The Knowledge Engineering Review, Vol. 36, e3, 1 of 24. © The Author(s), 2021. Published by Cambridge University Press
doi:10.1017/S0269888920000429

A generic model for representing openness in multi-agent systems

SONDES HATTAB and WIDED LEJOUAD CHAARI

National School of Computer Sciences, COSMOS Laboratory, University of Manouba, Manouba, Tunisia
e-mails: sondes.hattab@ensi-uma.tn, wided.chaari@ensi-uma.tn

Abstract

Openness is a challenging property that may characterize multi-agent systems (MAS). It refers to their ability to deal with entities leaving and joining agent society over time. This property makes the MAS behaviour complex and difficult to study and analyze, hence the need for a representative model allowing its understanding. In this context, many models were defined in the literature and we propose to classify them into three categories: structural models, functional models and interactional models. The existing models were proposed either for representing structural openness or for modelling functional or interactional ones independently. But, none of them was oriented to represent MAS openness in a global way while considering its three aspects at once. Besides, each one was defined in order to realize a specific objective and in a particular domain of application. In this paper, we propose an evolving KAGR graph. The latter provides a common understanding of openness and unifies its structural, functional and interactional aspects in a generic way. Our model is finally tested and validated on a multi-agent rescue simulator.

1 Introduction

We are living in a highly dynamic world where reciprocal interactions and data exchanges are continuously taking place between the different components of the environment. This reciprocity is referred to in the literature by the term openness. Software systems that constitute today a part of our daily lives, from business to entertainment, from shopping to education and communication, have to deal with such property in order to meet our needs in a realistic manner. They must be made to allow for extensions, shortenings as well as to interact with other software components, either to provide services and data or to acquire them. Consequently, it is nowadays almost out of the question to resort to closed systems that are based on fixed architectures, a prior definition of the environment and that allow nothing to enter or escape their boundaries. Thus, openness is an important property that characterizes current software systems: it calls for new engineering solutions and it is supported by emerging technologies such as wireless sensor networks, Internet of Things (IoT) and smart and wearable devices used in highly topical issues like energy management, health care, traffic systems, industrial applications as well as social networks, e-learning, e-commerce, e-tourism and so much other challenging applications.

In the past few decades, with the latest advances in both intelligent and distributed computing, the researchers interest has been focused on a particular type of software systems called multi-agent systems (MAS). The latter are considered as more than simple computer programs including a set of autonomous and interacting agents. They are rather seen as complex systems made to solve problems that are beyond the abilities of any individual member. Openness in such systems is as significant as in classic ones, even more since they are more suitable for simulating and solving real life issues. It is defined in the literature as the ability of the system to deal with entities leaving and joining an agent society over

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

time. It is moreover seen as a challenging characteristic of realistic MAS which causes transformations and dynamic changes at the level of the system's topology, internal functionalities and communicative behaviour between the corresponding agents.

The changing nature of such property makes the MAS behaviour complex and difficult to study and analyze, whence the need for a simplifying method allowing its understanding. The use of a representative and explicit model seems to be appropriate, even obvious. This is due to the ability of such mean to allow not only studying and assessing the system performances but also getting insight into MAS design, predicting future behaviour and detecting anomalies and agent failure. In this context, many researchers have been interested in inventing models to represent open MAS. Some of the existing ones are devoted to represent and study open MAS from an external point of view while considering their changing structure in time. Others are defined and used in order to simplify and analyze the evolution of their internal actions and functionalities. Some others put the spotlight on interoperability by representing the interactional behaviour between agents. Based on such observation, we propose to classify them into three categories: structural models, functional models and interactional models. In spite of their diversity, no one of them investigates structural, functional and interactional characteristics at once or is applicable to any open MAS and for any objective.

In the present paper, we aim to achieve a twofold objective: the first is to exhibit our classification proposal of MAS openness models and the second is to define a unifying model for representing openness in MAS. The latter must give a generic and common understanding of such property. In order to validate our theoretical proposal, we will test it on a multi-agent rescue application.

The paper is organized as follows: Section 2 explores briefly the concept of openness in MAS. Section 3 deals with a state of the art on modelling MAS openness while detailing our classification proposal. In Section 4, we select the more appropriate model for representing our desired property. The results validating our proposal are described in Section 5. Finally, Section 6 concludes this paper and presents the future research plans.

2 MAS openness

The concept of openness in MAS is a challenging property that has aroused the curiosity of numerous researchers. It has been defined at several points in the literature. In fact, openness was primarily defined from a managerial point of view. For example, in Sehory (2001), the openness of an MAS is defined as its capacity to introduce new agents in addition to the existing ones. Besides, Calmet *et al*. (2003) noted that an open society of agents is made to allow for extensions. In other words, it is a society that is open to new agents with no definite or irrelevant goals. In Valckenaers *et al*. (2007), openness in MAS is considered as the systems ability to deal with entities not only entering but also leaving the system.

By taking in consideration a functional point of view, open MAS are defined as systems dealing with three possible modifications: addition, removal and evolution of entities (Vercouter 2000; Boissier *et al*., 2004). Evolution is indeed seen as the gain of new capabilities and internal attributes or the loss of old ones. Similarly, Jumadinova *et al*. (2014) and Chen *et al*. (2015) distinguish between two types of openness: agent openness and task openness. The first one depends on the availability of the needed agents to perform tasks in the system and the second one refers to the dynamic changes in the tasks that appear over time. In the same context, Weyns *et al*. (2010) define MAS openness as the ability of the system to adapt itself to uncertainty in the environment and to handle dynamic perturbations like the coming of new tasks and resources.

Another point of view is taken into account: the interoperability between agents. In fact, as said in Petsch (2002), open MAS are combining sociological and software engineering aspects. Thus, they are seen as systems that are developed under consideration of standards of interoperability and with respect to social aspects. This definition is adopted and developed in Weyns *et al*. (2015). In the latter work, open MAS are seen as systems where agents interact together in order to achieve their objective. To this end, they call for common ontologies, communication protocols and suitable coordination infrastructure.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

From the different definitions of openness exhibited above, we can extract two observations. The first is that MAS openness has three aspects:

(i) **Addition and removal of agents**: such aspect considers agents as physical entities or black boxes and takes into account only unpredictable changes in the MAS topology. We propose to call this aspect structural openness;
(ii) **Addition, removal and modification of tasks and agent roles**: openness here is based on the internal evolution of agents and considers the modification of their content, goals, attributes and functionalities. We call this aspect functional openness;
(iii) **Interoperability and communicative behaviour between agents**: this aspect takes into account open interactions between agents and highlights the common followed rules, norms and protocols. We call this aspect interactional openness;

The second observation we make is that studying MAS openness is not an easy task: its uncertainty and dynamicity at the level of structure and functionalities make its analysis, assessment and understanding complex and difficult to handle. Thus, there is a pressing need for a technique permitting to effectively manage MAS complexity and to facilitate the comprehension of its global behaviour. Modelling is, indeed, a method which may be used to handle this issue. It has found broad applicability in several different areas and in particular in open MAS field.

Based on such observations, we will give in the next section an overview of the existing models while proposing a three-category classification: structural models, functional models and interactional models.

3 Modelling MAS openness: state of the art

Modelling open MAS has interested many researchers in the MAS community: several models have been proposed and applied to different applications and situations. As mentioned above, they can be classified into three categories: structural models, functional models and interactional models. They will be discussed in details in the following subsections.

3.1 Structural models

We define structural models as the set of methods and tools describing how MAS openness is seen from an external point of view, without taking in consideration the system internal properties and data evolution. Actually, each existing model in the literature is representing openness either explicitly by downright describing main structural openness operations: addition, removal and evolution of agents, or implicitly by tackling some related aspects such as structural controllability.

Depending on the objective for which the model was proposed, we enumerate and describe below each found model apart.

3.1.1 Openness management

The existing approaches in this context were dedicated to dealing with addition, removal and evolution of agents in MAS. We mention mainly: the centralized, the distributed and the hybrid model.

(i) **The Centralized Model** (Decker *et al*., 1997): it is based in a specific agent called broker. When an agent is integrated in the system, it must provide its characteristics to the broker, which updates itself according to the capabilities and requirements of the new agent. Any agent leaving the system has only to signal his removal to the broker.
(ii) **The Distributed Model** (Sehaba, 2002): in this model, agents communicate by broadcasting messages. The addition of a new agent requires to warn all the existing agents of his presence and to ask for their knowledge. When modifying an agent, this latter must send a message to all the agents in the system and they will update their representations. In a similar way, the removal of an agent requires him to signal his departure to all the elements.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

(iii) **The Hybrid Model** (Sehaba, 2002): the MAS is decomposed into groups of connections. In order to integrate the system, an agent must send its characteristics to the broker which assigns it to the appropriate group. Consequently, the agent will present itself to the elements of the group in question and receives their representations. As for the removal of an agent, this latter has just to report his departure to its acquaintances. Similarly, an agent having evolved must signal the disappearance of some of its characteristics and have to introduce its new functionalities to its acquaintances.

3.1.2 Openness design

Openness here is seen as an engineering challenge. The design of open MAS was considered as a topic of a major concern and researchers had started to propose architectural models and suitable infrastructures and mechanisms to support MAS structural openness. We mention as example here the Peer-to-Peer model which is the basis of some software frameworks mainly Jade as well as some reorganizational models.

(i) **Peer to Peer Model**: this model is suitable for designing open MAS thanks to its ability to scale up to thousands or even millions of nodes. The addition or the removal of an agent from MAS, whose conception is based on such model, causes a change to the whole system's structure. The use of this trend implies handling some openness aspects such as intrusion and exception detection, workflow management and security issues (Aldeeb *et al*., 2008; Platon *et al*., 2008; Bijani, 2012).
(ii) **Reorganizational Model**: they are based on what is called OCMAS (Organization-Centered MAS), Massonet *et al*. (2002). This approach gives a big attention to the MAS structure. It defines the logical relationship between agents and their organizations regardless of the agents behaviour and activities. We mention as example the AGR model (Agent/Group/Role), also called AALAADIN, and the AGRE model. We also mention GoOrg model (Amaral & Hübner 2019). It is a structural model defined in order to generate an automated design of the open MAS. It uses as input a goals tree, the necessary skills to achieve the goal and the predicted workload and throughput. It generates as output an organizational chart.

