have to receive the messages. If the matching process is executed by the environment then each message is sent to the environment, and the environment transmits it to the $p_r * n_a$ receivers. The total number of messages is therefore $n_m(1 + p_r * n_a)$. If the matching process is executed by the agents, then each message is sent to all the agents which locally execute the matching process. The total number of messages is therefore $n_m * n_a$. Except if p_r is close to 1 which means that the message is related to all the agents, the environment mediation is less costly. The more the selection is important, the more the mediation by the environment is beneficial.

Secondly we evaluate the number of messages that are the result of the update process of the MAS. To simplify, we consider only the update process of the descriptions of the agents and not of the context, although the principle is the same. In our example, it implies that the state of the victims is not updated (or that its cost is the same if it is computed by the environment or by the agents). When an agent is updated all its properties are updated in the environment. Remember that the filter example does not contain tests on the state of the agents to compute the context (f_c). Hence, the computation by the agents does not imply additional costs because we suppose the agents do not need to access updated information about the other agents, which is an underestimation of the cost of the "agent" solution.

When the communication is mediated by the environment, the agents update their descriptions by putting in the environment new descriptions that replace the old ones. This action is associated to the sending of a message. One agent can make between 0 and n updates during a time period. Let $freq_a$ be the average frequency of the description update during a time period. There is therefore $freq_a * n_a$ update messages if the interaction is mediated by the environment during the reference time period.

To have the total cost according to the number of messages criteria we have to sum these costs. Let $Cost_M^E = n_m(1 + p_r n_a) + freq_a n_a$ be the total environment cost and $Cost_M^A = n_m n_a$ be the total agents cost.

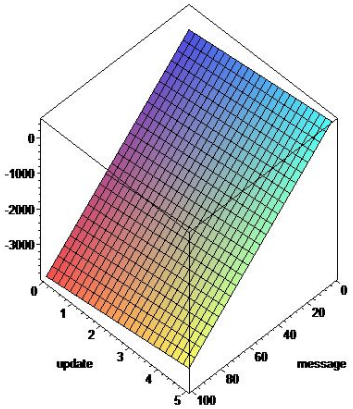

Figure 1: Comparison of the cost with $p_r = 0.6$ and $n_a = 100$.

To compare these two costs, we study the sign of the subtraction of the two costs $Cost_M = Cost_M^E - Cost_M^A$ in order to find when the environment mediation is more costly than the local computation in function of the number of messages and of the frequency of the update process. The resulting formula is $Cost_M = n_m(1 + n_a(p_r - 1)) + freq_a n_a$. Figure 1 gives the plan corresponding to this function in the three dimensional space (message, agent update, cost) if the number of agents (n_a) and the average proportion of receiver (p_r) are respectively fixed to 100 and 0.6. These assessment parameters are unfavourable to the mediation by the environment: 1) the cost of the update process is not taken into account when the context is computed by the agents; 2) the agents are interested by more than half the messages. Nevertheless we can see that there are few cases in which the use of the environment is more costly than the local matching according to the number of messages criteria.

The function $Cost_M(n_m, freq_a) = 0$ enables to find the number of messages dedicated to the interaction in the MAS (m) that have to be mediated by the environment to compensate the cost of the update process ($freq_a$). The relation between these two parameters is $n_m = -(n_a/(1 + n_a(1 - p_r))) * freq_a$, the value of n_m is therefore inversely proportional to the proportion of the agents interested in the messages. Figure 2 illustrates this proposition. The environment area represents the cases where it is better to use the environment than to mediate the messages in function of the number of messages. For instance, with $freq_a = 2$, if there is more than 5 messages related to the interaction in the MAS, then it is better to use the environment to mediate the communication even if that

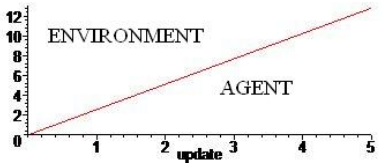

Figure 2: Comparison of the number of messages in function of the agent update frequency with $p_r = 0.6$ and $n_a = 100$.

means that there are $freq_a n_a = 200$ messages related to the update process.

