Engineering Applications of Artificial Intelligence 70 (2018) 130–141

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

The Information Flow Problem in multi-agent systems

Luis Búrdalo, Andrés Terrasa, Vicente Julián *, Ana García-Fornes

Departamento de Sistemas Informaticos y Computacion, Universitat Politecnica de Valencia, 46022 Valencia, Spain

ARTICLE INFO

Keywords:
Multi-agent systems
Communication strategies
Agent communication
Information exchange

ABSTRACT

One of the problems related to the multi-agent systems area is the adequate exchange of information within the system. This problem is not only related to the availability of highly efficient and sophisticated message-passing mechanisms, which are in fact provided with by current multi-agent platforms, but also to the election of an appropriate communication strategy, which may also greatly influence the ability of the system to cope with the exchange of large amounts of data. Ideally, the communication strategy should be compatible with how the information flows in the system, that is, how agents share their knowledge with each other in order to fulfill the system-level goals. In this way, MAS designers must deal with the problem of analyzing the multi-agent system with respect the communication strategy that best suits the way the information flows in that particular system. This paper presents a formalization of this problem, which has been coined as the *Information Flow Problem*, and also presents a complete case study with an empirical evaluation involving four well-known communication strategies and eight typical multi-agent systems.

1. Introduction

The Multi-Agent Systems (MAS) paradigm is progressively becoming one of the most successful paradigms for developing complex applications, especially in distributed scenarios where communication among different entities is one of the key features. The paradigm is mainly based on the use of cooperative agents, where each agent handles some particular knowledge and a small set of specialized tasks, and is able to cooperate with the other agents in order to achieve some system-level goals, which produces a high degree of flexibility (Gruver, 2004). In fact, it is the social behavior of agents and how they interact, more than their individual capabilities, what makes multi-agent systems so powerful and versatile in many scenarios (Búrdalo et al., 2011). Lately, multi-agent systems have become increasingly sophisticated, with a growing potential to handle large volumes of data and to coordinate the operations of many organizations (Helsinger et al., 2004). In this context, one of the problems related to this distributed computing is the *exchange of information* within the system. Thus, the multi-agent architecture must necessarily provide a robust communication layer with appropriate message-passing mechanisms that enable the interaction processes, since they condition how the intelligent agents are able to interact and coordinate with each other.

However, nowadays the availability of such message-passing mechanisms is not the main issue to consider since, in fact, many current systems are already provided with highly efficient and sophisticated mechanisms. In real systems, the election of an appropriate communication *strategy* may also greatly influence the ability of the system to cope with large amounts of data, especially in open systems with a large number of agents that may dynamically enter or exit the system. Ideally, the communication strategy should be compatible with how the information *flows* in the system, that is, how agents interact and share their knowledge with each other in their way to achieve the system-level goals. Conversely, a strategy badly adapted to such information flow may produce a significant communication effort, which in turn may hinder the agents' cooperation. We have coined this problem as the *Information Flow Problem* (IFP), which is defined as how to exchange information in the most efficient and effective way in a multi-agent system, depending on the characteristics of the system. Basically, the IFP is related, first, to identifying the significant characteristics of multi-agent systems related to the information exchange, and second, to be able to relate the values of such characteristics in a particular multi-agent system to the communication strategy that best suits the way the information flows in that particular system.

In this context, the study presented in this paper proposes a generic formalization model of the IFP, the instantiation of the model in order to define some MAS scenarios or scenes, each with a typical information flow, and an empirical analysis comparing the performances of four well-known communication strategies in these scenes. For the empirical

* Corresponding author.
E-mail addresses: lburdalo@dsic.upv.es (L. Búrdalo), aterrasa@dsic.upv.es (A. Terrasa), vinglada@dsic.upv.es (V. Julián), agarcia@dsic.upv.es (A. García-Fornes).

https://doi.org/10.1016/j.engappai.2018.01.011
Received 14 December 2016; Received in revised form 18 January 2018; Accepted 24 January 2018
Available online 9 February 2018
0952-1976/

analysis, an evaluation framework has been developed. This framework first generates a series of synthetic test multi-agent applications according to some predefined characteristics in the model, and then executes each application on a real MAS called Magentix2, once for each communication strategy. For each execution, the framework collects some run-time information about the message-exchange process. Then, the paper introduces a set of performance metrics, directly derived from such run-time information, in order to quantitatively analyze and compare the behavior of the strategies.

In addition to the general contributions regarding the IFP, part of the work proposed in this paper can also be considered a step towards solving one of the problems of Agent-Oriented Software Engineering (Jennings, 1999), namely the availability of testing techniques which can provide the MAS developers with appropriate software development processes and tools (Houhamdi, 2011). The idea of these interaction tests is to observe emergent properties, collective behaviors or just to ensure that all agents in a group work properly together. Different approaches have tried to include or derive test cases during the development process. In Carrera et al. (2014), test cases skeletons are automatically generated while developing the MAS, while (Nguyen et al., 2009) or (Thangarajah et al., 2011) try to extend well-known agent-oriented methodologies including some types of scenario testing. In this context, both the evaluation framework and the performance metrics proposed in this paper can be used to analyze the system and to determine to which extent some particular factors or characteristics in a MAS affect the behavior of current, well-known communication strategies. Thus, such tools can be considered a testing environment from which MAS developers can benefit.

The rest of the paper is structured as follows. Section 2 introduces the Information Flow Problem in MAS. Section 3 presents the formalization of the problem. Section 4 uses the formalization in order to describe eight typical multi-agent systems, or scenes. Section 5 introduces the case study, including the description of the four communication strategies under study and the evaluation framework where the strategies have been incorporated; this framework is used to generate and execute several hundred test applications corresponding to the previously defined scenes. Section 6 presents the analysis of the results obtained from these executions, for which some performance metrics are defined. Finally, Section 7 presents the conclusions of the paper.

2. The information flow problem

In multi-agent systems, it is commonly accepted that communication among agents is no longer an issue. Agents usually communicate by some sort of peer-to-peer messaging mechanism, which is provided by the platform middleware and allows any agent to send information to any other agent (or to some sort of agent aggregation) in the system. In this context, this paper focuses not on the mechanism used by agents to communicate with each other, but rather on the information being communicated. In particular, the paper studies how this information or knowledge being shared by agents is exchanged, or *flows*, through the system (Zhuge, 2002). This section presents first a brief review of some of the most relevant information exchange strategies in the literature, and then introduces the concept of *information flow problem* as a general way to analyze the appropriateness of such strategies to different multi-agent system scenarios.

Many strategies in the literature have addressed the information exchange among agents. Some of these strategies have focused on making every piece of information in the system to reach all agents, like *broadcasting* or by using a *blackboard* (Corkill, 1991). These strategies have been suggested in scenarios in which global knowledge is required, or as a straightforward way to ensure that agents are always informed about any particular datum they may need. In scenarios where such global communication is not appropriate (or affordable, in terms of computational resources), and agents need to locate first the particular agents they want to communicate to, middleman (or *middle agent*) strategies have been used, with the middle agent usually being either a *matchmaker* (Sycara et al., 1999) or a *broker* agent (Wong and Sycara, 2000). A third group of strategies are based on the idea of *indirect* communication, such as overhearing, which is defined as indirect interaction whereby an agent receives information for which it is not addressee (Busetta et al., 2001; Legras and Tessier, 2003; Dignum and Vreeswijk, 2004; Kaminka et al., 2002). Proposed in several different scenarios, indirect communication schemes have been used to maintain social situational and organizational awareness (Rossi and Busetta, 2004; Tummolini et al., 2005; Gutnik and Kaminka, 2006), to enable team organizations (Legras Onera, 2002; Legras and Tessier, 2003), to monitor teams in a non-intrusive way (Kaminka et al., 2002; Alberola et al., 2011), to improve information spreading (Maity et al., 2013), and to develop advising systems (Aiello et al., 2002; Busetta et al., 2002). However, most of the times, overhearing is internally implemented by using message broadcasting, which is simple but computationally expensive, and may also be considered conceptually contradictory. This is because the overhearer role, as defined in (Malouf, 1995) for multi-party dialogues, is defined to exist in situations where the sender (speaker) is not always aware of who is receiving its (her) messages, apart from the specified receivers (addressees). An alternative technique for indirect communication is the use of tracing facilities, as it has been recently proposed by Búrdalo et al. (2011). Tracing in multi-agent systems has been traditionally focused on providing human users with debugging or monitoring information (Bellifemine et al., 2007; Collis et al., 1998; Serrano et al., 2012); but some other examples show the usefulness of tracing as a general indirect information scheme available to agents, as in Criado et al. (2013), where norm control in open MAS is performed by using an event-tracing approach.