3.1.3 Openness simulation

Some models are proposed in the literature in order to represent and simulate structural openness in MAS. We mention, in this context, the algebraic models and more particularly Hendrick's work (Hendrickx & Martin 2016). In this work, integers are used to label the agents: the set of agents present in the system at time t is denoted by $N(t)$ and its cardinality is denoted by $n(t)$. Each agent i holds a value $x_i(t) \in IR$. At each time t, one of the following three events may occur causing a change in the MAS structure:

(i) **Departure of an agent**: it is mathematically expressed as follows: one uniformly randomly selected agent $i \in N(t)$ leaves the system so that $N(t+1) = N(t)/i$ and $n(t+1) = n(t) - 1$. This event may only occur when $n(t) > 0$;
(ii) **Arrival of an agent**: one new agent $i \in N(s)\ \forall\ s < t$, joins the system so that $N(t+1) = N(t) \cup i$ and $n(t+1) = n(t) + 1$. The initial value $x_i(t+1) \in IR$ of the arriving agent is drawn independently from a constant distribution D with mean 0 and variance σ^2;
(iii) **Gossip between two agents**: two agents $i, j \in N(t)$ are uniformly randomly and independently selected among the $n(t)$ agents and they update their values x_i and x_j by performing a pairwise average: $x_i(t+1) = x_j(t+1) = (x_i(t) + x_j(t))/2$.

3.1.4 Openness evaluation

Recently, the topic of performance evaluation in MAS has interested some researchers. In particular, structural openness (Hattab *et al*., 2017) and structural controllability (Liu *et al*., 2013; Guan *et al*., 2017; Zegzhda *et al*., 2017) were assessed using graph theory.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

(i) **Evaluation of structural openness**: in Hattab *et al*. (2017), evolving graphs were selected for assessing structural openness in MAS thanks to their ability to authorize any combination of addition and removal of entities. By applying a number of graph theory properties, some measures were defined in order to evaluate the structural changes at the level of MAS topology. We mention as example: alpha index, degree of structural extensibility, structural impact, transitivity, etc.

Actually, the work described above has a specific goal: the evaluation of structural openness in a generic way. The proposed approach was validated on a multi-agent road traffic application and could be applied on other domains since its generic nature. The difference between that work and the present one is that the first is generic only in the context of structural openness. However, the second aims to define a model which is generic in the sense that it could take in consideration any type of openness and any application domain.

(ii) **Evaluation of structural controllability**: to decide if an MAS is structurally controllable or not, dynamic graphs were used. For example, in Angluin *et al*. (2008), it was shown that an MAS is structurally controllable if and only if the union graph is connected. In Zegzhda *et al*. (2017), the goal was to compute controllability boundaries. For this purpose, a regular graph in the form of tree with virtual nodes was proposed and some methods of calculating the number of vertices were applied. Similarly, in Guan *et al*. (2017), the controllability of MAS was quantitatively studied. The latter work supposed that the interconnection between agents is directed, and it employed the notions of distance partition and almost equitable partition for the calculation;

3.2 Functional models

We define functional models as the set of methods, tools and approaches describing MAS openness from an internal point of view. Such models aim to show the inner behaviour of agents and how it changes over time due to the unpredictable addition, removal and evolution of some properties like agent knowledge, roles and goals. They stress the internal scalability in open MAS and highlight the resulted autonomic and adaptive behaviour in such systems. Similarly to the structural models, functional ones have been proposed in order to achieve some specific objectives namely: the design of open MAS, the analysis of open MAS internal properties, decision making as well as anomalies detection and the verification of some characteristics. We summarize below some existing functional models in the context of open MAS.

3.2.1 The design of open MAS

The design of open MAS has been primarily seen as a major preoccupation since the AOSE methodology works (Massonet *et al*., 2002). The latter were proposed in order to describe an MAS by covering functional and non-functional goal analysis. We mention as example here AgentUML, Gaia and MAS-CommonKads. Actually, these modelling languages are based on four main concepts: Organization, Role, Goal and Task. The drawback of such methodologies is their static and deterministic nature which contradicts with openness notion. Indeed, modifications at the level of the previously mentioned concepts were not taken into account. Moreover, the uncertainty in the deployment context, tasks and requirements was not considered. That's why, looking for new methods to design open MAS was in urgent need. We mention as example: the role-based model, the service-oriented approach, the logic-based model and the normative models.

(i) **The Role-Based Model** (Xu *et al*., 2007): it consists in a three-layer model namely, agent society, role space and role organization. In this model, openness appears at three levels and it is defined in this work as the dynamic additions and removals of agent roles during runtime;

(ii) **The Service-Oriented Model**: such model was used to deploy large-scale open MAS like THOMAS (Carrascosa *et al*., 2009; Argente *et al*., 2011). It was specifically addressed for the design of open organizations and was proposed in order to allow agents to handle the problem of discovery and composition in dynamic and changing open environments. It includes two categories of services: openness management services and structural services. Another work was recently done in this context. It consists in proposing an open based-services architecture in order to engineer Vehicle-to-Grid

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

(V2G) and Grid-to-Vehicle (G2V) energy transfer problem. To this end, some novel design patterns were defined to be used with statecharts in many real world situations (Spanoudakis *et al*., 2019).

(iii) **The Logic-Based Model** (Artikis *et al*., 2016): this work takes in consideration the unpredictability of open environment in MAS. It supposes that the behaviour of an agent in such systems cannot be predictable or controllable. In fact, an agent may disobey to the MAS specification in order to realize its individual goal, or it can fail to conform to the MAS specification because of reasons beyond its control. That's why, architectures with fixed specifications are seen as not suitable for the design of open MAS. As a solution, a logic-based model was proposed. The latter is characterized by its ability to modify, at runtime, the fixed specifications at design time. It consists of a well-defined procedures for a modification of the rules of the game, as well as decision making over and enactment of proposed modifications.

(iv) **Normative Models and Electronic Institutions**: the changing nature of open MAS, as well as the strong collaboration and coordination in such systems, requires an appropriate architecture not only to deal with additions, removal and evolution of agents but also to enhance performance and facilitate interactions between them. In recent years, normative modelling appears as an important method to attain these goals. Norms, which are considered as a set of defined conventions to articulate or to restrain interactions (Viana *et al*., 2018) or as rules and standards commonly understood by a group of agents in order to produce a desired behaviour (Mahmoud *et al*., 2014), are nowadays a powerful formalism in designing open MAS. The importance of such model is highlighted in Luck *et al*. (2003) by declaring: *'Human societies have successfully coped with similar issues by creating institutions that establish norms for group dynamics in open systems. The next challenge for agent-based computing is to develop appropriate representations of analogous computational concepts to the norms, legislation, authorities, enforcement, etc., that can underpin the development and deployment of dynamic electronic institutions'*. In Derakhshan *et al*. (2013), norms are considered as a practical method for the dynamic assignment of rights, responsibilities and sanctions to agents in open MAS. In this work, an architecture using the normative method was implemented. The latter facilitates the management of agents as follows:

- When an agent enters or exit the system, a new role will be assigned to the agent and as a consequence dynamic assignments of rights and responsibilities will occur.
- Dynamic assignment of rights and responsibilities may take place when an action or/and when an important time is reached.

Besides, normative solutions were used in Mahmoud *et al*. (2014) in order to propose frameworks and architectures for designing open MAS. We mention mainly: BOID Normative Architecture (Belief, Obligation, Intention and Desire), Programming BOID-Plan Agent, BIO Normative Architecture, Normative KGP Agents and OP-RND Normative Framework.

Added to that, the issue of synthesising normative systems has been recently studied in Huang *et al*. (2016) and after that a framework was designed and produced for this purpose in Morales *et al*. (2017). The latter carries out an evolutionary game theoretic process which allows to the agents to adopt the norms that are fittest to coordinate them.

In the same context and as a consequence to the use of normative models in designing open MAS, electronic institutions have been appeared as an important software element in engineering open environments and regulating agents behaviour. They are defined in Jonge *et al*. (2014) as components providing the rules of the game in agent societies by fixing what agents are permitted and forbidden to do and under what circumstances. The latter work surveys existing infrastructures to engineer open environments using electronic institutions such as EIDE and AMELI. In de Brito *et al*., (2019), an architecture of an institutional platform was proposed, such platform was conceived in order to achieve the normative regulation of the system. In a similar way, an institutional model for representing open MAS was proposed in Fornaro *et al*. (2008).

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

3.2.2 Analysis of open MAS internal properties

Some researchers' efforts were dedicated to analyzing internal properties of open MAS. Actually, interests have been essentially focused on measuring the stability of open MAS functioning. Thus, controllability was studied and assessed from a functional point of view, and graph theory was used as a tool for achieving such purpose. We mention Rahmani's work (Rahamani *et al*., 2009) in which graph theory was used to study the agreement problem of MAS. MAS controllability was also investigated based on path and cycle graphs in Liu and Ji (2018). For the same purpose, many research works were proposed such as Tian *et al*. (2018) and Lu *et al*. (2018). We mention also the work of Robin Milner that highlights the importance of modelling agents networks in order to understand and optimize their behaviour. It proposes a unified structural theory based on bigraphs to represent MAS (Milner 2009). Besides, since Hendler's works Hendler (2001) and (2007), the concept of semantic web has aroused the curiosity of the multiagent community and we are today face a new vision for a new generation of autonomous systems on web called Hypermedia MAS (Ciortea *et al*., 2019). In this context, a huge number of models was proposed in the literature in order to analyze and interpret agents' behaviour. For example, knowledge graphs were used to this end. They are defined in Chen and Chen (2019) as a set of structural data which are interconnected to describe objective things such as concepts, entities and attributes. The latter work enumerates a set of hypemedia MAS where agents exploit knowledge graph to extract useful information. For example, it is used in Intelligent Search Systems to analyze the needs of users and to provide the answers that best meet them. Such graph is also used by agents in Question and Answer Systems in order to retrieve correct answers to the raised questions by users. Finally, in recommendation systems, agents behaviour is modelled as a knowledge graph. The latter was useful to analyze the potential needs of users and to suggest, as a consequence, a relevant and personalized content to them. We mention, also, Ontologies-Based Models which have been recently used in a huge number of research works in the multiagent field. We cite as example Ma *et al*. (2019), Pico-Valencia *et al*. (2019) and Guo *et al*. (2019).

3.2.3 Decision making

Decision making in the context of open MAS is seen as the ability of the composing agents to predict if other agents left the system or new agents are now present. This is indeed in order to decide on possibly choosing a different line of action. To this end, Markov Decision Process was used as a tool to represent open agent systems. In this context, we mention the Interactive Partially Observable Markov Decision Process (I-POMDP). The latter focuses on studying the impact of agents leaving and reentering in a wildfire scenario by providing a set of equations to predict agent openness, that is to estimate when a neighbouring agent leaves or reenters the system (Chandrasekaran *et al*., 2016).