This study has been done in the worst case for the environment because the agents are not taken into account to evaluate the interaction context (f_c). If it were the case, there would be no modification of the update cost of the environment. However, when the matching is executed by agents, the number of messages in the update process would be increased by $freq_a\ (n_a - 1)$ (the cost of broadcasting the description updates), thus improving the comparative environment performance.

In this section the assessment has been done with the hypothesis that the agents have the same information to evaluate a filter. In the next section, we depict how these results relate to real experimentation where this hypothesis is not valid.

5 EMPIRICAL ASSESSMENT

In section 4, we have shown that using the environment to compute the contexts is less costly than computing them locally in the agents when the agents update frequency is not too high. In other words, the entities dynamics is the parameter that determines which solution is the best for a multi-agent system. This section consists in evaluating the cost of the context computing on a real example that is a simulation of the victim evacuation in the crisis situation. The simulation run-time results aggregate the previous calculations.

A prototype of our ABS framework has been implemented as a plugin for the multi-agent platform Madkit (Gutknecht and Ferber, 2001). This plugin is composed of an environment component with an API that enables the agents to add/retract/modify their descriptions and filters. We have chosen to implement the matching process within a Rules-Based System (RBS). The instantiation of the model into a RBS is straightforward: the descriptions are the facts of the rule engine, and the filters are its rules. Rule firing is based on the efficient RETE algorithm (Forgy, 1982). It is a network-based algorithm designed to speed the matching of patterns with data. RETE uses a static discrimination network, generated by the language compiler, that represents data dependencies between rule conditions.

We compare our model which encompasses contextual activation by the environment with a model which encompasses a classical activation process to evaluate the cost of supporting the simulation process through the environment. We have tested two simulation scenarios, without and with communication, using ten filters: seven filters for contextual activation (f_1 to f_7), two filters for communication (f_{recept} and f_{accept}) and one filter for classical activation $f_{classicalactivation}$. Except for the filter $f_{classicalactivation}$, each activation filter allows to activate an agent *medical porter* in a specific context to perform an action. The filter $f_{classicalactivation}$ is used in the scenarios S_1 and S_3 in order to simulate a classical activation process. This filter activates each agent once by simulation cycle. It does not take into account the context.

The scenarios are defined below. S_1 and S_2 are scenarios without inter-agent communication, and S_3 and S_4 are scenarios with inter-agent communication:

- –S_1– $\{f_{classical\ activation}\}$ + Local Agent Context Analysis (LACA),
- –S_2– $\{f_1, f_2, f_3, f_4, f_5\}$,
- –S_3– $\{f_{classical\ activation}\}$ + LACA + Broadcast,
- –S_4– $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$ + $\{f_{recept}, f_{accept}\}$

S_1 illustrates a classical scenario with an activation phase and a local agent context analysis; S_2 is a scenario with a contextual activation inside the environment. We describe the filters of S_2. The filter f_1 allows to activate the action *move randomly* that is triggered when a medical porter has no victim next to it (context *victim seeking*). The filters f_2 and f_3 correspond to the association of the action *move towards a location* with the contexts *closest victim discovery* (f_2) and *handled victim discovery* (f_3). The context *closest victim discovery* happens when an agent perceives the closest victim to evacuate in its perception field. The context *closest victim discovery* depicts the situation where an agent *medical porter* perceives a victim that is handled by another medical porter. The filter f_4 activates the action *wait* and triggers when a medical porter is alone close to a victim to evacuate (context *victim proximity*). The filter f_5 corresponds to the association between the action *evacuate victim* and the context *rescue victim*. This context appears when two medical porters with the complementary skill are close to a victim ready to be evacuated. This context is related to the following information: the availability of the porters, the location of the porter

and the victim. The comparison between S_1 and S_2 enables to study the cost of the activation process.