Other approaches in the literature have focused on the analysis of the communication process within multi-agent systems, with the purpose of helping MAS designers by offering tools capable to optimize the communication processes among the designed agents. One of the most well-known approaches is Goldman and Zilberstein (2003), which tries to value communication decisions of agents by using minimum time as a performance-based metric. There are more recent approaches, such as Althnian and Agah (2015), where authors propose a genetic algorithm-based approach for learning the most appropriated communication strategy. In Wei et al. (2014), the performances of different coordination protocols for cooperative multi-robot teams are compared. Another interesting work is proposed in González-Pardo et al. (2010), where authors search for an optimal communication topology in order to avoid overhead in the communication process among agents. Finally, a proposal for computing optimal communication policies is presented in Chakraborty and Sen (2007). Most of the proposed solutions are static, theoretical models that typically need to be specified by defining closed environments where a lot of information about the communication processes must be fixed and known a priori. Moreover, most of the analyzed solutions also assume some constraints and requirements which limit their use in more open or dynamic environments. Conversely, the proposed model presented in this paper is generic enough to be used in the specification of a wide range of systems, including open and dynamic ones. Another advantage of this proposal, as explained below, is the evaluation framework associated with the model, which can be used to observe emergent properties in the communication flows resulting from agent interactions, and to make sure that a group of agents work correctly together according to the expected information flow.

On a separate, but related, issue, MAS have also been used as a tool to study the behavior of humans organizations by taking into account how information (or knowledge) flows through their members. An example of this is the simulation framework developed in Jolly and Wakeland (2009), where the goal is to examine the result of the interactions between individuals in an organization with different preferences regarding knowledge sharing, using game-theoretic analysis and a Netlogo agent-based simulation model. In a similar way, Zhuge (2006) proposes techniques for planning and simulating the knowledge flow networks of

large teams or strategic alliances within organizations. Another example can be found in Nissen and Levitt (2004), where authors propose the formalization of the dynamics of enterprise knowledge flows and then simulate it by using agent-based computational models. Information or knowledge flow has also been studied in social networks, which can be seen as particular kind of large-scale human organizations. These studies try to analyze the evolution of user interactions through the tracing of the user actions in the social network. There are contributions related with this issue in several different areas, including marketing (Leskovec et al., 2007), outcomes prediction (Borondo et al., 2012), recommender systems (Guimerà et al., 2012), analysis of potential consumers (Ahn et al., 2007) or detection of the most influential users (Romero et al., 2011). According to del Val et al. (2015), the analysis of the evolution of a social network makes possible to study the dynamics that are associated to interactions among users on a global scale. In general, the many similarities between the analysis of the communication processes in human organizations (and social networks) and in MAS justifies the importance of the information flow problem now introduced in this paper.

We define the *Information Flow Problem* (IFP) as the problem of how to exchange information in the most efficient and effective way in a multi-agent system, depending on the characteristics of the system. As it will be presented in the paper, part of the IFP also consists on identifying such characteristics. From the viewpoint of the IFP, communication is seen as a process to achieve that the information which some agents want to share is delivered to the agents which are interested in that information. In the simplest case, a multi-agent system with only two agents that know in advance each other's interests, the IFP is trivially reduced to the two agents being able to send messages to each other. In more complex scenarios, however, when there are several agents with different and dynamic information requirements, the problem is not trivial. For example, a multi-agent system may use broadcasting as a way to ensure that potentially important information reaches all agents. If all agents are interested in the information being broadcast, this may be a good choice; if, however, the number of agents interested in any particular piece of information is reduced to a few, this can be an extremely inefficient way to communicate. Another complex example may be a large, open multi-agent system, where new agents need to register and then subscribe to the information that they need, by using some sort of matchmaker service. According to how dynamic are both the arrival of new agents and their information requirements, and how well (and fast) the matchmaker is able to put the right agents in contact, information being produced in the system may not arrive to all the requesting agents, resulting in an ineffective communication scheme.

In theoretical or simulation studies, it may be assumed that the information flow is produced instantly, without loss, and producing zero overhead to the agents. Such studies do not normally consider that communication may be degraded due to some run-time situations, including communication bottlenecks of certain agents or system components, as well as concurrency conditions or middleware and hardware errors, which may require to retransmit information. However, in real systems, such simplifications do not hold and so, the IFP becomes relevant. The goal of the IFP is to be able to analyze the system according to some predefined attributes or properties related to information exchange, and then to infer which communication scheme better fits the system requirements.

3. Formalization

From the viewpoint of the IFP, a MAS is comprised by set of entities which exchange information that can be classified in different subjects, or topics. The term *entity* includes the concept of autonomous agent, plus any other source or target of information in the MAS (such as a human operator, for example). On the other hand, the concept of *topic* is introduced by convenience, because considering the particular subjects or topics about which agents (and other entities) are communicating about allows for a more precise analysis of the way information flows in the MAS. For example, there may be much more interactions about some topics than others, the particular information about each topic may have a different, distinctive way of flowing (such as bottom up or top down, one to many or many to many, etc.), the communication effort of agents can be directly related to the number of simultaneous topics that agents are maintaining, etc.

Thus, this formal model defines a MAS, from the viewpoint of the IFP, as a tuple composed by E, the set of all entities in the MAS, and T, the set of all topics about which entities in E exchange information:

$$MAS \equiv \langle E, T \rangle \tag{1}$$

The information which is exchanged within the MAS is what is referred to as *information flow* in this paper. In order to analyze how the flow is produced, it is necessary to establish a relationship between each entity $e \in E$ and every topic $t \in T$. This relationship is set by means of the "*source*" and "*receiver*" functions, defined as follows in Eqs. (2) and (3):

$$source(e,t) = \begin{cases} \text{true:} & e \text{ transmits information} \\ & \text{related to the topic } t \\ \text{false:} & \text{otherwise} \end{cases} \tag{2}$$

$$receiver(e,t) = \begin{cases} \text{true:} & e \text{ receives information} \\ & \text{related to the topic } t \\ \text{false:} & \text{otherwise} \end{cases} \tag{3}$$

The set of all the topics an entity e transmits information about at least once during its life time (*source topics*) can be defined as follows:

$$T_S(e) \equiv \bigcup t_i \ : \ source(e, t_i) \tag{4}$$

In the same way, the set of all the topics an entity e receives information about at least once during its life time (*receiving topics*) can be defined in the following way:

$$T_R(e) \equiv \bigcup t_i \ : \ receiver(e, t_i) \tag{5}$$

The subset E_S of all source entities in E, is defined as the union of all entities in the MAS which transmit information about any topic:

$$E_S \equiv \bigcup_{e_i \in E} e_i \ : \ T_S(e_i) \neq \emptyset \tag{6}$$

In the same way, the subset E_R of all receiver entities in E is defined as the union of all entities in the MAS which receive information about any topic:

$$E_R \equiv \bigcup_{e_i \in E} e_i \ : \ T_R(e_i) \neq \emptyset \tag{7}$$

From the point of view of the IFP, only the entities which either transmit or receive information (or both) are producing any information flow, and thus, the set of all entities in the MAS can also be defined as the union of source and receiver entities:

$$E \equiv (E_S \cup E_R) \tag{8}$$

Furthermore, the definition of MAS according to the IFP logically restricts the system to be generating and consuming some information, hence the following condition is also assumed:

$$E_S \neq \emptyset \quad \wedge \quad E_R \neq \emptyset \tag{9}$$

Taking into account the basic definitions above, the following subsections introduce several properties or *factors* describing the MAS from the perspective of the IFP. The goal of these factors is to be able to express *quantitatively* the most relevant properties of the information exchange in the MAS, and hence, to *characterize* the MAS from the viewpoint of the IFP. Some of these properties are related to the

characteristics of the information itself (for example, how homogeneous or distributed it is among the MAS entities), while some others are centered on how entities behave in respect to the information they possess or need (for example, how specialized or exclusive they are about their own information).

The factors now introduced are in all cases defined for both the source and the receiver entities in the MAS. However, for the sake of brevity, only the factors corresponding to source entities are discussed. So, please consider that, for each definition or equation corresponding to source entities (with S subindex), there is an analogous definition for receiver entities (with R subindex).

3.1. Distribution factors (D_S, D_R)

The first property is related to how distributed the information and the requirement of information are among the entities in the MAS. Both distribution aspects will condition how information will flow across the system (for example, if information is generated by a small percentage of agents, and required by the rest of them, then the flow will be "few to many").

In order to model these aspects, a *Distribution Factor* (D_S) is defined for source entities, in the following way: D_S represents the ratio between the number of source entities and the total amount of entities in the MAS (in the equation, the cardinality of a set A is represented as $|A|$, as usual):

$$D_S = \frac{|E_S|}{|E|} \quad \text{with } D_S \text{ in }]0..1] \tag{10}$$

According to this definition, the more distributed the source entities are in the system, the higher the D_S factor will get. As for the extreme values of the range, a value of $D_S = 1$ would mean that all entities in the MAS are generating information to some other entities, while $D_S \to 0$ would mean that practically no agent is sending information. The value $D_S = 0$ is out of range, because a MAS in which no entity is sending (or receiving) information is by definition not possible according to the IFP (as expressed by Eq. (9)).

