Noname manuscript No.
(will be inserted by the editor)

The Role of the Environment in Agreement Technologies

E. Argente · O. Boissier · C. Carrascosa · N. Fornara · P. McBurney · P. Noriega · A. Ricci · J. Sabater · M. I. Schumacher · C. Tampitsikas · K. Taveter · G. Vizzari · G. Vouros

the date of receipt and acceptance should be inserted later

Abstract The notion of Multi-Agent System environment is currently considered as a mediating entity, functioning as enabler but possibly also as a manager and constrainer of agent actions, perceptions, and interactions. In this paper, we analyze how environment could be a first class abstraction to support the building, development and management of Agreements in decentralized and open systems between autonomous agents. To this aim we analyze the synergies between environment and the foundational dimensions of agreement technologies such as semantics, norms, organizations, argumentation & negotiation, trust.

Keywords Environment, Semantic, Norm, Organization, Argumentation, Negotiation, Trust, Agreement technologies

1 Introduction

In Agent-Oriented Software Engineering, the notion of Multi-Agent System (MAS) *environment* has gained a key role, becoming a mediating entity, functioning as enabler but possibly also as a manager and constrainer of agent actions, perceptions, and interactions [1] while addressing the requirements of openness and scalability. According to such a perspective, the environment is not a merely passive source of agent perceptions and target of agent actions which is, actually, the dominant perspective in agency, but a *first-class abstraction* that can be suitably designed to encapsulate some fundamental functionalities and services, such as coordination and organization, besides agent mobility, communications, security, etc [39].

In particular, the environment dimension appears to intersect with all the dimensions that should be addressed to define an *agreement* between autonomous agents,

{eargente,carrasco}@dsic.upv.es, olivier.boissier@emse.fr, nicoletta.fornara@usi.ch, peter.mcburney@kcl.ac.uk, {pablo,jsabater}@iiia.csic.es, {michael.schumacher,charalampos.tampitsikas}@hevs.ch, kuldar.taveter@ttu.ee, giuseppe.vizzari@disco.unimib.it, georgev@aegean.gr

[1] see [49] for comprehensive surveys

that is, all the different *Agreement Technologies* [2] (AT) giving support to the building, development and management of agreements in decentralized and open systems between autonomous agents. Those dimensions are the ones related to the development of technologies dealing with: Semantics, Norms, Organizations, Argumentation & Negotiation, and Trust. The objective of this contribution is then to provide a comprehensive analysis and discussion about the role that could play the environment for the engineering of Agreement Technologies, along such dimensions that define the agreement technology conceptual space. Besides identifying points and features that could concern the individual dimensions alone, the aim is to finally provide a conceptual framework which links together such points across the dimensions and gives an answer to two basic background questions: *(i)* how the design and development of agreement agent-based technologies could be improved by introducing environment as a first-class abstraction besides agents and direct communication models; *(ii)* how the environment can facilitate reaching agreements with respect to the AT dimensions (semantics, trust, etc).

The paper is organized as follows. In section 2 we provide a global definition of environment as it is considered in multi-agent literature in order to define the framework in which the following sections place their contribution. We then draw some motivations from the analysis of different applications in the context of Agreement Technologies (sec. 3). The following sections then analyze and describe the relations between environment and semantics (sec. 4), and norms (sec. 5), and organizations (sec. 6), and argumentation & negotiation (sec. 7), and trust (sec. 8). We finish the paper by discussing the different features and challenges facing the use of environment for agreement technologies.

2 Environment Definition

The notion of *environment* is a primary concept in agent and multi-agent systems, being the computational or physical place where agents are situated, and providing the basic ground for defining the notions of agent perception, action and then interaction. Fundamental features of the agent abstraction are directly or indirectly related to the environment: reactivity is an obvious example, but also pro-activeness, being the notion of *goal*, which is actually the essential aspect of pro-active behaviours, typically defined in terms of states of the world that one or more agents aim to bring about.

Besides being considered a mere source of agents percepts and target of agent actions, the environment can be framed as a *first-class abstraction* in Agent Oriented Software Engineering [31, 49], i.e. a part of the MAS that can be designed and programmed to uniformly and effectively encapsulate and modularize functionalities and services out of the agents [39, 50, 51].

In that view, the responsibilities and functionalities of environments can be sumarized by the following three different levels of support, identified in [49] (see Fig. 1):

– *a basic level*, where the environment is exploited to simply enable agents to access the *deployment context*, i.e. the external hardware/software resources which the

[2] Agreement Technologies refer to computer systems in which autonomous software agents negotiate with each other, typically on behalf of humans, in order to come to mutually acceptable agreements.