S_3 and S_4 are the extensions of respectively the scenarios S_1 and S_2 with the communication filters f_{recept} and f_{accept}, and the associated activation filters f_6 and f_7. The filters f_{recept} and f_{accept} provide the support for the communication process: f_{recept} enables the reception of the request messages and f_{accept} enables the reception of their answer(s). When the agents communicate, the simulation has to manage two new agent actions that are activated by the filters f_6 and f_7. The filter f_6 is related to the action *contact*: an agent puts a request message in the environment and waits for an answer. The filter is activated when the agent is close to a victim to evacuate but does not have another agent nearby to evacuate this victim. The filter f_7 is related to the action *coordination*: an agent answers to the contact agent and moves towards the victim location. The context of this filter is the reception of a request message containing the location of the victim. The comparison between S_3 and S_4 enables to study the cost of the activation when the medical porters use communication protocol to contact another medical porter with the skill required.

Activation. We have run three series of simulations characterized by two parameters: number of agents *medical porters* and field of vision. For each series of simulations, we evaluate our model using the average run-time over 50 simulations with similar parameters. In the first group, we have experimented with 5 medical porters with the skill "health monitoring", 5 medical porters with the skill "victim handling" and 5 victims, in the second group with 15 of each, and in the third group with 20 of each.

Figure 3 shows the result for the third experiment. As we can see, the scenario S_2 is faster than the classical scenario S_1, except for the lowest field of vision. For a field of vision from 8 to 30, S_2 run-time curves (dotted curves) are below S_1 run-time curves (solid curves). The increase of the field of vision improves the knowledge of the agents on the environment and should improve their efficiency: the probability for an agent to perceive victims and other agents increases. This is true with the use of the environment in the scenario S_2, a larger field improves the efficiency of the agents: as the field of vision increases from 5 to 30, run-time decreases from around 4000s to 800s. But with the classical activation process in the scenario S_1, the improvement of victim perception barely offsets the cost of local context analysis. We can observe that in most cases, the cost of the context computing in each agent is more expensive in terms of simulation run-time than the cost of computing the context

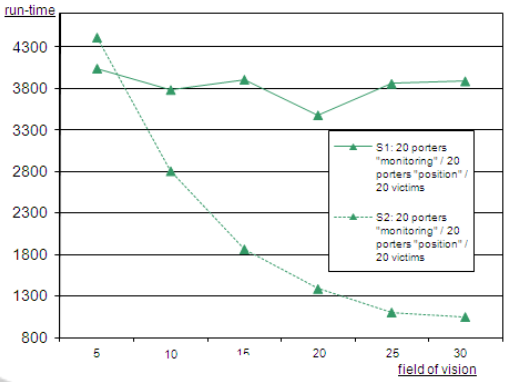

Figure 3: Simulation run-time with the strategies S_1 and S_2 for the third group.

inside the environment. The results are similar for the first and second group of experimentations.

Communication. In the following experimentation, we compare the use of broadcast and the use of the environment for the communication. In these tests, only the communication filters f_{recept} and f_{accept} of the scenario S_4 are evaluated and compared to the broadcast of the scenario S_3. The message exchange to find partners has been instantiated:

- The agents send "request" messages. The agents interested in these messages are those which are close to the sender and have a different skill, if the victim can be moved. In scenario S_3, the message is broadcasted to all the agents, while in S_4, they are managed by f_{recept}.
- Those agents answer with either an accept or a reject message through an addressed message. In scenario S_3, the message is sent to the agent via point-to-point communication, while in S_4, they are managed by f_{accept}.

Each data is an average on 50 simulations. Each simulation is composed of 2000 steps. We have run two series of simulations characterized by two parameters: number of agents and number of updates during the simulation.

Firstly, we have tried to use the RETE algorithm to implement the communication filter, and compared it to the broadcast and addressed messaging capabilities of MadKit. The following table sums up the simulation run-time in function of the messaging implementation: EASI with a RETE tree, MadKit Messaging support, *ad hoc* Environment and Broadcast.