3.2.4 Anomalies detection

As openness becomes an important concern, open MAS are becoming more vulnerable to intrusions, misuses and attacks. Hence, the detection of such threats seems to be necessary. For such purpose, modelling technique was used in some works. For example, some anomalies detection methods were mentioned in Bijani's review (Bijani 2012). For example, two attack modelling approaches in open MAS were introduced namely: coordination graph and statistical techniques. Coordination graphs, in which nodes are states and arcs represent attacks, allow detecting maximal malicious group of attackers. As for statistical techniques, they help detecting some intrusions at the level of functionalities like probing attacks, ontology attacks and denial of services attacks in open MAS. In the same context, a graph-based approach has been introduced in Kanfert *et al*. (2014). It is proposed in order to detect suspicious behaviour in open MAS at runtime. Such formalism helped applying some metrics such as prestige, actor centrality, reputation Buckets.

3.2.5 Stability verification

With the development and deployment of open MAS in diverse applications, a growing need has emerged to verify and validate agents behaviour. In fact, such task allows estimating not only the stability of the

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

system but also the communicative behaviour correctness. Based on a literature overview, we mention the following existing models:

(i) **Finite State Automata**: in multi-agent self-stabilizing systems, where interacting agents can undergo temporary or permanent failures representing, respectively, the replacement or the departure of an agent, varying compositions were modelled as a graph of finite state machines also called communication graph (Angluin *et al*., 2008). Such automata show how agents change dynamically their internal states while interacting. Their objective is to help in designing algorithms that eventually stabilize on the desired answer if the system compositions stop changing, that is, once the system has become closed.
(ii) **Markov Chain Model**: in order to get information about the level of stability in evolving MAS, a Markov Chain Model was used in De Wilde and Briscoe (2011). The latter provides a better understanding and quantification of MAS stability. It represents an attempt to control such systems.
(iii) **Formal methods**: we mention here kouvaro's work (Kouvaros *et al*., 2019). The latter defined a semantics for open MAS and their corresponding verification problem which are called the open model checking problem. In this context, two classes of systems that admit verification procedures were defined and they aimed to assess specifications on MAS scenarios.

3.3 Interactional models

We define interactional models as the set of methods and protocols highlighting the interoperability and the communicative behaviour between agents in open MAS. These models present rules, norms and protocols ensuring agent interactions in open environment. They are defined in order to realize some objectives as: trust and reputation modelling, agent conformance and compliance verification and knowledge exchange representation. Depending on these objectives, we present in the following sections, the different interactional models that exist in the literature.

3.3.1 Trust and reputation modelling

Because of the decentralization and the high distribution of open MAS, agents may be said to be unreliable and self-interested (Hyunh *et al*., 2006). Hence, they need mechanisms to control their interactions and to choose partners to cooperate with. At this level, trust and reputation mechanisms have arisen as key elements are not only facilitating the interactions among agents but also in the design of MAS. They are defined as expectation of competence and willingness to perform a given task in order to estimate the trustworthiness of each agents partner and to decide whether the partner is sufficiently reliable to interact with. We provide below an overview on trust and reputation models that currently exist in the literature. Indeed, has enumerated and classified an amount of trust and reputation models as follows:

(i) **The Centralized Trust and Reputation Models**: this family of models works very well in centralized environments. It is suitable for representing systems putting together two types of agents: users and sellers. It aimed to calculate the reputation value for each agent. We mention mainly the online reputation models that are appropriate for designing e-commerce sites such as eBay, Amazon and OnSale. The main examples in this context are Sopras and Histos model, Carter and al model as well as Dirichlet reputaion systems.
(ii) **The Agent-Oriented Trust and Reputation Models**: this family of trust and reputation models is suitable for peer-to-peer systems. It considers reputation or trust as subjective or cognitive properties. We can mention mainly Abdul-Rahman and Hailes model (Abdul-Rahman & Hailes 2000) and AFRAS (Carbo *et al*., 2003). The first aims to qualify the relationship between any two agents in terms of trustworthiness. To do so, for each agent, a discrete set of four elements is assigned: Very Trustworthy, Trustworthy, Very Untrustworthy, Untrustworthy. Then, for every instance of the previous set, the number of the corresponding past experiences is kept and the maximum of the tuple values is taken into consideration to decide how much an agent is trustworthy for its partner. The second one is based on a fuzzy set to represent reputation values, and it uses a weighted aggregation

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

in order to update an old one. Castelfranchi & Falcone model (Castelfranchi & Falcone 1998) is based on the beliefs of an agent *i* in the ability and performances of an agent *j* and its dependency on it to obtain a goal *g*. This notion was refined in ForTrust model (Herzig *et al*., 2008) which differentiates between occurrent and dispositional trust. The first refers to the trust on other agents to act here and now, and the second denotes the disposition of the trustee to perform an action in order to achieve a goal. Both of them are represented and defined using predicates. Esfiandiari *et al*., model (Esfandiari *et al*., 2001) aims to evaluate trust in different ways: observation-based-evaluation using bayesian networks and interaction-based-evaluation using protocols to ask agents about some information. To deal with the received witness information, a trust net is built as a directed graph with interval-labelled edges. We mention also the cognitive models having an ontological dimension such as ReGret (Sabater & Sierra 2001) and FIRE (Hyunh *et al*., 2006). They compute as well as trust value and reliability measure using direct experiences, witness information and social structures. Some other agent-oriented models are proposed in order to represent online market places and to provide reputation information to the buyers mainly Padovan *et al*. model (Padovan *et al*., 2002) and Regan and Cohen model (Regan & Cohen 2005). We also mention Repage (Sabater-Mir *et al*., 2006) which is a cognitive model based on a tuple of predicates expressing whether a given target agent is good or bad with respect to a certain context. BDI+Repage model (Pinyol & Sabater-Mir 2013) integrates Repage model with a multi-valued BDI architecture providing a logic-based framework. It is defined as a multi-context system where each context is seen as a logic and a set of formulas written in that logic.

3.3.2 Agent compliance and conformance verification

In open MAS where entities interact in an unpredictable way, we cannot assume that agents will always act so as to conform with conversations policies and to comply to interaction protocols. Thus, conformance and compliance verification seems to be important, and many research works have studied this topic. For example, Baldoni *et al*. (2006) verify the conformance of an agent's conversation policy to a public conversation protocol, which is based on the theory of formal languages. The conformance test was done using a special finite state automaton. In the same context, a tool which is able to observe and check for agent compliance to interactions protocols was defined in Alberti *et al.* (2006). The verified protocols are written in a logic-based formalism called Social Integrity Constraints (ICs), and it can be used to provide semantics to communicative actions and protocols in terms of expectations about agents' behaviour.

In De Pinninck *et al*. (2010), graph theory was used in order to represent the normative behaviour of agents in open MAS. The latter work proposes to model the system as a network where agents interact under defined rules and using a set of enforcement techniques. Such techniques are meant for scenarios in which the agents do not know in advance which agent to interact with. They permit verifying that the agent code will fulfill all the systems norms. The verification, in this work, happens at two levels:

- **At the level of agents**: by verifying the partner agent's abidance to the normative behaviour.
- **At the level of interactions**: by verifying the truthfulness and validity of a sent message.

Added to that, runtime norm verification has shown its significance. As mentioned in Alechina *et al*. (2018), it deals with techniques that allow checking whether an execution of the system satisfies or violates a given correctness property. Such verification is based on different formalisms: extended regular expressions, trace matches by the AspectJ team and we cite here the work of Ancona *et al*. (2017), query-oriented languages like PQL and rule-based approaches.

We mention, also, the work concerning enforcing equilibria in MAS (Perelli 2019). In fact, the equilibrium issue is implicitly related to MAS openness and its verification seems to be important in such systems. It refers to checking whether the Nash equilibria are compatible with a temporal specification that is desired from the designers perspective. In the latter work, verification techniques including model checking are used to this end.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Using the same model, strategic abilities in MAS with private data sharing (Belardinelli, F., Boureanu, I., *et al*., 2019) and those under imperfect information (Belardinelli, F., Lumuscio, A., *et al*., 2019) were verified. We propose to call the models exhibited above AIPs models with reference to the Agent Interaction Protocols.

3.3.3 Knowledge exchange representation

In MAS, the communication between agents is ensured through exchanging messages. Such exchange needs mutual understanding of the message meaning. Agent communication language (ACL) is used to this end. It is considered as the de facto standard of ACL's implementation in MAS. During the last decades and due to the Internet spread, semantic web technologies were exploited in order to facilitate the data exchange. Gan *et al*., (2018) propose a combination of the existing FIPA-ACL with Semantic Web (SW). This approach is proposed to enhance the communication between agents and to ensure semantic interoperability. In Jirkovsky and Kadera (2019), a solution for facilitating data exchange between agents is proposed. It is based on the modelling of system interfaces and messages instead of data model.

3.4 Critical discussing and positionning

Through the state of the art exhibited above, we note that the proposed models in the context of open MAS were not defined from a generic point of view. Indeed, they were proposed either for representing structural openness or for modelling functional or interactional ones. But, none of them was oriented to represent MAS openness in a global way while considering its three aspects at ones. Furthermore, each proposed model in the literature was defined in order to realize a specific goal and in a particular domain of application. This models diversity may be the cause of exhausting research works: researchers have always to study the existing models, to establish a critical discussion and to make a comparative study in order to decide which model is more appropriate for achieving the research goal. This is in fact not the case when a generic model combining all openness aspects and which may be applicable to any multi-agent application and for any objective. The latter may actually facilitate the far researchers work.

In this context, we propose in the present paper to define a generic model for representing MAS openness. The latter must provide a common understanding of openness and have to unify its structural, functional and interactional aspects in a generic way. This is what we will do in the next section.

4 A generic model for representing MAS openness

In this section, we define a generic model for representing openness in MAS, a model that may provide a more general and shared understanding of MAS openness and that might be useful for multiple objectives. The choice of such model is based on the following reasoning.