3.2. Specialization factors (S_S, S_R)

This second set of factors is related to the specialization of the information that entities possess or require from other entities in the MAS. For example, some entities may be highly *specialized*, in the sense that they transmit or receive information only about very few topics, while others may be less specialized, meaning that they have a broader set of topics that they talk about. This factor also influences the information flow: for example, if the population of receiver entities in the MAS are highly specialized, the flow will mostly be "point to point", whereas a MAS with low specialized receiver entities will produce a flow similar to "broadcasting".

In order to be able to quantify these differences in the interests of entities, a *Specialization Factor* for source entities (S_S) is defined as 1 minus the average for all entities in E_S of the ratio between the number of topics about which the entity is transmitting information and the number of topics in the MAS, that is:

$$S_S = 1 - \frac{1}{|E_S|} \sum_{e_i \in E_S} \frac{|T_S(e_i)|}{|T|} \qquad \text{with } S_S \text{ in } [0..1[\tag{11}$$

In this case, the values of this factor is in the range $[0, 1[$. A higher value of the Specialization Factor means that a higher number of sender entities are more specialized, in the sense that they send information about few topics (out of the total amount of topics in the system). The extreme values of the range have the following meaning: on the one hand, $S_S = 0$ happens when every sender entity in the system sends information about every existing topic, which makes the sum equal to $|E_S|$, and thus, the fraction equal to 1. On the other hand, the highest possible value ($S_S \to 1$) happens when information about each topic is only being sent by one entity. In this particular case, the expression is reduced to $S_S = 1 - \frac{1}{|E_S|}$. This expression calculates the highest value of S_S in any given system, which depends on the system's number of sending entities ($|E_S|$).

3.3. Topic exclusivity factors (X_S, X_R)

The factors introduced in this section, along with the ones introduced below in the following section, are related to the popularity of topics among entities, either for sending or receiving information. The previous section was focused on the amount of topics each entity is interested on. This can be further refined by considering to which extent the different entities in the MAS share their interest in particular topics. For instance, some topics may be broadly shared by many entities which send and/or receive information about them, while some others can be used in a more *exclusive* way by only a few entities, or even by a single one.

So, in order to quantify to which extent topics are used by entities in the MAS in an exclusive way, the Topic Exclusivity Factor for source entities (X_S) is now introduced, after some previous definitions.

The set of *exclusive topics* about which an entity $e_i \in E$ is sending information is named $T_S^X(e_i)$, and it is defined as follows:

$$T_S^X(e_i) \equiv \bigcup_{t_j \in T_S(e_i)} t_j : \nexists e_k \in E, i \neq k | t_j \in T_S(e_k)$$

$$T_S^X(e_i) \subseteq T_S(e_i)$$

From this set, it is possible to define X_S as the average for all source entities (in E_S) of the proportion of the topics of each source entity that are exclusive, that is:

$$X_S = \frac{1}{|E_S|} \sum_{e_i \in E_S} \frac{|T_S^X(e_i)|}{|T_S(e_i)|} \qquad \text{with } X_S \text{ in } [0..1] \tag{12}$$

So, X_S calculates how many of the different topics used by a source entity are not shared with any other source entity. Since the value of X_S represents an average of proportions, it is in the range $[0, 1]$. A higher value of X_S denotes a MAS where, in average, there is a higher proportion of topics to which entities are sending information in an exclusive way. In the extreme values of the range, $X_S = 0$ indicates that there is no sender entity which has an exclusive topic, while when $X_S = 1$, all the topics of every sender entity in the MAS are exclusive, that is, $\bigcap_{e_i \in E} T_S(e_i) \equiv \emptyset$.

3.4. Topic overlapping factors (O_S, O_R)

This section, which complements the previous one, is related to non-exclusive topics. Non-exclusive topics are by definition, topics shared by two or more entities. However, from the perspective of the IFP, it is interesting to be able to determine the degree of topic sharing in the MAS, since the flow may greatly vary if a significant amount of topics are shared by most of the system entities, rather than by a few of them. Thus, the Topic Overlapping Factor for source entities (O_S) is here introduced, after some preliminary definitions.

The set of shared topics about which an entity $e_i \in E$ is sending information is named $T_S^H(e_i)$ and, based on previous definitions, can be defined as the set of source topics of e_i minus the set of exclusive source topics of e_i, as follows:

$$T_S^H(e_i) \equiv T_S(e_i) \setminus T_S^X(e_i) \tag{13}$$

Then, the set of all topics about which more than one entity in the MAS is sending information can be obtained as the union of all of these shared topics for every source entity in the MAS:

$$T_S^H \equiv \bigcup_{e_i \in E_S} T_S^H(e_i) \tag{14}$$

It is also useful to define the set of all source entities which share a given topic t_i. This set is named $H_S(t_i)$, and it can be defined as follows:

$$H_S(t_i) \equiv \bigcup_{e_j \in E} e_j : t_i \in T_S^H(e_j) \tag{15}$$

With all these definitions, the *Topic Overlapping Factor* of a source entity $e_i \in E_S$, named $O_S(e_i)$, which calculates to which extent its

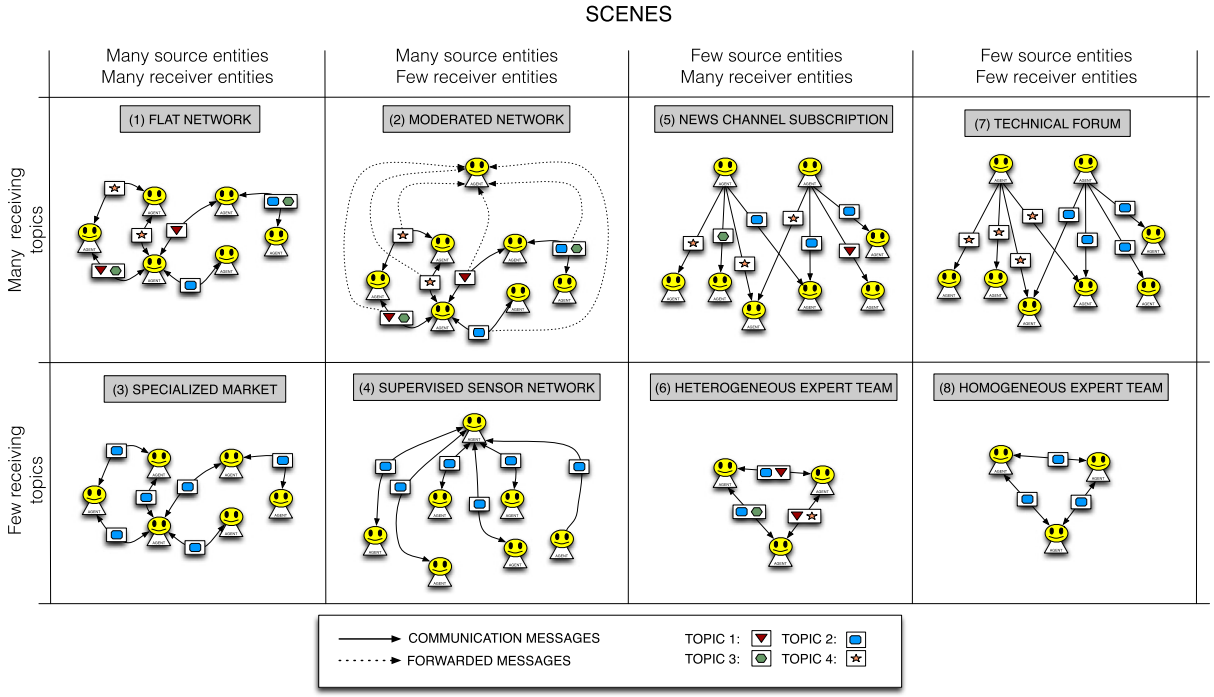

Fig. 1. Examples of different MAS scenes, according to different values of T_r (in rows), and E_S and E_R (in columns).

topics are shared with other source entities, can be introduced. This is calculated as the average for all its shared topics (in $T_S^H(e_i)$) of the ratio between the number of entities which share each topic and the number of source entities in the MAS, E_S. This is:

$$O_S(e_i) = \frac{1}{|T_S^H(e_i)|} \sum_{t_j \in T_S^H(e_i)} \frac{|H_S(t_j)|}{|E_S|} \tag{16}$$

Finally, it is possible to naturally generalize the previous expression to include all source entities in the MAS, in the following terms. The *Topic Overlapping Factor* of source entities in the MAS, O_S, is calculated as the average topic overlapping values for all source entities:

$$O_S = \frac{1}{|E_S|} \sum_{e_i \in E_S} O_S(e_i) \qquad \text{with } O_S \text{ in } [0..1] \tag{17}$$

For any particular MAS, the value of O_S represents the average (for all source entities) of the average "sharing proportion" of the *shared* topics in each source entity ($O_S(e_i)$). That is, for each source entity e_i, only its shared topics are considered, and the value $O_S(e_i)$ represents the average proportion of other source entities that also send information about these shared topics. Thus, values of O_S are in the range $[0, 1]$. When comparing two given MASs, a higher value of O_S implies a MAS where the shared topics of source entities are, in average, shared among a higher number of other source entities, that is, the degree of topic overlapping for shared topics in source entities is higher. In the extreme values of the range, a value of $O_S = 0$ is produced in a system where there is no topic sharing among source entities, while a value of $O_S = 1$ means that for each source entity in the MAS, any shared topic is shared with every other source entity.