Agent	RETE	Madkit	Environment	Broadcast
10	4624	1096	125	173
20	15215	4050	394	596
50	83785	26306	2377	4755

These results show that the cost of a RETE tree offsets the gains in the message treatment. This discrepancy with the previous results for activation can be explained by the cost of the addition and removal of the messages. In the activation part, the entities are the same all along the simulation, and only rule firing and description modifications take place.

This has lead us to use an *ad hoc* implementation of the environment for communication, which treats the calculations in the same way as described in section 4. However, the comparative results show that the MadKit implementation of the broadcast and addressed messaging is not efficient because of its genericity. Therefore, we also re-implemented the broadcast and point-to-point communications using the same data structures as the environment, in order to have a fair comparison between the environment and the broadcast solutions.

In the following, we compare the two last solutions. Firstly, we study the impact of a variation in the number of agents. The results shown in Fig. 5 feature 10 to 100 agents. The curve shows a clear advantage of the environment over the broadcast, and that the more agents there are, the more EASI is comparatively interesting (from 27.8% for 10 agents to 53.9% for 100 agents).

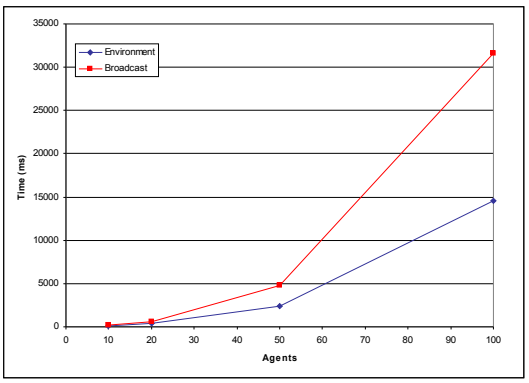

Figure 4: Comparative results of Broadcast and EASI in function of the number of agents.

In the second phase (Fig. 5),we study the impact of a variation in the number of updates, for 50 agents. During each update, all the agents move and in the case of the Environment implementation, they modify their description accordingly. The run-time gain of EASI over broadcast gets from 50.7% for 20 updates to 48.8% for 1000 updates. Therefore, the cost of the update mechanism is not significant in comparison with the difference in efficiency.

These experimentations show a clear advantage to the environment over the broadcast, and this result is backed up by the theoretical results. It shows the efficiency of the context computing inside the environment in comparison with the classical approach where the context is computed in each agent.

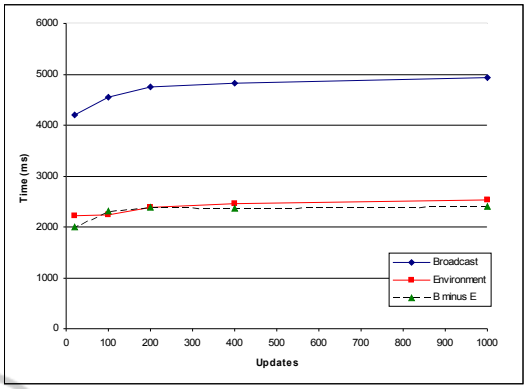

Figure 5: Comparative results of Broadcast and EASI in function of the number of description updates.

6 DISCUSSION AND CONCLUSIONS

In the EASI/EASS models, the environment offers a technical support for communication and activation. Its processing is the result of the filters triggering according to the MAS descriptions. In that way, the agents add or remove dynamically filters and update their description. Thanks to the filter triggering, an agent receives a message or is activated in a specific context. The processing of the messages and the action execution remain to the agents responsibility.

This paper is focused on the cost of the centralization, which is closely related to the cost of the update process. Organizations, such as the agent-group-role model (Ferber et al., 2002) enable to decrease the cost but do not take into account ambient criteria like the location of the agents (e.g. in the simulation). Let us note that our modeling enables to reproduce a selection of percepts according to the organisation. Institutions (see e.g. (Esteva et al., 2004)) generally do not share the same objective. The focus is on control and not on the filtering / matchmaking. However, the technical solutions may be close to ours.