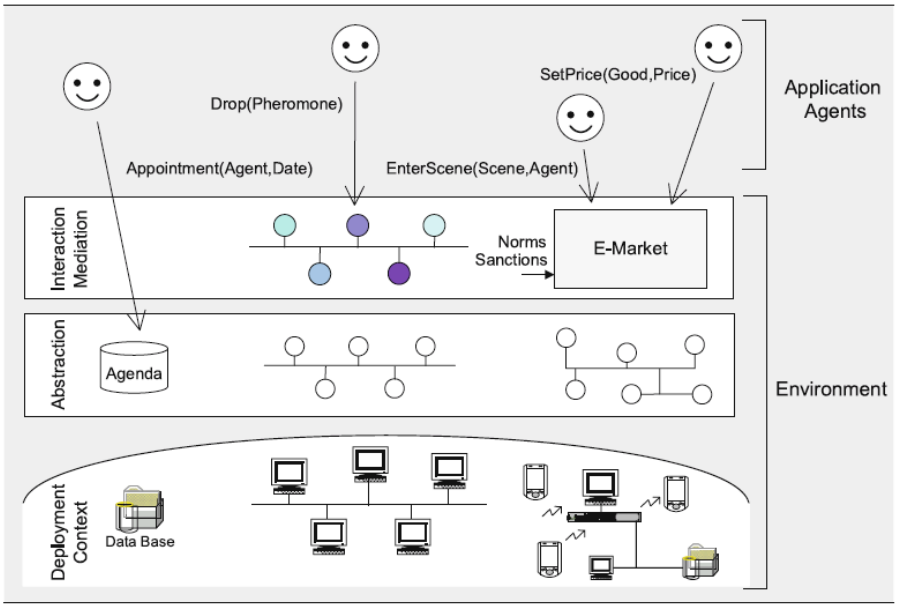

Fig. 1 Environment support levels, as depicted in [49]

MAS interacts with (sensors and actuators, a printer, a network, a database, a Web service, etc.);
- *abstraction level*, exploiting an environment abstraction layer to bridge the conceptual gap between the agent abstraction and low level details of the deployment context, hiding such low level aspects to the agent programmer;
- *interaction-mediation level*, where the environment is exploited to both regulate the access to shared resources, and mediate the interaction between agents—becoming then an enabler and facilitator of agent coordination, organization, cooperation.

These levels represent different degrees of functionality that agents can use to achieve their goals.

In this perspective, the environment becomes both a first-class abstraction for modelers / designers / developers and a first-class entity for agents. In the former view, it can be exploited at design/development time to modularize, encapsulate, structure and compose functionalities and services and possibly to define the software architecture of the MAS. To this purpose, in the literature concrete environment architectures have been proposed [48], as well as environment programming models [9] and APIs [39]. In the latter view, it is perceived and exploited by agents as part of their world, possibly reasoning about and adapting it at runtime. For instance, in artifact-based environments as introduced by the A&A conceptual model [34] and the CArtAgO platform [40] as well as in GOLEM [4], environment is structured and perceived by agents in terms of dynamic set of passive entities (artifacts or objects) with affordances that allow for observing and using them.

3 A Motivating Application

The notion of environment has found application in several fields. For instance *multi-agent simulations* such as pedestrians simulations (see, e.g., [1, 2]) are generally employed by decision makers to analyze the plausible outcomes of their choices. Applications with *spatial aspects* profit significantly from environment-enabled mechanisms to realize *control systems*. For instance, in [52], the environment is actually employed by agents controlling automatic guided vehicles managing the transportation of loads in an automated warehouse. Agents share the physical environment and coordinate in order to avoid collisions: they use the environment to project their movement intention to *detect and resolve potential conflicts* before they actually happen. *Ubiquitous and pervasive computing* applications also consider the spatial aspects of the environment in which devices are situated. The environment is used to *manage context awareness* information, providing indications of the local state of the system or even suggesting potentially exploitable services [26]. Other examples are applications sharing information. For instance in [24] medical information about patients, their histories and known conditions are shared. Patients are described according to the local rules of a given local healthcare system, but the patient might require some form of treatment when he/she is abroad. In this case, environments representing the distinct but integrated healthcare systems, react to requests coming from end-users by providing basic mechanisms for the translation of relevant pieces of information from the original ontology to the one adopted by the local system.

In order to further illustrate the environment dimension for Agreement Technologies, consider a personalized and social e-health system that will read the barcode on a package in order to identify a medicine [25, 41] and give access to a set of services. This will considerably diminish the possibility of buying a counterfeit product and will assist a patient by providing alternative medicines to choose from. Moreover, with access to the patient e-health profile, the system will filter out unsuitable medicines that can interact with the other on-going treatments or cause allergies. This will help a buyer to find the right product with a minimal intervention by the seller/pharmacist. In addition, the system offers the possibility to rate and review medicines anonymously. The ratings and comments offer actual feedback by the consumers.