4. Description of possible scenes

This section illustrates the capabilities of the theoretical framework above in order to describe several multi-agent systems with very different behaviors with respect to their respective information flows. This description can be carried out in the design phase of systems where at least there is some a priori knowledge about how the information flow should be, which covers a very wide range of possible systems to be described. The aim of the section is not to be comprehensive, but on the contrary, to show that with some simple, discrete variations of a few basic characteristics of the MAS according to the model, it is possible to describe many well-known multi-agent system scenarios, or *scenes*.

In particular, the study has considered three basic characteristics of the MAS (the cardinalities of E_S, T_S and E_R), and two values for each one (*many* and *few*). These two values have been chosen to be broad, representative values of the characteristics, but in any case, they represent values that differ in one order of magnitude at least. Then, a typical, well-known MAS scene has been selected as one representative system of each of the eight resultant value combinations.

These eight scenes are graphically depicted in Fig. 1, where they are distributed with varying number of C_S in rows, and varying number of E_S and E_R in columns. The following subsections discuss these scenes in more detail, introducing in each case the particular factors defined in the model which are relevant in order to describe the scene's characteristic information flow.

4.1. Flat network

A flat network of agents, like the one shown in Fig. 1(1), is a MAS where many different agents possess and require information about several different topics, as for example, a multiproduct market, where several sellers and customers are interested in exchanging goods of different types, with each type being a different topic about which agents talk about.

In such scenario, some factors in the model can further reflect particular cases. For example, if most of the products in the market were exclusive of some sellers (or sellers specialize in certain brands of the same product), then the MAS would show a high value of X_S. The existence of a few popular or basic products that may be sold by many sellers would increase the value of O_S, while a low value in this factor would imply that there are no such general products shared by most sellers. On the other hand, from the viewpoint of the buyers, the existence of some trendy products that may be of interest for many customers would be reflected by a high value of O_R, while a high value of X_R would imply that customers have almost disjoint sets of products of interest.

4.2. Moderated network

In moderated networks, like the one in Fig. 1(2), some regular agents communicate with each other about some topics, while a few supervisor (or moderator) agents receive all the information that the regular agents exchange (in the Figure, there is only one moderator agent, the one at the top). Thus, supervisor agents can be seen as receiver entities which are interested about many different topics and from many different sources, while the rest of (regular) agents could, for example, be modeled as a Flat Network described above.

In a moderated network, a high value of X_R and a low value of O_R would describe a system where supervisors are organized in order to moderate a few topics each, while if the entire moderation process were carried out by all supervisors regardless of the topics, then $X_R = 0$ and $O_R = 1$.

4.3. Specialized market

A specialized market is a particular case of the general market described in Section 4.1, where there are few types of products (or brands) to exchange. Thus, in a specialized market, there may be many different agents that buy or sell such products, and hence communicate to each other about them, but messages refer to few topics. Fig. 1(3) depicts the particular case where there is only one product/topic about which agents communicate.

In a specialized market, the values of S_S and S_R are related to the specialization degree of sellers and buyers, respectively, about the available products (e.g., a high S_R would imply that each buyer is interested in a few of the available product types). In addition, the X_S and X_R factors can describe the proportion of product types that sellers or buyers share, respectively (e.g., a high X_S indicates that each product type is sold by few sellers).

4.4. Supervised sensor network

A sensor network is a MAS where a group of several sensor agents register information which typically is not much diverse (that is, it relates to some few different topics), and then send this information to a reduced group of supervisor agents, which filter and process it. Fig. 1(4) depicts a very simplified supervised sensor network, with only one supervisor and some sensor agents, all sending information about a single topic.

Supervised sensor networks are characterized by a high D_S and a low D_R, since information flows from the many sensor agent to the few supervisor agents. In addition, other factors can describe particular cases within this scene. For example, S_S can distinguish between systems where sensor agents are specialized, each sending information about a different data set ($S_S \to 1$) from other systems where sensor agents are capable of informing about a wide range of data sets ($S_S \to 0$).

4.5. News channel subscription

In a news channel subscription scenario, some expert agents (typically a few) share information (news) about different topics with other agents (typically many) by means a of subscription mechanism, where each topic is a possible subscription channel. This scenario is represented in Fig. 1(5), where the two expert agents at the top feed the other agents about some different topics.

This scenario can be further analyzed by considering the X_R and O_R factors. For example, if X_R is high, this means that many channels have single (or very few) subscribers, while the value of O_R informs about the popularity of the channels with more than one subscriber.

4.6. Heterogeneous expert team

A heterogeneous expert team is a MAS where a reduced group of agents discuss about several topics. This is depicted in Fig. 1(6) by three agents broadcasting and receiving messages about different subjects.

Typically, in this type of MAS, all the agents in the team are usually interested about (almost) every topic, and so both S_R and X_R are usually very low, and O_R is high. The values of the equivalent factors for source agents will basically depend on the amount of topics that each agent is expert about. If, in general, all agents know about all topics, then S_S and X_S will also be low, and O_S high, since there is a common benefit in sharing all the information that each agent possesses.

4.7. Technical forum

A technical forum may be seen as a special case of the one described in Section 4.5, where the expert agents are providing information about few, specialized topics. In Fig. 1(7), this is depicted as two expert agents at the top, each one providing information about a different topic, to some other agents that need that information.

In this scenario, S_S is usually high, since information about each topic is normally provided by its own expert agent, and the information flow is greatly influenced by the amount of agents interested in each topic. So, probably the most relevant factor in this type of MAS is S_R, since a low value of S_R would mean that agents require information about several different topics, hence producing a great deal of information flow.

4.8. Homogeneous expert team

A homogeneous expert team is a particular case of the heterogeneous expert team described in Section 4.8. In a homogeneous expert team, a reduced group of agents discuss about one topic (or a few topics, at most). This is represented in Fig. 1(8) by three agents that exchange information about a single topic.

A MAS of this type will typically have values of S_S and S_R close to zero, since every agent will talk about all the (few) topics with each other. Consequently, the values of both X_S and X_R will be very low, and the O_S and O_R will usually be high.

5. Case study

This section presents a case study where the scenes introduced in the previous section are tested against four well-known communication strategies, which are widely used in many multi-agent systems. The section introduces first the four strategies, illustrating their typical behavior by explaining their respective solution to an example of a simple communication problem. Then, the case study itself is presented. The study comprises several tests where some variations of the eight scenes have been implemented in a real MAS, which has the four communication strategies available, with the general goal of correlating the performance of each strategy to the typical information flow factors of the scenes. The performance results of such tests are later presented in Section 6.

5.1. The communication strategies

We define a simple communication problem with a reduced number of agents and topics. In this case, there are two source agents (or publishers) which can produce information about 4 different topics (numbered from 1 to 4). There are also two receiver agents (or subscribers) which are interested in such topics. In particular, the first subscriber is interested in topics 1 and 2, while the second one is interested in topics 2 and 3. None of the two receiver agents is interested in topic 4. The central part of Fig. 2 shows this trivial example. Ideally, the receiver agents would only receive information about those contexts in which they are interested, and the contexts in which no agent is interested, such as context 4, would not produce any information exchange. For sake of simplicity, we assume that all agents in the case study are benevolent and they send their messages whenever are required.