4.1 Evolving graphs

Open MAS are dynamic systems with strong evolution over time. Its changing nature at the level of structure or/and functionalities requires the adoption of a dynamic model. Indeed, static and deterministic ones are not able to show and highlight the different modifications and systems updates, which make impossible a lot of tasks such as evaluation, prediction, management, analysis and so much others that need changes monitoring over time. Furthermore, whenever a system is defining elements that relate and interact with each other, it seems natural to represent it as a graph where each node stands for one component and each edge symbolizes an interaction between two components. Thus, a dynamic graph seems to be suitable for representing open MAS. It is defined in the literature as follows Zaki *et al*., (2009):

A dynamic graph is a sequence:$=(G_1, G_2, ..., G_n)$ *where each static graph* $G_i := (V_i, E_i)$ *models a set of objects* V_i*, called vertices or nodes and their relationships* $E_i \subseteq V_i \times V_i$ *called edges or arcs. The indices refer to a sequence of time steps :*$=(t_1, t_2, ..., t_n)$*).The graph* G_{i+1} *is obtained from* G_i *by simple modifications: additions and removals of vertices and edges.*

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Table 1 A comparative table of dynamic graphs

Criteria	Graph dynamicity		Knowledge of evolution events		Evolution process	
Graph	Fully dynamic	Partially dynamic	Prior	Progressively revealed	Described in the model	Randomly generated
Complex networks	√		√		√	
Evolving graphs	√			√		√
Re-optimization graphs		√	√			√
Space-time networks		√		√		√

Many dynamic graphs were studied and analyzed in the literature. We mention mainly: Complex Networks, Evolving Graphs, Re-optimization Graphs, Cumulative Graphs and Space-Time Networks. These graphs are classified in Pigné (2009) following three criteria:

(i) **Graph dynamicity**: it refers to the ability of the dynamic graph to undergo modifications in its components. Two types of dynamic graphs exist: the fully dynamic graphs allowing modifications in nodes, arcs and their associated attributes and the partially dynamic graphs allowing changes only at some components.
(ii) **Prior knowledge of evolution events**: dynamic graphs are classified into two categories: those whom evolution events are known in advance and those whom changes are not initially known but progressively revealed in time.
(iii) **Evolution process**: this criterion distinguishes between dynamic graphs depending on how the evolution events are generated. In fact, the process to generate the events may be described in the model or is simply resulting from random applications.

Based on these criteria, a comparative study of the different dynamic graphs was established. We summarize it in the following Table 1. We marked with '√' the graphs satisfying the criterion described in the corresponding column.

Actually, an open MAS is a strongly dynamic system which permit any operation of addition, removal and internal modification of agents and interactions. Its environment is so uncertain that it is impossible to know in advance which are the events responsible for the changes or how are they generated. Therefore, the dynamic graph that we will base on should have the following properties: a full dynamicity, a progressively revealed evolution events and a randomly generated evolution process. So, an evolving graph is more appropriate to model MAS openness. Evolving graphs are studied in several works. We mention mainly Ferreira (2002) and (2003). Its definition is given below:

Let be given a graph G (V, E) and an ordered sequence of its sub-graphs $SG = G_1, G_2, ..., G_T$ *such that* $\bigcup_{(i=1)}^{T} Gi = G$. *Let* $ST = t_0, t_1, t_2, ..., t_T$ *be a sequence of time instants. Then, the system* $G_E = (G, S_G, S_T)$, *where each* G_i *is the sub-graph during* $[t_{i_1}, t_i [$, *is called an evolving graph. The graph G is called underlying graph.*

Figure 1 shows an illustrating example of an evolving graph built from three sub-graphs G1, G2 and G3.

4.2 A generic evolving graph for modelling MAS openness

In this section, we define and describe the evolving graph for modelling MAS openness in a generic manner. In other words, we give the common characteristics that can be after that used for multiple specific contexts and goals. As we previously said, our model has to unify structural and functional aspects at once. So, we propose it to be a two-layer evolving graph. The first is an external layer representing the

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

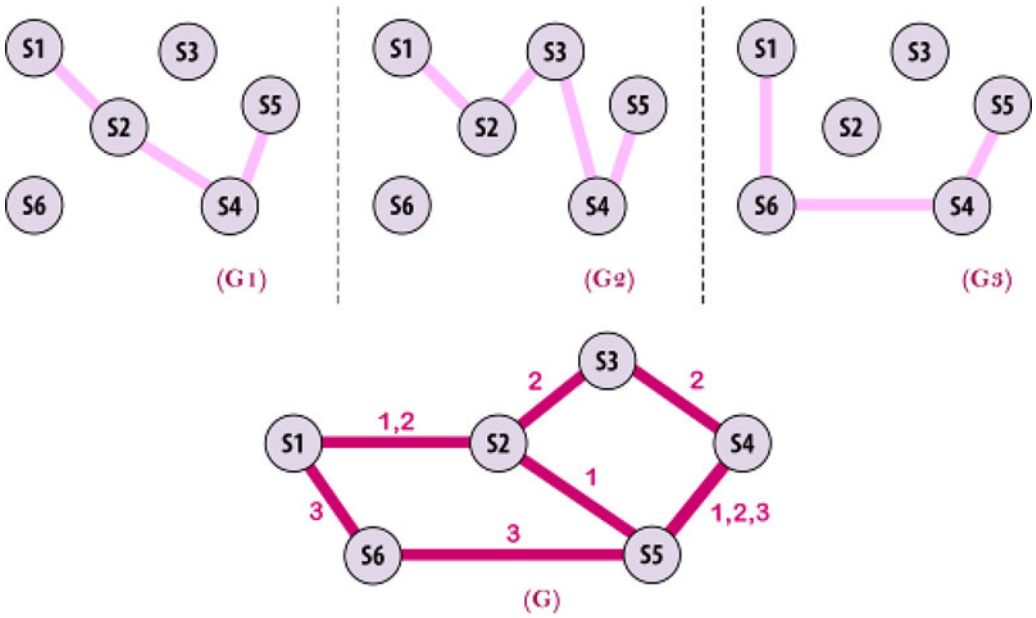


Figure 1 Illustrating example of an evolving graph

open MAS structure that is showing agents and their relationships evolution in time. In the second layer, we put the spotlight on the internal attributes and functionalities. In the following paragraphs, we detail each layer apart.

4.2.1 The first layer: a structural evolving graph

Our evolving graph consists of a set of vertices V representing the agents of the system and a set of arcs E symbolizing the interactions relating the agents over time. Indeed, an interaction may be unidirectional or bidirectional. That is to say, it involves two types of agents: a requester agent, which asks for the interaction, and receiver agents, those undergoing the interaction. Thus, it seems judicious to make the generic graph directed. Besides, we propose to label each component of the graph as follows:

(i) Each node v of the evolving graph symbolizes a corresponding agent ai of the system. Actually, each agent is characterized by an identifier id that distinguishes it from the other agents. Added to that, in an open MAS, an agent can appear and disappear a number of times, which make the system structurally open as previously defined. Thus, we propose to assign for each vertex v of the underlying graph, the set of the structural properties of the corresponding agent ai. We represent such set by:

$$S_v = < id, A_v, R_v, P_v > \tag{1}$$

where

- id is the identifier of the corresponding agent ai.
- A_v is the number of appearances (additions) of the agent ai in the system.
- R_v is the number of disappearances (removals) of the agent ai from the system.
- P_v is the presence vector of the node v. Such vector consists in keeping a historical record of the agent structural evolution in time. It is defined in Pigné (2009) as follows:

  Let $Gi = (V_i, E_i)$ be the subgraph of the underlying graph $G = (V, E)$ at the date t_i. A vertex v is said present at time t_i if $v \in V_i$.

  Thus, the presence vector of a vertex v of the underlying graph G is a vector of integers such that:

$$\begin{cases} Pv(v, Gi) = 1 & \text{if } v \in Vi \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

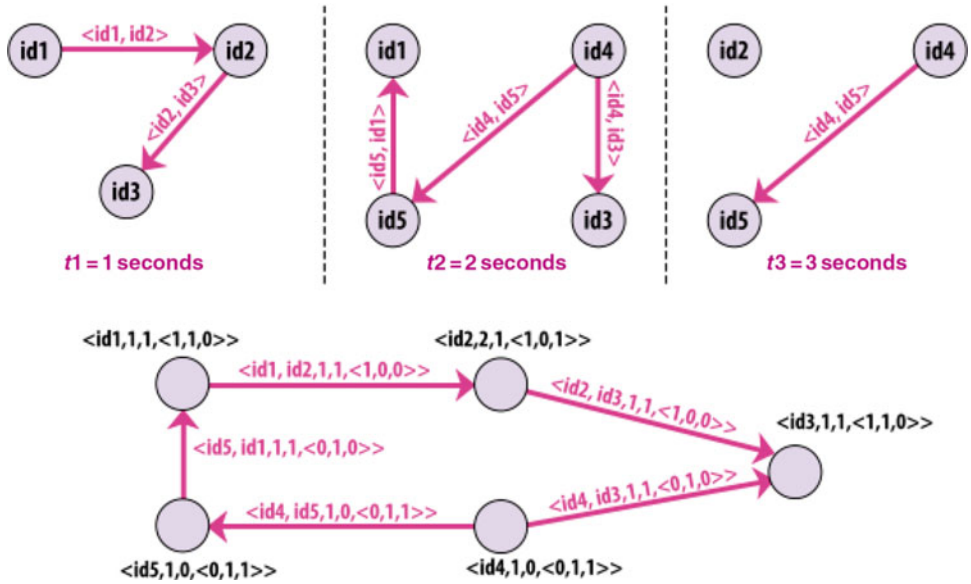


Figure 2 Illustrating example of a random structural evolving graph

(ii) Each arc e of the evolving graph corresponds to an interaction between two agents. Actually, each interaction is characterized by a sender identifier SrcId and a set of receivers identifiers DesId. Added to that, in an open MAS an interaction can appear and disappear a number of times, causing changes in the MAS structure. Thus, we propose to assign for each arc e of the underlying graph, the set of the structural properties of the corresponding interaction. We represent such set by:

$$Se = < SrcId, DesId, A_e, R_e, P_e > \tag{3}$$

where

- SrcId is the sender identifier.
- DesId is the set of the receivers identifiers.
- A_e is the number of appearances (additions) of the corresponding interaction.
- R_e is the number of disappearances (removals) of the corresponding interaction.
- P_e is the presence vector of the arc. Such vector consists in keeping a historical record of the interaction structural evolution in time. It is defined in Pign (2009) as follows:

  Let $G_i = (V_i, E_i)$ be the subgraph of the underlying graph $G = (V, E)$ at the date ti. An arc e is said present at time ti if $e \in Ei$.

  Thus, the presence vector of an arc e of the underlying graph G is a vector of integers such that:

$$\begin{cases} Pe(e, Gi) = 1 & \text{if } e \in Ei \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

  Based on the description above, we give in Figure 2, a random example of a structural evolving graph which represents a MAS regardless of its domain of application.