The proposed system can be designed and implemented as a complex system with the architecture involving a combination of both software agents and services invoked by them, which are embedded in the agents environment. To come up with a well-justified decision for the architecture, the system can be analyzed and designed using the principles of agent-oriented modeling (AOM) for designing socio-technical systems [44]. In addition to modeling the knowledge and behaviours of agents as well as interactions between them by AOM, we should also represent by appropriate models how different aspects of agreement technologies manifest themselves in the environment that should be designed for the agents:

- *Semantics*: As the catalogues of medicaments are constantly growing, ontologies representing those catalogues may be integrated directly within the environment, in order to offer to all agents an up-to-date domain model.

- *Norms*: The environment may store different types of norms for different medicament market segments, some being stricter than the others.
- *Organizations*: The role model may differentiate between different sorts of patients, nurses, pharmacists and medical doctors. Specific patient groups (such as diabetic type 1 patients) may need to interact within specific environment spaces with specific roles and norms.
- *Argumentation & negotiation*: When searching for adequate medicines on a market, agents may use the environment to objectivize their negotiation processes (e.g. making it public to some control agents), in order to ensure the truthfulness of the negotiation and transaction.
- *Trust*: The environment may offer a mechanism for defining ratings on other users, and services to evaluate the impact of those ratings.

After having properly modeled the five AT dimensions of the environment, we can design and implement the application according to the models that represent both agents and their environment.

This example illustrates and motivates the potential of the environment point of view for defining agreement technologies. The next sections will describe and analyze the relations between environment and different aspects of agreement technologies in more detail.

4 Environment for Semantics

Semantics is about the meaning of symbols. Real-world semantics associate symbols and expressions in a language to entities and sets of entities in the real-world (e.g. medicines, allergies, people, services). Some of these entities belong to the basic level of the environment where agents are deployed (e.g. medicines, people and services), to the abstraction level (e.g. service profiles, categories of objects, roles played by people), or to the interaction/mediation layer of the environment (e.g "translation" services, argumentation services).

Semantics in Agreement Technologies play a twofold role: i) agreement technologies (e.g. argumentation [10]) bring agents into agreements on their representations, and ii) semantics drive agents to reach "meaningfull" agreements (e.g. on trust or norms [43] and [22]).

As far as the environment dimension is concerned, in the former case agents must be capable to compute "meaningful" correspondences between the symbols they use, so as to interact and coordinate effectively while acting in common environment regions. On the other hand, the environment, acting as a manager or interaction mediator must provide the appropriate facilities (e.g. services, repositories, information sources, argumentation spaces) for agents to reach semantic agreements.

For agents to compute "meaningful" correspondences between their representations, we may consider three different "levels" of disagreement:

- the first level concerns agents that share the same formal language (e.g. for specifying their ontologies), although they may not use the same vocabularies or conceptualizations of the environment entities,

- at the second level, agents use different languages and maybe different vocabularies and conceptualizations,
- and at the third level agents exploit implicit/informal semantics that are hand-wired into their decision/action cycle.

It is clear that moving from the first to the third level, the difficulty of establishing semantic agreements increases. Nevertheless, as the difficulty increases (something expected in open settings), the role of the environment towards establishing semantic correspondences should become more important, as it provides the common basic level, i.e. the actual context at which all agents are deployed.

To reach agreements, agents may face different challenges depending on the means used and on the heterogeneity of representing the environment. Agents that use commonly agreed formal languages may establish agreements on their representations by computing proper correspondences between the represented entities, also driven by the semantics of their common representation framework: For instance, agents need to consider the same identifiable artifact when they both refer to an entity, e.g. to the "X service" or to the "X person". However, these semantic correspondences may be subjective (the correspondences between agents A and B computed by A, may not agree to the correspondences computed by B), or objective (a distinguished trusted entity may be delegated to compute all correspondences). It must be pointed out that the computation of objective correspondences may be delegated to (an entity / service in) the environment. An important aspect in any case, is that agents in a community or in an ad-hoc team need to reach a coherent set of correspondences, so as to function coherently and interact consistently. In the case of subjective correspondences in large scale and open settings, this is a challenge.

Nevertheless, agents may need to uncover the semantics of terms and expressions they use by exploiting syntactic correspondences between their distinct representations. Semantics thus emerge and evolve out of syntactic correspondences. In such a case, the environment may provide the appropriate services to further drive the emergence of meaningful correspondences.

Finally, in the case where implicit semantics are hard-wired into the deliberation - reaction cycle of an agent, agents need to exploit the environment itself to establish connections between them and further constrain the actions of others. One further (challenging) possibility here is agents to build formal representations for their semantics and then, use these representations to establish connections and agreements to the others.