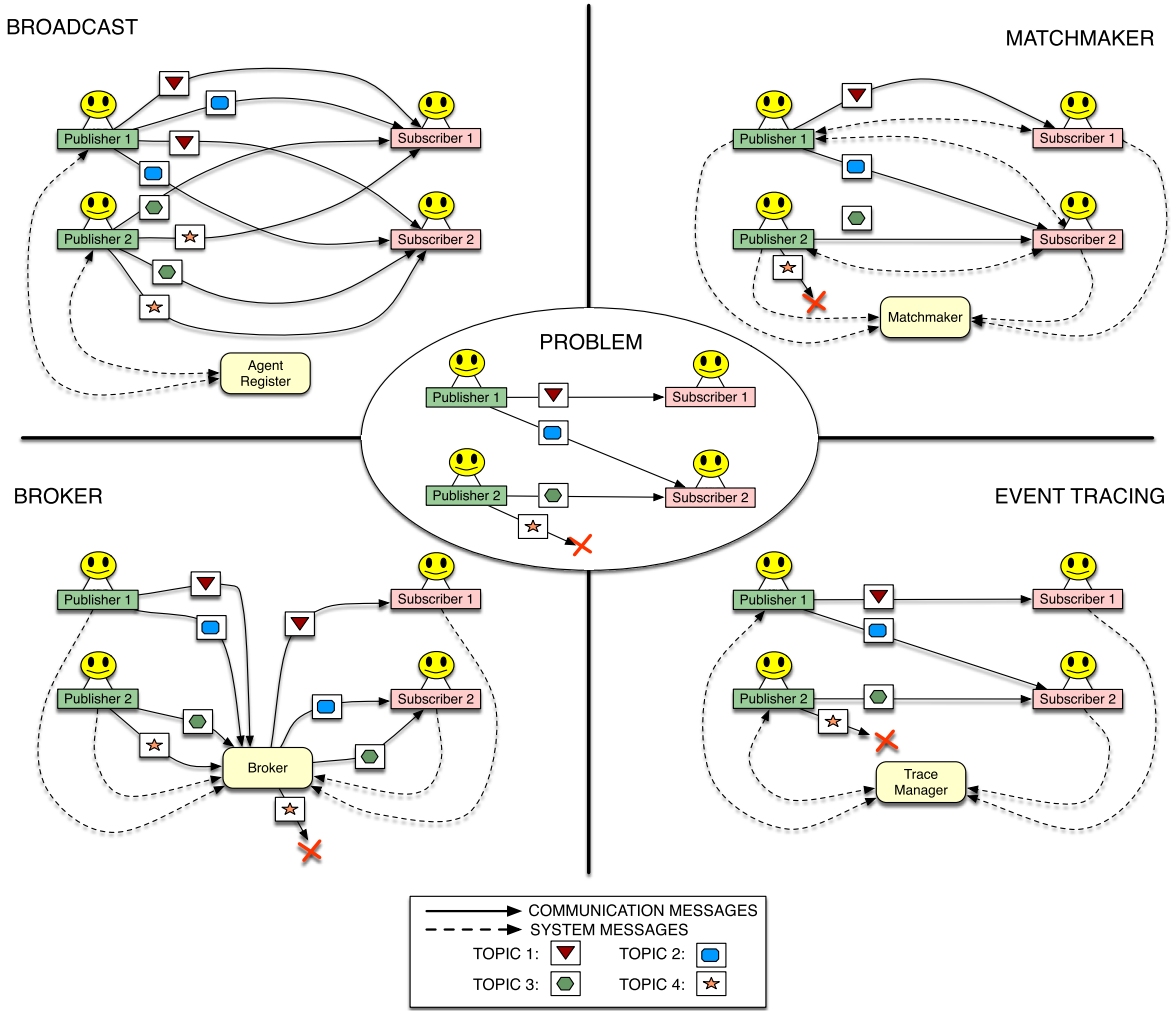

Fig. 2. The communication problem and four solutions, using different communication strategies.

This communication problem can be solved by many different strategies. In particular, the case study introduces four of the best known communication strategies in MAS, each depicted in a different corner of Fig. 2. For each strategy, please note that the picture introduces some sort of "system agent" (a system entity), depicted as a rounded box. This is because such entities are required to support the corresponding strategies in any real MAS. As it will be presented in the following sections, such entities have been incorporated to the testing framework and their influence in the information exchange process have been particularly analyzed.

The solution with the **Broadcast** strategy (top left) consists of each agent sending each other agent every possible message on any given topic. In this case, the system agent is required at least to register the existing agents, and to provide the list of agents to the source agents(for example, in a typical FIPA platform, this agent would be the so-called Agent Management System, or AMS). In the **Matchmaker** strategy (top right), the system agent is a middle agent, called precisely matchmaker, which gathers information about the topics of interests of source and receiver agents and then is able to put in contact the right source and receiver agents. The **Broker** strategy (bottom left) also uses a system agent as a middle agent, called broker in this case. The purpose of the broker agent is to forward all messages sent by source agents to the appropriate receiver agents, for which it needs to be previously informed about the topics of interest of each source and receiving agent. Finally, the **Event Tracing** (bottom right) strategy uses some tracing facilities available in the multiagent platform in order to provide a subscription mechanism for the so-called trace services. Source agents create trace services for the topics about which they want to send information, and receiver agents subscribe to these services according to their interests. In this case, the system agent ("Trace Manager") is the one that supports the required subscription mechanism.

When comparing the problem description with the proposed solution of any of the strategies, it is clear that the strategies often solve the problem in a non-optimal way. For example, in the Broadcasting strategy, the processing and sending of messages about topic 4 is useless for the Publisher 2 agent, and also to both subscriber agents that will have to receive them only to discard them.

5.2. *Implementation of the case study*

This section presents the implementation of the case study, where several multiagent test applications corresponding to the eight scenes described in Section 4 have been generated and executed in an evaluation framework which incorporates the four communication strategies above. The section describes, in order, the evaluation framework, the output data of each test, and the actual set of tests included in the study.

The evaluation framework can be basically seen as a source code generator plus an execution environment, which work sequentially. First, the code generator uses the definitions and factors defined in Section 3 as entry configuration parameters in order to produce the *source code* of several multiagent test applications. Among others, these parameters include the number of publisher and subscriber agents, the number of

topics per agent, and the specialization, exclusivity, overlapping factors for both publisher and subscriber agents. The source code of the agents inside each test application (in Java) is specifically generated in order to use the facilities of a particular multiagent platform called *Magentix2*[1] (Fogués et al., 2010). Then, after being compiled, each test application is executed in the execution environment, which is a modified version of the *Magentix2* platform where the four communication strategies described above have been implemented, and where some tracing facilities allows for the collection of run-time information about the messages exchanged during the execution.

Since the goal of the case study is to measure the performance of the strategies respect to the IFP, the framework generates *synthetic* test applications, each one comprising a set of publisher (source) agents and a set of subscriber (receiver) agents which exchange messages about some topics. As some of the strategies rely on an actual subscription mechanism, topics have been implemented, and are referred to, as subscription *channels*. The applications are synthetic in the sense that the only purpose of their agents is to publish/subscribe to any of the available channels and then to send/receive messages corresponding to such channels, with the actual content of these messages not being relevant. Please note that, in order to work properly, the code of each test application is generated in order to use a particular communication strategy, and this implies that the publisher/subscriber agents will need to get in contact to the corresponding *system agent* that registers both the agents and the available channels and, in some cases, plays the middle-agent role in the strategy. The agents will also need to follow the strategy negotiation in order to contact each other, if required, before being able to exchange messages.

Since agents are by definition dynamic entities which may change their motivations and goals, the framework allows agents to change the channels in which they are interested or about which they generate information, at run time. In particular, in the process of generating the source code of each test application, the framework calculates off line some random changes of the channels of interest for each (publisher or subscriber) agent and the moments at which such changes need to be produced, while keeping the values of the factors (specialization, exclusivity, etc.) of that test application constant. Such changes are then introduced in the code of each agent, ensuring that they will be the same for each application independently of the communication strategy being used, and hence allowing the performance results of the strategies for the same test application to be comparable.

For the sake of simplicity, the set of test applications generated by the framework for this case study has limited each agent to act as either a publisher or a subscriber. However, this limitation does not affect the performance study about the communication strategies, since the study does not consider the internal complexity of agents with respect to the simultaneous conversations they are keeping with each other, but the message exchange itself. However, the system agent for each strategy can indeed send and receive messages, and the communication effort of such agents is actually relevant to the study.

As commented above, the execution of each test application in the execution environment produces some run-time data about the messages exchanged during the execution, both from the publisher/subscriber agents and the strategy system agent. This run-time data can be processed after the execution, in order to compute some statistic results that can be then combined and compared with the results of other executions, as necessary. For each execution, the collected run-time information includes, among others, four types of messages according to their sending and delivery status during the execution: *SNT*, *OK*, *SPAM* and *UNRCV*, which are now described. From the viewpoint of publisher agents, all messages that are sent by any publisher is registered in the agent's log as a *SNT* message. On the other hand, these "sent" messages can be classified in three groups, from the viewpoint of any given subscriber agent. The first group is made of the messages correctly delivered to the subscriber agent according to its interests, that is, messages that correspond to channels in which the agent was interested; messages in this group are labeled as *OK*. In some communication strategies, such as broadcast, it may happen that some messages are delivered to an agent which was not interested in their corresponding channels; such messages form the second group, labeled as *SPAM*. Finally, it may also happen that a sent message is not delivered to a subscriber agent which was actually interested in the message's channel. This may happen, for example, in the matchmaker strategy: when publisher and/or subscriber agents dynamically change their interests at run time, this may produce some changes in the subscriptions, and such changes are not instantaneous, since the matchmaker (system) agent needs to re-negotiate the subscriptions with the affected agents. Messages in this third group (sent but not delivered where they were supposed to) are labeled as *UNRCV*.

Table 1
Summary of the experiment parameters.