4.2.2 The second layer: a KAGR graph

In the second layer, each agent a_i has a set of internal properties. They refer to the internal attributes that may be added, removed or/and changing value over time. Actually, the literature gives a conventional definition of an agent. It is defined as a computer system which is situated in a dynamic environment and which is able to communicate with other agents and to use its knowledge and experiences in order to achieve a certain goal. This definition exposes the different internal attributes of an agent namely: Knowledge, Action, Goal and Relation. Indeed, these attributes were defined in Sebai and Chaari (2014)

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

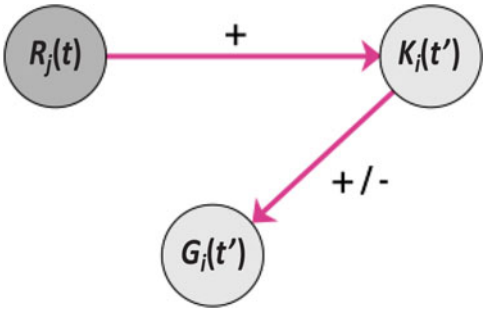


Figure 3 The addition of knowledge to an agent a_j

in order to explain and interpret multi-agent reasoning. In the same work, it was shown that the four selected attributes are in a causal relationship. Thus, a causal graph seems to be more appropriate for their modelling. They are represented as a set: $< k_i(t_j), A_i(t_j), G_i(t_j), R_i(t_j) >$ where

- $k_i(t_j)$: the set of knowledge encapsulated in the agent a_i at time t_j.
- $A_i(t_j)$: the value of the executed action by the agent a_i at time t_j.
- $G_i(t_j)$: the value of the goal that has to be satisfied by the agent a_i at time t_j.
- $R_i(t_j)$: it includes a communication details namely the sender identifier, the receivers identifiers and the content of the message.

This graph was called KAGR graph in Sebai and Chaari (2014). It allows the representation of any modification at the level of agents internal properties, in between interactional ones. In the context of open MAS, the represented modifications by the KAGR graph are:

(i) **Knowledge addition**: knowledge addition may be caused by a communication between two agents. That's to say, when an agent a_i interacts with an agent a_j at time t, new knowledge can be added to a_j at time t′ in order to achieve a certain goal $G_i(t)$. This is graphically represented in the KAGR graph as in Figure 3. We denote by + a positive influence and by – a negative one between the two attributes.

(ii) **Action addition**: an action may be added into an agent for three reasons:
   (a) An already carried out action at time t can promote the possibility of realizing another one by the same agent. This causes the addition of a new action to its set of actions A_i at time t′.
   (b) The addition of new knowledge k_i at time t to an agent a_i may favour the execution of a new action by the same agent at time t'.
   (c) When an agent a_j interacts with another agent a_i at time t, a new action may be executed by a_i at time t', which causes its addition to the set of actions A_j.

   Action addition is graphically represented as in Figure 4.

(iii) **Goal update**: a reached goal at time t by an agent ai can favour the achievement of another goal at time t'. We call this, a goal modification. It is represented as shown in Figure 5.

(iv) **Interaction addition**: an established interaction by an agent a_i may be the consequence of another interaction established by an agent a_j. We call this modification, interaction addition. It is represented as shown in Figure 6.

(v) **The removal of knowledge/actions/goals/interactions**: such removal can be caused only by the physical suppression of agents.

Following the description above, we propose to call our model an evolving KAGR graph. In order to more clarify it, we give an illustrating example in the following Figure 7.

The graph below shows how agents and their internal attributes evolve in an open environment. The described system is decomposed of four agents whose identifiers are id_1, id_2, id_3 and id_4. Structural openness is shown at the first layer. We find that agents id_1, id_3 and id_4 appear only once in the system at, respectively, time t_1 and t_2 and remain permanent for the rest of the system time evolution. As

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

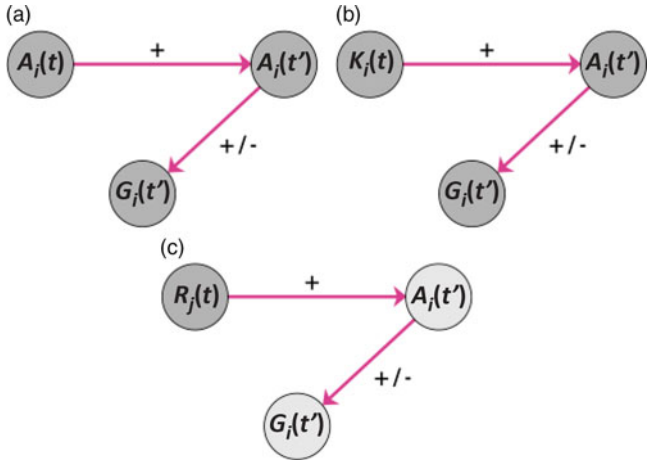


Figure 4 The addition of new actions to an agent a_i



Figure 5 Goal update in an agent a_i



Figure 6 Interaction addition to an agent a_i

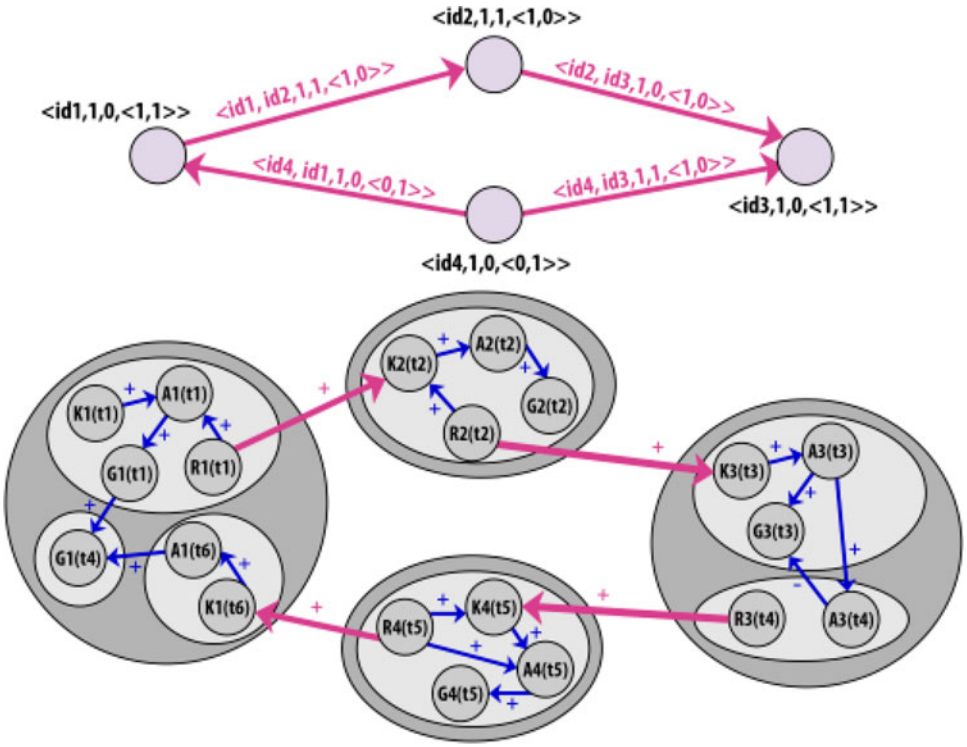


Figure 7 An example of a KAGR evolving graph

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

for the agent id_2, it is added once at time t_1 and removed once at time t_2. Besides, four interactions were established:

- An interaction between id_1 and id_2 is established at time t_1 and disappeared at time t_2.
- An interaction between id_2 and id_3 is established at time t_1 and remains until t_2.
- An interaction between id_3 and id_4 is established at time t_2.
- An interaction between id_4 and id_1 is established at time t_2.

In the second layer, we are interested in the internal attributes evolution for each agent. For example, at time t_1, agent id_1 establishes an interaction $R_1(t_1)$ with agent id_2. This causes the addition of new knowledge to the set k_2 at time t_2. As a consequence, a new action is executed by the latter agent and added to the set A_2 at time t_2 in order to reach a goal $G_2(t_2)$. Besides, agent id_2 establishes an interaction $R_2(t_2)$ with agent id_3. This interaction has a double effect: the addition of new knowledge to k_2 at time t_2 and new ones to k_3 at time t_3. The latter knowledge addition causes the execution of a new action by id_3 in order to realize a certain goal $G_3(t_3)$. This action promotes the execution of a new one by the same agent at time t_4 to reach the same goal $G_3(t_3)$. At the same time, two internal modifications are undergone in the system. The first is a goal update into agent id_1 from $G_1(t_1)$ to $G_1(t_4)$. The second is a new interaction between agents id_4 and id_2 through the relation $R_3(t_4)$. The latter has as a consequence the addition of new knowledge to the set k_4 at time t_5 which makes the corresponding agent able to execute new actions and to add them to the set A_4 in order to reach the goal $G_4(t_5)$. Also, a new relation $R_4(t_5)$ is established between agents id_4 and id_1. It adds new knowledge at both the set k_4 at time t_5 and set k_1 at time t_6. This causes the addition of new action into A_1 at time t_6 in order to reach the goal $G_1(t_4)$.

5 Validation

In order to validate our proposed model, we test it on a multi-agent rescue application. The latter is described as follows.

5.1 Agent-based test application

The tested multi-agent application is a rescue simulator. It allows, at any time, the addition and removal of agents and interactions. Indeed, this application presents a natural disaster and shows how two teams of intelligent agents collaborate together in order to rescue the victims. We believe that testing such application is useful for us in the sense that we can take advantages of its dynamicity and openness to validate our proposal. Actually, our rescue simulator is developed upon the JADE framework and made of two agent types: Rescuer agents and Ambulances. In order to achieve their goal, these agents appear and disappear a number of times all along the simulation generating evolving data sets: each new simulation may provide a different result. That is why we will give only a test case in the next paragraph.

5.2 A test case results

In this paragraph, we show how our rescue simulator can be represented using an evolving KAGR graph. To obtain the results, we opt for an AspectJ programming and we use Graphstream, a Java Library which focuses on generating dynamic graphs. The following Figures 8 and 9 represent the results of a random test case.