The use of a shared knowledge has already been done within agents (for example Sycaras work, e.g. (Sycara and Wong, 2000)) but in that case the update of the properties is not considered. In this paper, we focus on the cost of the update process that could limit the interest to centralize specific information. Furthermore, middle-agents are generally used only as a first step to find contacts, and not to manage all the communications. In our view, the environment

is a facility, which can be used to facilitate the interaction, apply norms, or verify some rules related to the application design. These roles do not belong to the same design level as the agents.

Theoretically, we have shown that if the communication takes the context into account, there is no strictly dominant solution. It depends on the dynamicity of the multi-agent system, the number of agents and the average percentage of agents interested in each message. According to the number of tests criteria, we have shown that the environment is always better than the local context computation. According to the number of messages criteria, the result has to take into account the number of messages related to the MAS activity and the number of messages related to the update process. We have shown that the environment solution is generally better to mediate the communication of the MAS activity and that few messages to mediate are needed to compensate the cost of the update process.

To propose an empirical assessment of the cost of the environment, we have studied the run-time criterion in the crisis simulation example. We compare the cost of the local context analysis for each agent to a central and global control ensured by the environment, and the cost of communication. The main conclusion is that the environment cost is significantly lower than the local agent calculation of the context perception, except when there are very few agents.

In the future, we intend to investigate different ways to improve the environment performance. An ongoing effort concerns the theoretical evaluation of a RETE-based instantiation of the model. We also study how to take advantage of the filter and entity structures to speed up the matching process.

REFERENCES

Badeig, F., Balbo, F., and Pinson, S. (2007). Contextual activation for agent-based simulation. In *ECMS'07*, pages 128–133.

Bousquet, F., Bakam, I., Proton, H., and Page, C. L. (1998). Cormas: Common-pool resources and multi-agent systems. In Pobil, A. P. D., Mira, J., and Ali, M., editors, *IEA/AIE (Vol. 2)*, volume 1416 of *Lecture Notes in Computer Science*, pages 826–837. Springer.

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

Esteva, M., Rodriguez-Aguilar, J., Rosell, B., and Arcos, J. (2004). Ameli: An agent-based middleware for electronic institutions. In Jennings, R., Sierra, C., Sonenberg, L., and Tambe, M., editors, *Proceedings of the third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, pages 236–243. ACM Press.

Ferber, J., Gutknecht, O., Jonker, C., Muller, J., and Treur, J. (2002). Organization models and behavioural requirements specification for multi-agent systems.

Forgy, C. L. (1982). Rete : A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19:17–37.

Gutknecht, O. and Ferber, J. (2001). The madkit agent platform architecture. In *Revised Papers from the International Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55. Springer-Verlag.

Kesaniemi, J., Katasonov, A., and Terziyan, V. (2009). An observation framework for multi-agent systems. *Autonomic and Autonomous Systems, International Conference on*, pages 336–341.

Laberge, D. (1975). *learning and cognitive processes (Vol. 4)*, chapter 5 - Perceptual Learning and Attention, pages 237–273. W.K. Estes.

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527.

Picco, G. P. and Buschini, M. L. (2002). Exploiting transiently shared tuple spaces for location transparent code mobility. In *COORDINATION '02: Proceedings of the 5th International Conference on Coordination Models and Languages*, pages 258–273, London, UK. Springer Verlag.

Platon, E., Sabouret, N., and Honiden, S. (2005). Overhearing and direct interactions: Point of view of an active environment. In *Proceedings of Environment for Multi-Agent Systems, Workshop held at the Fourth Joint Conference in Autonomous Agents and Multi-Agent Systems*, volume 3830 of *Lecture Notes in Artificial Intelligence*, pages 121–138. Springer Verlag.

Saunier, J. and 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.

Sycara, K. and 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)*, pages 465–466, Washington, DC, USA. IEEE Computer Society.

Weiss, G., editor (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, USA.

Weyns, D., Omicini, A., and Odell, J. (2007). Environment as a first-class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1).

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