(1) Flat network		(2) Moderated network	
# publisher agents	200	# publisher agents	200
# channels per publisher	7	# channels per publisher	7
# subscriber agents	200	# subscriber agents	20
% of interest	50	% of interest	50
(3) Specialized market		(4) Supervised sensor network	
# publisher agents	200	# publisher agents	200
# channels per publisher	1	# channels per publisher	1
# subscriber agents	200	# subscriber agents	20
% of interest	50	% of interest	10, 50, 80
(5) News channel subscription		(6) Heterogeneous expert team	
# publisher agents	20	# publisher agents	20
# channels per publisher	7	# channels per publisher	7
# subscriber agents	200	# subscriber agents	20
% of interest	50	% of interest	10, 50, 80
(7) Technical forum		(8) Specialized expert team	
# publisher agents	20	# publisher agents	20
# channels per publisher	1	# channels per publisher	1
# subscriber agents	200	# subscriber agents	20
% of interest	50	% of interest	10, 50, 80

Finally, the remainder of the section describes the actual set of experiments that have been generated and executed by using the evaluation framework. The framework has generated several test applications for each of the eight scenes described in Fig. 1, and has specified the characteristics of each application by using the definitions and the factors introduced in Section 3. The main input parameters were: the number of publisher and subscriber agents, the number of channels for publisher agents, and the percentage of available channels that were of interest for subscriber agents. Table 1 presents a summary of the particular values of these parameters for the eight scenes.

After some preliminary tests, the experiments assigned some particular values to the defining factors which are typical of each scene, in order to be able to detect, if possible, significant performance differences among the communication strategies. In particular, the cardinalities of both E_S and E_R were assigned to 20 agents ("few") or 200 agents ("many"), and the cardinality of T_S were set to 1 channel ("few") or 7 channels ("many"), according to the characteristic of each scene. Some particular scenes considered up to three possible values of the specialization factor for receiver entities (S_R), in this case expressed (for implementation convenience) as the percentage of the available channels which were of interest for the subscriber agents. In addition, all these scenes were tested with some variations of other parameters and factors: each scene was tested for both 1 and 7 channels per subscriber. The *Exclusivity Factor* (X) was set to 20% and 80% for both publisher and subscriber agents. In the same way, the *Overlapping Factor* (O) for both publisher and subscriber agents was also set to 20% and 80%. For each combination of values in these factors, 5 replicas were generated. This made a total amount of $(8 * 2 * 2 * 2 * 2 * 2 * 5) = 1280$

[1] http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/index.php.

Table 2
Effectiveness results of the experiments with varying numbers of publishers (#Pub), channels per publisher (#Ch/Pub) and subscribers (#Sub).

% effectiveness											
ID	#Pub	#Ch/Pub	#Sub	BCAST		Match		Broker		Trace	
				Avg	Dev	Avg	Dev	Avg	Dev	Avg	Dev
1	200	7	200	100	0	83.79	6.43	93.47	6.12	92.02	7.98
2	200	7	20	100	0	84.70	7.69	93.55	6.27	92.75	6.83
3	200	1	200	100	0	86.09	7.49	93.40	5.91	91.53	6.92
4	200	1	20	100	0	85.42	6.24	91.64	6.89	92.50	7.52
5	20	7	200	100	0	85.26	5.34	91.56	6.42	92.20	7.78
6	20	7	20	100	0	84.63	6.43	94.59	6.22	94.21	5.83
7	20	1	200	100	0	89.82	9.82	93.90	8.95	90.96	12.47
8	20	1	20	100	0	85.09	8.59	89.32	11.07	92.20	7.85

single tests, each of which was executed once for each communication strategy on the Magentix2 multiagent platform for a total execution time of 120 s, during which publication periods were randomly chosen for each channel of every publisher agent between 20 and 25 s.

6. Result analysis

As explained in the previous section, the execution of each test application in the case study produced some run-time logs, from which certain basic values or results can be extracted. In particular, these results include the number of messages sent by publisher agents (N_{SNT}), and the number of messages wanted and delivered (N_{OK}), delivered but unwanted (N_{SPAM}) and wanted but not delivered (N_{UNRCV}) to subscriber agents, as well as the messages sent and received by the system agent which supports each of the four communication strategies under study.

From these basic results, this section proposes five performance metrics in order to evaluate and compare the run-time behavior of the communication strategies in the eight scenes defined in the case study. The following subsections introduce these metrics, which will consider not only different performance aspects, but also the run-time overhead introduced by the strategies.

6.1. Effectiveness

The first performance metric analyzes the ability of each strategy to deliver all the messages which are of interest to subscriber agents. In this sense, an strategy is considered to be completely effective if it is able to deliver each and every message sent through a channel to which any agent was subscribed, during the entire execution. This can be calculated, for each test application execution, in the following terms:

$$\text{Effectiveness} = \frac{N_{OK}}{N_{OK} + N_{UNRCV}} * 100 \tag{18}$$

The equation above defines the *Effectiveness* metric as a percentage value, where a value of 100% indicates that every message of interest for any subscriber agent was delivered (i.e., there were no *UNRCV* messages).

Table 2 shows the *Effectiveness* average and standard deviation values for the different experiments and strategies. The table shows that *Broadcast* is the only strategy which is always 100% effective, since all messages are delivered to all subscriber agents, independently of their interests. In the other three strategies, the dynamic changes of topics (channels) for publisher or subscriber agents may affect the effectiveness, since they produce a reconfiguration of the system (middle) agent and/or the affected agents. In particular the values for both the *Broker* and *Trace* strategies stand between 90% and 95%, depending on the type of system. *Broker* seems slightly better in terms of effectiveness than *Trace* in most of the cases, due to the fact that *Broker* is a pure centralized strategy, and it may react faster to changes in the subscribed channels. The strategy with worse effectiveness is *Matchmaker*, clearly outperformed by all the other strategies, with values ranging between 83% and 90%. This is because the dynamic reconfiguration of channels takes longer than in other strategies, since it involves not only the middle agent, but also the affected publisher and subscriber agents.

6.2. Precision

While the *Effectiveness* metric above measures to which extent agents are provided with all the information generated in the MAS that they considered relevant, it does not provide any measure about the amount of information received by agents and then discarded, because it was of no interest to them. Such unwanted (*SPAM*) messages cause unnecessary traffic in the system and a processing overload to agents. This can be measured by introducing the *Precision* metric, which can be computed for each test application execution, in the following terms:

$$\text{Precision} = \frac{N_{OK}}{N_{OK} + N_{SPAM}} * 100 \tag{19}$$

Thus, the *Precision* metric is a percentage value where a value of 100% indicates that no message received by any agent was unwanted (i.e., there were no *SPAM* messages)

Table 3 shows the *Precision* average and standard deviation values obtained from the different experiments. As expected, results for *Broadcast* are very poor in terms of precision, specially when the number of publisher agents and channels grow. The other strategies provide much better results (with values between 99% and 100%), since all of them incorporate an actual subscription mechanism. In these strategies, this slight loss of precision in some scenarios is due to the dynamic changes in publisher/subscriber channels.

6.3. Performance

The previous metric can be complemented by another one that quantifies the impact of the unwanted messages to publisher agents, in terms of the resources spent to create and send messages that were of no interest to (some) subscriber agents. This is the purpose of the *Performance* metric, which can be computed for each test application execution, in the following terms:

$$\text{Performance} = \frac{N_{OK}}{N_{SNT}} \tag{20}$$

Thus, a *Performance* value of 1 indicates that all messages sent by subscriber agents where considered relevant by their respective receiver agents, and so, these subscriber agents did not waste any resource sending unwanted messages. A value lower than 1 would express this wasted effort, where not all sent messages were considered relevant when received. A value higher than 1 is also possible, since because of how some strategies work, publisher agents are helped to distribute their messages to several subscriber agents, and so a single *SNT* message may end up being several *OK* messages, received by different (interested) subscriber agents.

Table 4 shows the *Performance* results for the different tests in the case study. In the table, the average results for *Broadcast* are very close

Table 3
Precision results of the experiments with varying numbers of publishers (#Pub), channels per publisher (#Ch/Pub) and subscribers (#Sub).

% precision											
ID	#Pub	#Ch/Pub	#Sub	BCAST		Match		Broker		Trace	
				Avg	Dev	Avg	Dev	Avg	Dev	Avg	Dev
1	200	7	200	0.48	0.54	99.67	0.62	99.97	0.04	99.28	0.43
2	200	7	20	0.63	0.70	99.40	0.89	99.99	0.03	99.48	0.64
3	200	1	200	1.42	2.98	99.91	0.18	99.99	0.04	99.73	0.35
4	200	1	20	5.33	11.69	99.77	0.59	100	0.01	99.90	0.27
5	20	7	200	1.13	0.93	99.94	0.10	99.99	0.04	99.73	0.36
6	20	7	20	4.75	4.09	99.74	0.78	100	0	99.93	0.16
7	20	1	200	1.17	1.16	100	0	99.99	0.07	99.79	0.76
8	20	1	20	9.01	11.28	99.99	0.06	100	0	99.97	0.13

Table 4
Performance results of the experiments with varying numbers of publishers (#Pub), channels per publisher (#Ch/Pub) and subscribers (#Sub).