5.3 The obtained graph usefulness

The obtained evolving KAGR graph can be used in order to realize multiple objectives. We manage to prove some:

(i) **Openness Simulation**: through the obtained graph above, we can describe our system agents behaviour. The latter is given in the following Figure 10.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

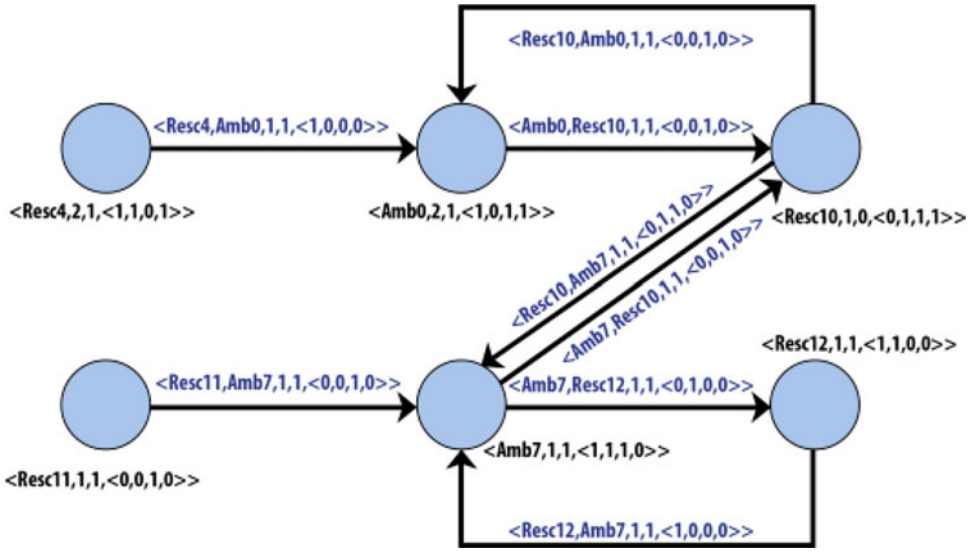


Figure 8 The obtained structural layer

Indeed, at time $t_1 = 2$ seconds, two rescuers Resc4 and Resc12 are present, respectively, in the positions (2,4) and (11,4). Both manage to find victims and to call for ambulances as a consequence. Ambulance Amb0, called by Resc4, executes the action: go-to(2,4) while having as a goal removing the victim from this position. Similarly, Ambulance Amb7 is called by Resc12. So, it executes the action: go-to(11,4) while aiming to remove the corresponding victim.

At time $t_2 = 3$ seconds, Amb0 disappears. However, Amb7 is still present. It interacts with Resc12 in order to delegate its current state: available. According to its positive response, Amb7 executes a new action: remove-victim-from (11,4) and it is planning to go back to the hospital as a next step. At the same time, a new 6 whom the identifier is Resc10 appears at position (6,7) where it finds a new victim. So, it calls for Amb7 in order to rescue the victim. As for Resc4, it is still present at time $t = 3$ seconds. It does not execute a new action but searches for a new victim to save instead.

At time $t_3 = 4$ seconds, some agents are no more present: Resc4 and Res12. However, a new rescuer Resc11 is added and Ambulance Amb0 appears again. Amb7 which was called by Resc10 at time $t_2 = 3$ seconds establishes two new interactions: the first is with Resc10 and the second is with Resc11 in order to inform them about its state: Busy. It is actually occupied by removing a victim from the position (11,4). Consequently, Resc10 looks for another ambulance and delegate its position to Amb0. At the same time, the new rescuer Resc11 finds a victim at position (6,8) and calls for ambulance Amb7.

At time $t_4 = 5$ seconds, Amb7 and Resc11 disappear. As for Amb0, it responds positively to Resc10 and gives it its current state: available. So, it executes new actions: go-to (6,7) to remove the corresponding victim. We notice that Resc4 reappears and tries to find a new victim.

By obtaining such description, we can say that we manage to simulate the behaviour of agents in an open environment and for a specific situation using a generic evolving KAGR Graph. The generated model shows how our simulator agents evolve in time: they appear and disappear in an unpredictable manner. All along the simulation, they were acquiring new knowledge, executing new actions, updating goals and establishing new interactions when needed.

(ii) **Openness management**: the obtained evolving KAGR graph shows that openness was managed at two levels: at the structural level where agents are directly connected via interactions following a peer-to-peer topology and at the functional level where internal properties are managed in a causal way.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Table 2 Structural openness evaluation using evolving KAGR graph

Metrics	Values	Interpretation
Alpha index	$\alpha = 3$	$\alpha > 0$: our system is increasing addition. It allow more than removal of agents. We can say that such index reflects how intense rescuers efforts are, in order to save the victims
Lambda index	$\lambda = 0$	$\lambda = 0$: the number of added interactions is equal to the number of removed ones. This shows that communication between agents was managed in a well-balanced way in order to reach the system global goal
Structural distribution	$\rho = 1$	Our test case scenario presents one collaborative group of agents. We say that in our case, the system have a low performance

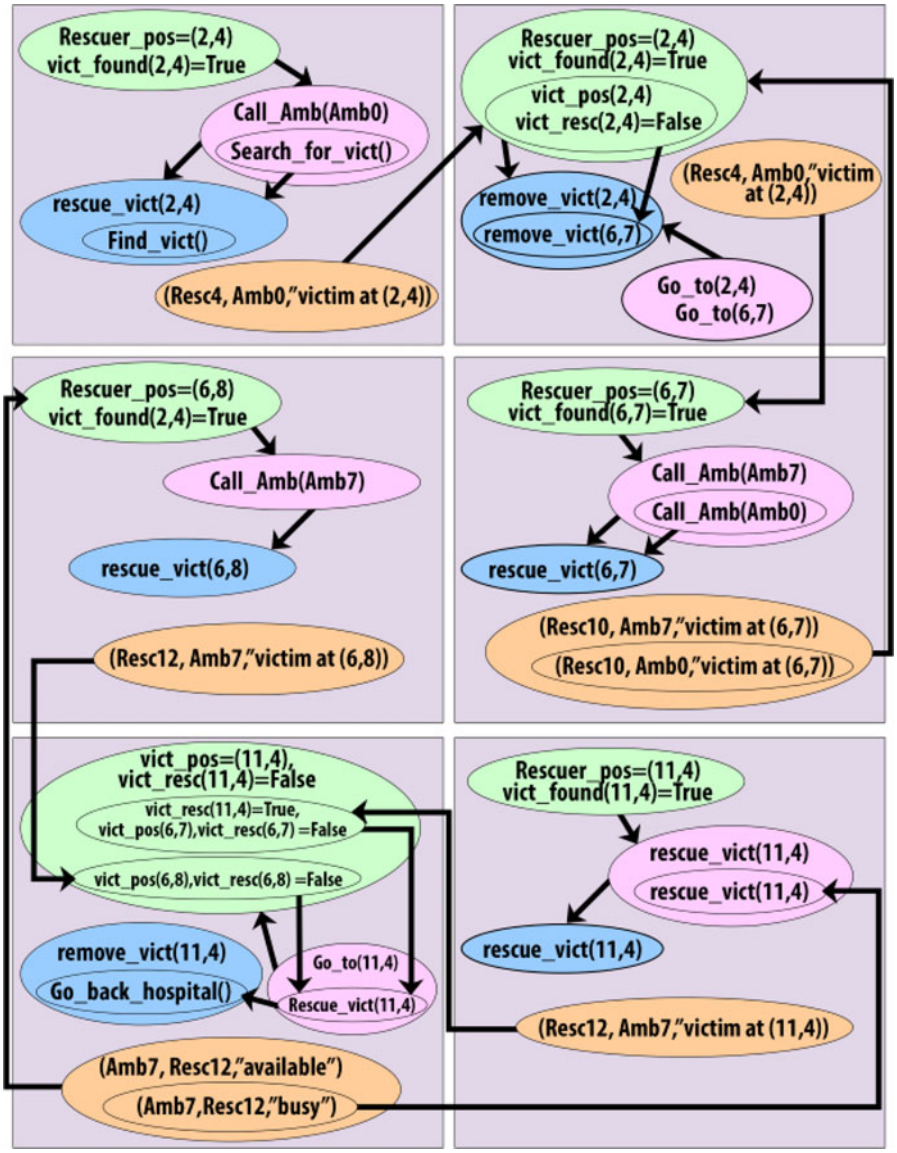


Figure 9 The obtained functional layer

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

	Agents	*K*	*A*	*G*	*R*
$t0 = 0$ seconds	Ø	Ø	Ø	Ø	Ø
	Resc4	Rescuer_position=(2,4) Victim_found_at(2,4)=True	Call_Ambulance (Amb0)	Rescue_victim (2,4)	(Resc4,Amb0, "victim at (2,4)")
	Amb0	Victim_position=(2,4) Victim_rescued_at(2,4)=False	Go_to(2,4)	Remove_victim (2,4)	Ø
	Resc12	Rescuer_position=(11,4) Victim_found_at(11,4)=True	Call_Ambulance (Amb7)	Rescue_victim (11,4)	(Resc12,Amb7, "victim at (11,4)")
$t1 = 2$ seconds	Amb7	Victim_position=(11,4) Victim_rescued_at(11,4)=False	Go_to(11,4)	Remove_victim (11,4)	Ø
	Resc4	Rescuer_position=(2,4) Victim_found_at(2,4)=False	Call_Ambulance (Amb0)	Find_victim()	Ø
$t2 = 3$ seconds	Resc10	Rescuer_position=(6,7) Victim_found_at(6,7)=True	Call_Ambulance (Amb7)	Rescue_victim (6,7)	(Resc10,Amb7, "victim at (6,7)")
	Amb7	Victim_position=(11,4) Victim_found_at(11,4)=False	Go_to(2,4) Remove_victim	Go_back_hospital()	(Amb7,Resc12, "available")
	Resc12	Rescuer_position=(11,4) Victim_found_at(11,4)=True	Call_Ambulance (Amb7)	Rescue_victim (11,4)	Ø
	Resc11	Rescuer_position=(6,8) Victim_found_at(6,8)=True	Call_Ambulance (Amb7)	Rescue_victim (6,8)	(Resc11,Amb7, "victim at (6,8)")
	Amb7	Victim_position={(11,4),(6,7),(6,8 Victim_rescued_at(11,4)=False Victim_rescued_at(6,7)=False Victim_rescued_at(6,8)=False	Remove_victim (11,4)	Go_back_hospital()	(Amb7,{Resc10,Resc 11}, "busy")
	Resc10	Rescuer_position=(6,7) Victim_rescued_at(6,7)=False	Call_Ambulance (Amb0)	Rescue_victim (6,7)	(Resc10,Amb0, "victim at (6,7)")
$t3 = 4$ seconds	Amb0	Victim_position=(6,7) Victim_rescued_at(6,7)=False	Ø	Go-to(6,7)	Ø
	Amb0	Victim_position=(6,7) Victim_rescued_at(6,7)=False	Go_to(6,7)	Remove_victim (6,7)	(Amb7,Resc10, "available")
	Resc10	Rescuer_position=(6,7) Victim_found_at(6,7)=True	Call_Ambulance (Amb0)	Rescue_victim (6,7)	Ø
$t4 = 5$ seconds	Resc4	Ø	Search_for_victim()	Find_victim()	Ø