Performance											
ID	#Pub	#Ch/Pub	#Sub	BCAST		Match		Broker		Trace	
				Avg	Dev	Avg	Dev	Avg	Dev	Avg	Dev
1	200	7	200	0.004	0.005	0.997	0.006	0.911	1.065	2.62	2.54
2	200	7	20	0.006	0.006	0.994	0.008	0.12	0.136	2.01	1.43
3	200	1	200	0.014	0.03	0,999	0.002	2.693	5.819	2.27	3.18
4	200	1	20	0.053	0.117	0.998	0.006	1.026	2.322	2.05	2.03
5	20	7	200	0.011	0.009	0.999	0.001	2.074	1.783	2.85	3.22
6	20	7	20	0.047	0.041	0.997	0.008	0.913	0.798	2.11	1.37
7	20	1	200	0.012	0.012	1	0	2.134	2.106	2.50	4.11
8	20	1	20	0.09	0.113	0.999	0.0006	1.65	2.26	2.11	2.12

to zero, hence demonstrating the huge wasted publication effort of this strategy, where most of the sent messages are discarded by subscriber agents. The average values for *Matchmaker* are very close to 1, showing that this strategy is very successful in delivering the right messages to the right subscriber agents (at the internal cost of publisher agents, which need to maintain the subscription lists), with the exception of the reconfiguration phases after the publisher or subscriber agents change their interests. Values are never higher than 1, since in this strategy, messages are directly sent from publisher to subscriber agents. The average values for *Broker* show very different performances depending on the scene characteristics: the performance is very low with many publishers and channels but few subscribers (Scene No. 2), since all messages from every publisher agent are sent to the middle agent, but many of them may not be relayed to any other agent if there are few active subscriptions. However, the performance is much high when the number of publisher agents is low compared to the number of subscriber agents and of subscriptions (Scene No. 5 for example); in such cases, the average reaches values over 1, because a single message sent to the middle agent may be then relayed to many subscriber agents. Finally, the average performance values of *Trace* are all above 2, since in this strategy, the communication (tracing) layer is in charge of delivering each message (event) produced by a publisher agent to the right subscriber agents, but the event is not transmitted if no agent is subscribed to it, hence not producing any extra traffic in the MAS.

6.4. Performance with middle agent

The results in the previous section are biased for the *Broker* strategy, since they do not take into account the messages retransmitted by the middle agent to the right subscriber agents, nor the potential bottleneck that the broker agent may produce in the MAS. For this reason, this section modifies the *Performance* metric by incorporating the effect of the middle-agent, producing a new metric called *Performance MA*, which can be defined as follows:

$$\text{Performance MA} = \frac{N_{OK}}{N_{SNT} + N_{SNT\_MA}} \tag{21}$$

where N_{SNT_MA} is the number of communication messages sent by the middle agent. This metric introduces a ratio similar to the previous one, where the number of messages correctly received by the subscriber agents are compared against all the communication messages transmitted in the system by both the publisher agents and the middle agent, if any.

The results of this new metric can be seen in Table 5. If compared with the values of the *Performance* metric in Table 4, all strategies except *Broker* produce the same results, as expected. In the *Broker* strategy, the average values are now much worse than in the previous metric, which places this strategy between *Broadcast* and *Matchmaker*. As this new ratio also considers all the communication messages sent in the system, the values of *Performance MA* are considered a much more accurate measure of the performance than the previous metric.

6.5. System overhead

The previous sections have shown significant differences in the performance metrics for the four strategies. Normally, better performance results are the result of a more sophisticated strategy, but this usually comes at a cost. The aim of this section is to quantify this cost, which is related to the way the strategy works. In particular, any of the communication strategies requires some protocol that allow agents to contact each other and agree on the channels about which messages will be exchanged. As explained before, this protocol is supported by the strategy's *system agent*, which in some cases also plays the middle-agent role. In this context, all the messages that agents exchange with the system agent, or *system messages*, produce extra traffic in the MAS that can be considered as system overhead.

However, the measure of the system overhead by using the system messages is not straightforward, since it depends on many factors. The mere amount of system messages on any given test does not express the degree of overhead traffic in that test. The same happens with the ratio between the system and the non-system (communication) messages, since this ratio is affected by the duration of the test (in particular, by the time between channel reconfigurations). Thus, the metric now introduced proposes to compare the system overhead of

Table 5
Performance results including the middle agent (MA) of the experiments with varying numbers of publishers (#Pub), channels per publisher (#Ch/Pub) and subscribers (#Sub).

Performance MA											
ID	#Pub	#Ch/Pub	#Sub	BCAST		Match		Broker		Trace	
				Avg	Dev	Avg	Dev	Avg	Dev	Avg	Dev
1	200	7	200	0.005	0.005	0.997	0.006	0.384	0.204	2.62	2.54
2	200	7	20	0.006	0.007	0.99	0.009	0.096	0.096	2.01	1.43
3	200	1	200	0.014	0.030	0.999	0.002	0.577	0.129	2.27	3.18
4	200	1	20	0.054	0.117	0.998	0.006	0.326	0.23	2.05	2.03
5	20	7	200	0.011	0.009	0.999	0.001	0.604	0.13	2.85	3.22
6	20	7	20	0.047	0.041	0.997	0.008	0.393	0.216	2.11	1.37
7	20	1	200	0.012	0.012	1	0	0.594	0.143	2.50	4.11
8	20	1	20	0.09	0.112	0.999	0.001	0.546	0.117	2.11	2.12

Table 6
System messages ratios of the experiments with varying numbers of publishers (#Pub), channels per publisher (#Ch/Pub) and subscribers (#Sub).

System messages ratios											
ID	#Pub	#Ch/Pub	#Sub	BCAST		Match		Broker		Trace	
				Avg	Dev	Avg	Dev	Avg	Dev	Avg	Dev
1	200	7	200	1	0	28.68	21.07	14.16	4.19	16.76	4.77
2	200	7	20	1	0	19.81	4.21	16.63	0.76	19.99	0,89
3	200	1	200	1	0	11.37	9.86	7.06	4.40	7.62	4.49
4	200	1	20	1	0	6.78	6.29	3.30	0.77	3.90	0.87
5	20	7	200	1	0	16.20	8.90	12.18	8.02	12.92	8.15
6	20	7	20	1	0	25.88	14	14.1	4.15	16.72	4.76
7	20	1	200	1	0	10.20	6.74	11	8.03	11.09	8.06
8	20	1	20	1	0	9.95	4.81	7.09	4.42	7.60	4.49

any strategy against the overhead of the *Broadcast* strategy in the *same* test, since *Broadcast* is considered by definition the "best case" value (i.e., the minimum amount of system messages) in any given scenario. The metric, called *System Messages Ratio*, is defined as follows:

$$\text{System Messages Ratio (strategy)} = \frac{N_{SYS}(strategy)}{N_{SYS}(Broadcast)} \quad (22)$$

where $N_{SYS}(strategy)$ is the number of system messages using a specific *strategy* and $N_{SYS}(Broadcast)$ is the number of system messages using *Broadcast*, both in the same test. The ratio therefore quantifies the *extra* overhead which is produced in the MAS when using a particular strategy compared with the overhead of *Broadcast*.

Table 6 shows the average and standard deviation values of this ratio for all the experiments. The results show that*Matchmaker* is the strategy with more overhead traffic, and it is specially sensitive to scenes where a higher heterogeneity of topics (see scenes No. 1 and No. 6). These variations in the system overhead for the same strategy on different scenes is also produced in *Broker* and *Trace*, but in a lesser degree.

7. Conclusions

Along this paper different contributions have been proposed. First, the paper has defined the Information Flow Problem (IFP) in a MAS as the problem of how to exchange information in the most efficient and effective way within the MAS. Second, this IFP has been formalized by proposing a model which introduces a set of definitions and properties that can be used to describe any MAS from the perspective of the IFP, allowing the system designer to analyze the expected flow of information among the agents in the MAS. The formalization has not considered if agents are cooperative or selfish about the information flow, since this is orthogonal to the approach. In fact, the framework can be used as a way to identify deviations about the expected information flow that could be then used to track down anomalous behaviors of some agents in the system.

Third, the proposed formalization has been instantiated to describe different MAS scenarios, in order to demonstrate its descriptive capabilities. Specifically, the instantiation has included eight well-known MAS scenes which globally compose a representative set of most real MAS existing today. These descriptions have proven that describing a MAS according to the concepts and properties of the proposed model facilitates the analysis and the design of a MAS from the perspective of the information flow.

Fourth, a case study including these eight scenes has been designed and built, in order to compare four different, well-known communication strategies. To do this, an evaluation framework has been implemented. The evaluation framework first generates real, synthetic multi-agent applications conforming to some predefined parameters which match the properties of the formalization model, and then executes such applications on a real MAS called Magentix2. The framework also traces and collects the relevant events about information exchange among agents during each single execution, and produces a complete set of statistical outcomes by combining the results of all executions. Based on these outcomes, the paper has also proposed a set of metrics in order to quantify several performance dimensions of the four communication strategies.

The behaviors of the communication strategies in the different scenes, each one with its own distinctive information flow, have been analyzed by using the results from the performance metrics. In general, the evaluation framework and the performance metrics have been proven as useful tools to analyze the behavior of communication strategies in MAS from the perspective of the information flow problem.

Finally, the proposed contributions can also be considered a step towards providing tools for the MAS design phase in order to determine to which extent some particular factors in a MAS or some communication strategies affect the behavior of the system.

Acknowledgments

This work was partially supported by MINECO/FEDER TIN2015-65515-C4-1-R and TIN2014-55206-R of the Spanish government.

References

Ahn, Y.Y., Han, S., Kwak, H., Moon, S., Jeong, H., 2007. Analysis of topological characteristics of huge online social networking services. In: Proc. of the 16th Int. Conference on World Wide Web. ACM, pp. 835–844.

Aiello, M., Busetta, P., Donà, A., Serafini, L., 2002. Ontological overhearing. In: ATAL '01: Revised Papers from the 8th International Workshop on Intelligent Agents VIII. Springer-Verlag, London, UK, pp. 175–189.

Alberola, J.M., Julian, V., Garcia-Fornes, A., 2011. Cost-aware reorganization service for multiagent systems. In: Advanced Agent Technology. Springer, pp. 442–456.

Althnian, A., Agah, A., 2015. Evolutionary learning of goal-oriented communication strategies in multi-agent systems. J. Autom. Mobile Robot. Intell. Syst. 9 (3), 52–64.

Bellifemine, F.L., Caire, G., Greenwood, D., 2007. Developing Multi-agent Systems with JADE, vol. 7. John Wiley & Sons, pp. 1–285.

Borondo, J., Morales, A., Losada, J.C., Benito, R.M., 2012. Characterizing and modeling an electoral campaign in the context of Twitter: 2011 Spanish Presidential election as a case study. Chaos 22 (2 023138), 1–7.

Búrdalo, L., Terrasa, A., Julián, V., García-Fornes, A., 2011. TRAMMAS: A tracing model for multiagent systems. Eng. Appl. Artif. Intell. 24 (7), 1110–1119.

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 Multiagent Systems. ACM, New York, NY, USA, pp. 1280–1287.

Busetta, P., Serafini, L., Singh, D., Zini, F., 2001. In: Batini, C., Giunchiglia, F., Giorgini, P., Mecella, M. (Eds.), Extending Multi-agent Cooperation by Overhearing. pp. 40–52.

Carrera, Á., Iglesias, C.A., Garijo, M., 2014. Beast methodology: An agile testing methodology for multi-agent systems based on behaviour driven development. Inf. Syst. Front. 16 (2), 169–182.

Chakraborty, D., Sen, S., 2007. Computing effective communication policies in multiagent systems. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, p. 36.

Collis, J., Ndumu, D., Nwana, H., Lee, L., 1998. The ZEUS agent building tool-kit. BT Technol. J. 16 (3), 60–68.

Corkill, D., 1991. Blackboard systems. AI Expert 6 (9), 40–47.

Criado, N., Argente, E., Noriega, P., Botti, V., 2013. MaNEA: A distributed architecture for enforcing norms in open MAS. Eng. Appl. Artif. Intell. 26 (1), 76–95.

Dignum, F., Vreeswijk, G.A., 2004. Towards a test bed for multi-party dialogues. In: Dignum, F. (Ed.), Workshop on Agent Communication Languages. In: LNCS, vol. 2922, Springer Berlin / Heidelberg, pp. 212–230.

Fogués, R.L., Alberola, J.M., Such, J.M., Espinosa, A., Garcia-Fornes, A., 2010. Towards dynamic agent interaction support in open multiagent systems. In: Proc. of CCIA'2010, vol. 220. pp. 89–98.

Goldman, C.V., Zilberstein, S., 2003. Optimizing information exchange in cooperative multi-agent systems. In: Proc. of the 2nd International Conference on Autonomous Agents and Multiagent Systems. ACM, pp. 137–144.

González-Pardo, A., Varona, P., Camacho, D., Ortiz, F.d.B.R., 2010. Optimal message interchange in a self-organizing multi-agent system. In: Intelligent Distributed Computing IV. Springer, pp. 131–141.

Gruver, W., 2004. Technologies and applications of distributed intelligent systems. IEEE MTTChapter Presentation, Waterloo, Canada, pp. 1910–1917.

Guimerà, R., Llorente, A., Moro, E., Sales-Pardo, M., 2012. Predicting human preferences using the block structure of complex social networks. PLoS One 7 (9 e44620), 1–7.

Gutnik, G., Kaminka, G.A., 2006. Representing conversations for scalable overhearing. J. Artificial Intelligence Res. 349–387.

Helsinger, A., Thome, M., Wright, T., 2004. Cougaar: a scalable, distributed multi-agent architecture. In: Systems, Man and Cybernetics, 2004 IEEE International Conference on, vol. 2. IEEE. pp. 1910–1917.

Houhamdi, Z., 2011. Multi-agent system testing: A survey. Int. J. Adv. Comput. 2 (6), 135–141.

Jennings, N.R., 1999. Agent-oriented software engineering. In: European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Springer, pp. 1–7.

Jolly, R., Wakeland, W., 2009. Using agent based simulation and game theory analysis to study knowledge flow in organizations: The KMscape. Int. J. Knowl. Manag. (IJKM) 5 (1), 17–28.

Kaminka, G.A., Pynadath, D.V., Tambe, M., 2002. Monitoring teams by overhearing: A multi-agent plan-recognition approach. J. Artificial Intelligence Res. 17 (1), 83–135.

Legras, F., Tessier, C., 2003. LOTTO: group formation by overhearing in large teams. In: Proc. of the 2nd International Conference on Autonomous Agents and Multiagent Systems. ACM, pp. 425–432.

Legras Onera, F., 2002. Using overhearing for local group formation. AAMAS '02 Workshop 7 "Teamwork and Coalition Formation". pp. 8–15.

Leskovec, J., Adamic, L.A., Huberman, B.A., 2007. The dynamics of viral marketing. ACM Trans. Web 1 (1), 1–39.

Maity, S.K., Mukherjee, A., Tria, F., Loreto, V., 2013. Emergence of fast agreement in an overhearing population: The case of the naming game. Europhys. Lett. 101 (6), 68004.

Malouf, R., 1995. Towards an analysis of multi-party discourse.

Nguyen, C.D., Perini, A., Tonella, P., 2009. Goal-oriented testing for MASs. Int. J. Agent-Oriented Softw. Eng. 4 (1), 79–109.

Nissen, M.E., Levitt, R.E., 2004. Agent-based modeling of knowledge flows: illustration from the domain of information systems design. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004. IEEE, pp. 1–8.

Romero, D.M., Galuba, W., Asur, S., Huberman, B.A., 2011. Influence and passivity in social media. In: Proceedings of the 20th International Conference Companion on World Wide Web. ACM, pp. 113–114.

Rossi, S., Busetta, P., 2004. Towards monitoring of group interactions and social roles via overhearing. In: Proceedings of CIA-04, vol. 3191. pp. 47–61.

Serrano, E., Muñoz, A., Botia, J., 2012. An approach to debug interactions in multi-agent system software tests. Inform. Sci. 205, 38–57.

Sycara, K., Lu, J., Klusch, M., Widoff, S., 1999. Matchmaking among heterogeneous agents on the internet. In: Proceedings of the 1999 AAAI Spring Symposium on Intelligent Agents in Cyberspace. pp. 22–24.

Thangarajah, J., Jayatilleke, G., Padgham, L., 2011. Scenarios for system requirements traceability and testing. In: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, pp. 285–292.

Tummolini, L., Castelfranchi, C., Ricci, A., Viroli, M., Omicini, A., 2005. "Exhibitionists" and "Voyeurs" do it better: A shared environment for flexible coordination with tacit messages. In: Proceedings of the First International Conference on Environments for Multi-Agent Systems, E4MAS'04. Springer-Verlag, Berlin, Heidelberg, pp. 215–231. http://dx.doi.org/10.1007/978-3-540-32259-7_11.

del Val, E., Rebollo, M., Botti, V., 2015. Does the type of event influence how user interactions evolve on Twitter? PLOS One 10 (5), 1–32.

Wei, C., Hindriks, K.V., Jonker, C.M., 2014. The role of communication in coordination protocols for cooperative robot teams. In: ICAART (2). pp. 28–39.

Wong, H.C., Sycara, K., 2000. A taxonomy of middle-agents for the internet. In: Proc. of the 15th Int. Joint Conf. on Artificial Intelligence. pp. 578–583.

Zhuge, H., 2002. A knowledge flow model for peer-to-peer team knowledge sharing and management. Expert Syst. Appl. 23 (1), 23–30.

Zhuge, H., 2006. Knowledge flow network planning and simulation. Decis. Support Syst. 42 (2), 571–592.