Figure 10 A random test case

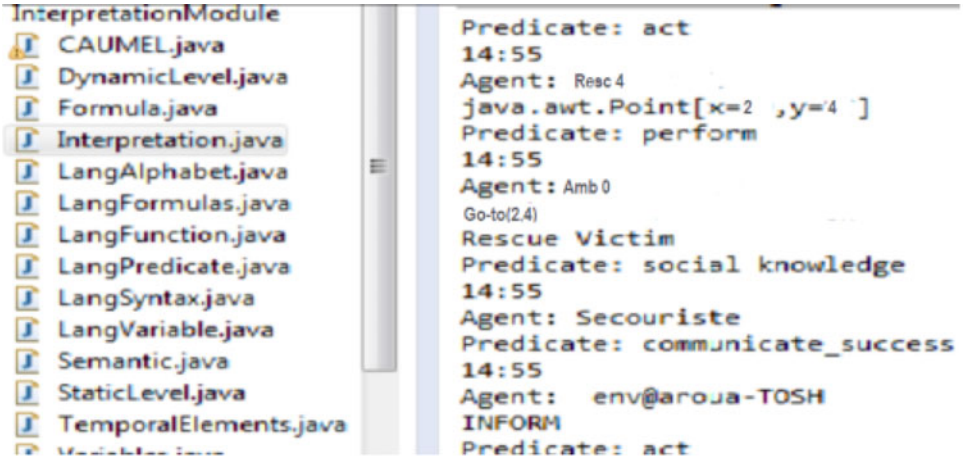


Figure 11 The obtained interpretation in CAUMEL

(iii) **Structural openness evaluation**: the obtained graph can permit the evaluation of the structural openness. Indeed, we apply some metrics developed in Hattab *et al*. (2017), and we obtain the results in the following Table 2.

(iv) **Openness analysis and interpretation**: in this paragraph, we demonstrate that the analysis and interpretation of openness are possible through our evolving KAGR graph. To this end, we use the same system in Sebai and Chari (2015): an explicative knowledge-based system. The latter

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Table 3 Comparative table of existing models and g KAGR graph

	Structural openness	Functional openness	Interactional openness	Objectives of use
Centralized model	√			
Distributed model	√			Openness management
Hybrid model	√			
Peer to peer model	√			
Reorganizational model	√			
Role-based model		√		Openness design
Service-oriented model		√		
Logic-based model		√		
Normative models and electronic institutions		√		
Algebraic model	√			Openness simulation
Graph theory	√	√		Structural openness evaluation Openness analysis
Semantic models		√	√	Openness analysis Representation of knowledge exchange
Finite-state automata		√		
Markov Chain model		√		Stability verification
Formal methods		√		
AIPs models			√	Agent compliance and conformance verification
Markov-decision process		√		Decision making
Trust and reputation models			√	Trust and reputation modelling
Coordination graph		√		Decision making
Statistical techniques		√		
Evolving KAGR graph	√	√	√	Openness management Openness simulation Structural openness evaluation Openness analysis and interpretation

integrates an interpretation module which analyzes the KAGR graph using a specific language called CAUMEL and generates an explication as a result. In the following Figure 11, we present a part of the obtained interpretation via CAUMEL.

Finally, compared to the models found in the literature, we say that our proposed model has two major advantages. It is, as shown in Table 3 generic and multi-objective which makes its use easier and less costly. In the following table, we grouped the previously described models by objectives of use. The symbol '√' means that the model represent the indicated type of openness.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

6 Conclusion

Open MAS are complex systems whose behaviour is difficult to understand. In order to reduce this complexity, several researchers have resorted to the modelling technique. The present paper contains two proposals. The first is a three-category classification proposal of open MAS existing models. At this level, we manage to distinguish between three types of models in the literature: structural models describing how MAS openness is seen from an external point of view while considering changes on the MAS structure, functional models showing the internal behaviour of agents and how it changes over time, and interactional models representing the agents communicative behaviour and the corresponding followed rules and protocols. The second proposal is about to define a generic model for representing MAS openness, a model that provides a general and shared understanding of openness property and which is useful for multiple objectives. Based on a literature review, we manage to define the sought model. We call it an evolving KAGR graph. It consists in a combination of two paradigms: an evolving graph to focus on structural openness aspects namely addition and removal of agents in time and KAGR model to put the spotlight on both functional and interactional openness, more particularly the evolution of agents internal properties namely: Knowledge, Actions, Goals and Relations. Our model was tested on a rescue multi-agent simulator and we managed to validate it through a test case scenario. Also, we proved its usefulness for different objectives. We hope that such proposal will be a stimulus for further study, development, and maturation of open multi-agent environment field. We particularly believe that it may help us in evaluating open MAS in a generic way. Such evaluation will permit the improvement of MAS in the sense that they would be more accurate and more realistic. We plan, as a next step, to exploit our model and use it in different MAS applications.

References

Abdul-Rahman, A. & Hailes, S. 2000. Supporting trust in virtual communities. In *Proceedings of Hawaii's international Conference on Systems Sciences*. IEEE.

Alberti, M., Gavanelli, M., Lamma, E., Chesani, F., Mello, P. & Torroni, P. 2006. Compliance verification of agent interaction: a logic-based software tool. *Applied Artificial Intelligence* **20**(2–4), 133–157.

Aldeeb, A., Crockett, K. & Stanatan, M. J. 2008. Multi-Agent based peer-to-peer workflow management system. In *Scope of the Symposium*, **11**.

Alechina, N., Dastani, M. & Logan, B. 2018. Norm specification and verification in multi-agent systems. *Journal of Applied Logics* **5**(2), 457.

Amaral, C. J. & Hübner, J. F. 2019. GoOrg: Automated organisational chart design for open multi-agent systems. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 318–321. Springer.

Ancona, D., Ferrando, A. & Mascardi, V. 2017. Parametric runtime verification of multiagent systems. In *AAMAS*, 17, 1457–1459.

Angluin, D., Aspnes, J., Fischer, M. J. & Jiang, H. 2008. Selfstabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **3**, 13.

Argente, E., Botti, V., Carrascosa, C., Giret, A., Julian, V. & Rebollo, M. 2011. An abstract architecture for virtual organizations: the THOMAS approach. *Knowledge and Information Systems* **29**, 379–403.

Artikis, A., Sergot, M., Pitt, J., Busquets, D. & Riverat, R. 2016. Specifying and executing open multi-agent systems. In *Social Coordination Frameworks for Social Technical Systems*, 197–212. Springer, International Publishing.

Belardinelli, F., Boureanu, I., Dima, C. & Malvone, V. 2019. Verifying strategic abilities in multi-agent systems with private data-sharing. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 1820–1822. International Foundation for Autonomous Agents and Multiagent Systems.

Belardinelli, F., Lomuscio, A. & Malvone, V. 2019. An abstraction-based method for verifying strategic properties in multi-agent systems with imperfect information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 6030–6037.

Baldoni, M., Baroglio, C., Martelli, A. & Patti, V. 2006. Verification of protocol conformance and agent interoperability. In *International Workshop on Computational Logic in Multi-Agent Systems*, 265–283. Springer.

Bijani, S. 2012. A review of attacks and security approaches in open multiagent systems. *Artificial Intelligence Review* **42**(4), 607–636.

Bijani, S. & Robertson, D. 2011. Intrusion detection in open peer-to-peer multi-agent systems. In *IFIP International Conference on Autonomous Infrastructure, Management and Security*, 177–180. Springer.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Boissier, O., Gitton, S. & Glize, P. 2004. Caractéristiques des systems et des applications, Observatoire Francais des Techniques Avancées. Systèmes multi-agents/Observatoire francais des techniques avancées, ARAGO 29, 25–54, Diffusion Editions & Tec Doc.

Calmet, J., Daemi, A., Endsuleit, R. *et al.* 2003. A liberal approach to openness in societies of agents. In *International Workshop On Engineering Societies in the Agents World*, 81–92. Springer.

Carbo, J., Molina, J. M. & Davila, J. 2003. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems* **12**, 135–155.

Carter, J., Bitting, E. & Ghorbani, A. 2002. Reputation formalization for an information-sharing multi-agent systems. *Computational Intelligence* **18**, 515–534.

Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E. & Botti, V. 2009. Service oriented MAS: an open architecture. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 1291–1292. International Foundation for Autonomous Agents and Multiagent Systems.

Castelfranchi, C. & Falcone, R. 1998. Social trust. In *Proceedings of the First Workshop on Deception, Fraud and Trust in Agent Societies*, Minneapolis, USA, 35–49.

Chandrasekaran, M., Eck, A., Doshi, P. & Soh, L. 2016. Individual planning in open and typed agent systems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 82–91. AUAI Press.

Chen, B., Chen, X., Timsina, A. & Soh, L. 2015. Considering agent and task openness in ad hoc team formation. In *Proceedings of the 2015 International conference on Autonomous Agents and Multi-Agent Systems*, 1861–1862. International Foundation for autonomous Agents and Multi-Agent Systems.

Chen, Y. & Chen, X. 2019. Research on knowledge graph application technology. *Journal of Physics: Conference Series* **1187**(4), 042083. IOP Publishing.

Ciortea, A., Mayer, S., Gandon, F., Boissier, O., Ricci, A. & Zimmermann, A. 2019. A decade in hindsight: the missing bridge between multi-agent systems and the World Wide Web. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 1659–1663. International Foundation for Autonomous Agents and Multiagent Systems.

Decker, K., Sycara, K. & Williamson, M. 1997. Middle-Agents for the internet. *In IJCAI*, 578–583.

de Brito, M. & Hubner, J. 2017. Architecture of an institutional platform for multi-agent systems. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, Nice, France, 313–329.

De Pinninck, A. P., Sierra, C. & Schorlemmer, M. 2010. A multiagent network for peer norm enforcement. *Autonomous Agents and Multi-Agent Systems* **21**(3), 397–424.

Derakhshan, F., Bench-Capon, T. & McBurney, P. 2013. Dynamic assignment of roles, rights and responsibilities in normative multi-agent systems. *Journal of Logic and Computation* **23**(2), 355–372.

De Wilde, P. & Briscoe, G. 2011. Stability of evolving multi-agent systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **41**(4), 1149–1157.

Esfiandiari, B. & Chandresekharan, S. 2001. On how agents meke friends: mechanism for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies*. Montreal, Canada, 27–34.

Ferreira, A. 2002. On models and algorithms for dynamic networks: the case for evolving graphs. In *4e Rencontres Francophones sur les Aspects Algorithmiques des Tlcommunications*, Mze, France.

Ferreira, A. 2003. *Building a Reference Combinatorial Model For Dynamic Networks: Initial results in Evolving Graphs. PhD dissertation*, INRIA.

Fornara, N., Vigano, F., Verdicchio, M. & Colombetti, M. 2008. Artificial institutions: a model of institutional reality for open multiagent systems. *Artificial Intelligence and Law* **16**, 89–105.

Gan, K. S., Chin, K. O., Anthony, P. & Hamdan, A. R. 2017. A FIPA-ACL ontology in enhancing interoperability multiagent communication. In *International Conference on Computational Science and Technology*, 151–160. Springer.

Guan, Y., Ji, Z. & Zhang, L. 2017. Controllability of multi-agent systems under directed topology. *International Journal of Robust and Nonlinear Control* **50**(8), 809–812.

Guo, J., Liu, W., Xu, L. & Xie, S. 2019. Adaptive graph planning protocol: an adaption approach to collaboration in open multi-agent systems. In *CCF Conference on Computer Supported Cooperative Work and Social Computing*, 297–303. Springer.

Hattab, S., Ben Hmida, F. & Lejouad Chaari, W. 2017. Using evolving graphs to evaluate structural openness in multi-agent systems. In *Proceedings of the 19th International Conference on Entreprise Information Systems*, 591–598. SCITEPRESS.

Hendler, J. 2001. Agents and the semantic web. *IEEE Intelligent Systems* **16**(2), 30–37.

Hendler, J. 2007. Where are all the intelligent agents?. *IEEE Intelligent Systems* **22**(3), 2–3.

Hendrickx, J. M. & Martin, S. 2016. Open multi-agent systems: Gossiping with deterministic arrivals and departures. In *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on Communication, Control, and Computing*, 1094–1101. IEEE.

Herzig, A., Lorini, E., Hubner, J. F., Ben-Naim, J., Castelfranchi, C., Demelombe, R., Longin, D. & Vercouter, L. 2008. Prolegomena for a logic of trust and reputation. In *NORMAS'08*, 143–157.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Huang, X., Ruan, J., Chen, Q. & Su, K. 2016. Normative multiagent systems: a dynamic generalization. arXiv preprint arXiv:1604.05086.

Hyunh, T., Jennings, N. R. & Shadbolt, N. R. 2006. An integrated trust and reputation model for open multi-agent systems **13**, 119–154.

Jirkovsky, V. & Kadera, P. 2019. Data exchange ontology for interoperability facilitation within industrial automation domain. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, 145–158. Springer.

Jonge, D., Rodriguez-Aguilar, J. A., Rosell I Gui, B. & Sierra, C. 2014. Infrastructures to engineer open agent environments by means of electronic institutions. In *Proceedings of the 4th International Workshop on Agent Environments for Multi-Agent*, 232–254. Springer-Verlag.

Jsang, A., Ismail, R. & Boyd, C. A. 2007. Survey of trust and reputation systems for online service provision. *Decision Support Systems* **43**, 618–644.

Jumadinova, J., Prithviraj, D. & Leen-Kiat, S. 2014. Strategic capability-learning for improved multi-agent collaboration in ad hoc environmrnts. *IEEE Transactions on Systems, Man, and Cybermetrics: Sytems* **44**, 1003–1014.

Kanfert, J., Scharf, H., Edenhofer, S., TomForde, S., Hahner, J. & Muller-Schoeler, C. 2014. A graph analysis approach to detect attacks in multiagent systems at runtime. In *IEEE Eighth International Conference on Self Adaptative and Self Organizing Systems*, London, UK.

Kouvaros, P., Lomuscio, A., Pirovano, E. & Punchihewa, H. 2019. Formal verification of open multi-agent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 179–187. International Foundation for Autonomous Agents and Multiagent Systems.

Liu, L. & Ji, Z. 2018. Controllability of multi-agent systems based on path and cycle graph. *International Journal of Robust and Nonlinear Control* **28**(1), 296–309.

Liu, X., Lin, H. & Chen, B. M. 2013. Graph-theoretic characterisations of structural controllability for multi-agent system with switching topology. *International Journal of Control* **86**(2), 222–231.

Lu, Z., Zhang, L. & Wang, L. 2018. Controllability of discrete-time multiagent systems with switching topology. *International Journal of Robust and Nonlinear Control* **28**(6), 2560–2573.

Mahmoud, M. A., Ahmad, M. S., Mohd Yusoff, M. Z. & Mustapha, A. 2014. A review of norms and normative multiagent systems. *The Scientific World Journal* **2014**.

Luck, M., McBurney, P. & Preist, C. 2003. Agent technology: enabling next generation computing (a roadmap for agent based computing). AgentLink.

Ma, Z., Schultz, M. J., Christensen, K., Vrbak, M., Demazeau, Y. & Jrgensen, B. N. 2019. The application of ontologies in multi-agent systems in the energy sector: a scoping review. *Energies* **12**(16), 3200.

Massonet, P., Deville, Y. & Nve, C. 2002. From AOSE methodology to agent implementation. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, Italy, 27–34.

Milner, R. 2009. *The Space and Motion of Communicating Agents.* Cambridge University Press.

Morales, J., Wooldridge, M., Rodríguez-Aguilar, J. A. & Lopez-Sanchez, M. 2017. Synthesising Evolutionarily Stable Normative Systems. arXiv preprint arXiv:1710.00709.

Padovan, B., Sackmann, S., Eymann, T. & Pippow, I. 2002. A prototype for an agent-based secure electronic marketplace including reputation-tracking mechanisms. *International Journal of Electronic Commerce* **6**, 93–113.

Perelli, G. 2019. Enforcing equilibria in multi-agent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 188–196. International Foundation for Autonomous Agents and Multiagent Systems.

Petsch, M. 2002. Openness ans security in the FIPA standard. In *Proceedings of MAI*, **2**, 13–26.

Pico-Valencia, P., Holgado-Terriza, J. A. & Senso, J. A. 2019. Towards an internet of agents model based on linked open data approach. *Autonomous Agents and Multi-Agent Systems* **33**(1–2), 84–131.

Pigné, Y. 2009. *Modelisation et Traitement decentralise des Graphes Dynamiques: application aux rseaux dynamiques ad hoc*. *PhD dissertation*, University of Havre, France, 1–41.

Pinyol, I. & Sabater-Mir, J. 2013. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review* **40**, 1–25.

Platon, E., Sabouret, N. & Honiden, S. 2008. An architecture for exception management in multiagent systems. *International Journal of Agent-Oriented Software Engineering*, **2**(3), 267–289.

Rahmani, A., Ji, M. & Mesbahi, M. 2009. Controllability of multi-agent systems from a graph theoretic perspective. *SIAM Journal of Robust and Optimization* **48**(1), 162–186.

Regan, K. & Cohen, R. 2005. Indirect reputation assessment for adaptive buying agents in electronic markets. In *Business Agents and the Semantic Web Workshop*.

Sabater, J. & Sierra, C. 2001. Regret: a reputation model for gregarious societies. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies*, Montreal, 61–69.

Sabater-Mir, J., Paolucci, M. & Conte, R. 2006. Repage: reputation and image among limited autonomous partners. *Journal of artificial societies and social simulation* **9**(2).

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press

Sebai, A. H. & Chaari, W. L. 2014. CAUMEL: a temporal logic based language for causal maps to explain agent behaviors. In *Agent and Multi-Agent Systems: Technologies and Applications. AISC*, Jezic, G., Kusek, M., Lovrek, I., Howlett, R. J. & Jain, L. C. (eds.), **296**, 127–138. Springer.

Sebai, A. H. & Chaari, W. L. 2015. Explain-M as: an agent behavior explanation system. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 284–287. Springer.

Sehaba, K. 2002. Gestion de louverture des systems multi-agents. Memoire de DEA. Universite de la Rochelle, France.

Sehory, O. 2001. Software architecture attributes of multi-agent systems. In *Agent-Oriented Softaware Engineering*, 770–790. Springer.

Spanoudakis, N., Akasiadis, C., Kechagias, G. & Chalkiadakis, G. 2019. An open MAS services architecture for the V2G/G2V problem. In *AAMAS 19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Tian, L., Guan, Y. & Wang, L. 2018. Controllability and observability of multi-agent systems with heterogenous and switching topologies. *International Journal of Control* **93**(3), 437–448.

Valckenaers, P., Sauter, J., Sierra, C. & Rodriguez-Aguilar, J. A. 2007. Applications and environments for multi-agent systems. *In Autonomous Agents and Multi-Agent Systems*, 61–85.

Vercouter, L. & White, E. B. 2000. Conception et mise en oeuvre des systemes multi-agents ouverts et distribues. Ecole Nationale Superieure des Mines de Saint Etienne, Universite Jean Monnet-Saint-Etienne.

Viana, M., Caetano, L., Cunha, F., Alencar, P. & Lucena, C. 2018. Governance in adaptive normative multiagent systems for the internet of smart things: challenges and future directions. In *2018 IEEE International Conference on Big Data (Big Data)*, 5193–5196. IEEE.

Weyns, D. 2010. *Architecture-Based Design of Multi-Agent Systems*. Springer Science & Business Media.

Weyns, D., Michel, F., Van Dyke Parunak, H., Boissier, O., Schumacher, M. & Ricci, A. 2015. Agent environments for multi-agent systems – A Research roadmap. In *Agent Environments for Multi-Agent Systems* IV, 3–21. Springer.

Xu, H., Zhang, X. & Patel, R. J. 2007. Developing role-based open multi-agent software systems. *International Journal of Computational Intelligence Theory and Practice* **2**(1).

Zaki, A., Attia, M., Hegazy, D. & Amin, S. 2016. Comprehensive survey dynamic graph models. *International Journal of Advanced Computer Science and Applications* **7**(2), 573–582.

Zegzhda, D. P., Stephanora, T. V. & Suprun, A. F. 2017. Multiagent systems controllability evaluation using the multilevel structure of the graph agents. *Automatic Control and Computer Sciences* **50**(8), 809–812.

https://doi.org/10.1017/S0269888920000429 Published online by Cambridge University Press