Considering the environment as a manager or as an interaction mediator, it must provide the appropriate facilities (e.g. services, repositories, information sources, argumentation spaces) for agents to reach semantic agreements. These interaction-mediation facilities constrain, or provide evidence to agents towards reaching agreements: In few words, the environment itself provides the context(s) where symbols can be interpreted. Thus, agents may interpret and constrain the interpretations of their representations while interacting with the environment (also in conjunction to other agents). For instance, to learn the semantics of service input/output parameters, agents may need to gather training examples, e.g. by exploiting other agents'

interactions. As another example, agents may share interactions so as to establish a common understanding of their trust assessments. Agents may further exploit contextual features to understand the meaning of tokens attributed to some resources. Services, parameters, interactions, and tokens are entities that agents share, and establish a context of interaction.

This leads to situated cognition where knowledge exists inseparably from context. Therefore, establishing agreements is seen in terms of agents' increasingly effective performance across situations rather than in terms of an "objective" combination of knowledge, since what is known is co-determined by the agents and their context.

5 Environment for Norms

Norms are fundamental elements for regulating open interaction systems for autonomous heterogeneous agents where no assumption can be made on the behaviour of participating agents. Norms can be used to guide the activities of the interacting agents, they can directly affect the agents' expectations on the behaviour of the other agents, and they can affect the state of the entire system, that is, they can constrain the interactions for the achievement of some global desirable properties. Norms are a fundamental component of agreement technologies[3], indeed they are essential for regulating the spaces of interaction where agreements may be defined and negotiated by software agents that typically act on behalf of their human owners. Norms are usually used to define the obligations, the permissions, and the institutional powers to perform certain actions for the agents that play certain roles [7, 14, 16]. The study of those deontic concepts is fundamental also for the execution agreements, in fact for bringing agents to fulfill their agreements it is crucial to transform them into specific obligations and prohibitions. A requirement for norms is that they should be perceivable by the agents interacting through an open system. Moreover while some of them may be regimented (it is not possible for the agents to violate them), some other may be violated and mechanisms for their enforcement should be realized. Norms can represent physical or institutional constrains for the behaviour of agents. A crucial characteristic of physical norms is that they are situated [32], that is, they are meant to be followed within certain physical spaces and for a certain period of time. In a similar way institutional norms that regulate the social interactions of agents should be situated in limited spaces of interaction [45] and active for a given interval of time. For example a space may be used for representing a group of agents that collaborate on a given task for a month. This aspect is important because in a given open system many norms may be specified but just some of them may be relevant in a given context of interaction. Moreover it is crucial for an agent being able to clearly single-out the set of norms that are regulating its behaviour in a given situation. As presented in Section 2, the functionalities of multi-agent environments' can be schematized in different levels of support. The higher of these levels is the interaction mediation level

[3] Memorandum of Understanding of the Agreement Technologies COST Action IC0801, `http://w3.cost.eu/fileadmin/domain_files/ICT/Action_IC0801/mou/IC0801-e.pdf` last accessed 27th September 2012

devoted to mediate, coordinate, and organize agents' interaction, that is functionalities that are strictly connected with the notion of norm. It is therefore reasonable to propose to consider multi-agent environment a crucial component of the architecture of open systems that should be in charge of providing:

- Mechanisms for representing norms, making them *perceivable* by the agents. In existing agent environment meta-models, norms may be represented as first-class abstractions by using for example objects [45], normative objects [32] or artifacts [39].The perception of norms depends on their contextual level, that is, depends on the role of the agent, the position of the agent inside the environmental spaces, the state of the environment, and the state of the norm. The perception of norms when situated inside an agent environment contributes to the optimization of agent reasoning and goal planning. Since norms can be discovered and perceived, the agents can adapt their strategies according to the norms, measuring in a clear and concrete way the consequences of their actions into their institutional context.
- Mechanisms for *contextualizing* at run-time norms that are expressed at design-time in an abstract form in terms of roles and that may contain some un-specified parameters [15]. Think for example to the mechanisms for transforming a generic obligation of the auctioneer of an auction to declare the winner of a run of the auction into a specific obligation related to specific agents that should be fulfilled within a specific deadline.
- A norm *enforcement* infrastructure, directly related to the event-based model of the environment and in particular a model of norms that apply as a response to the production of events. This level of norm support require that the environment should be able to offer the following functionalities: (i) monitoring of the state of norms in accordance with the event types that trigger the transition from one norm state to another; (ii) detecting norms violation, that is being in charge of checking the consistency of the produced events against the current state of norms and decide whether it will allow the effects of the events to happen or not; (iii) applying sanctions or giving rewards when norms are violated or fulfilled.
- Finally, agent environment can clearly define the *boundaries*, that is the environmental spaces where a norm can apply and the event types that trigger the norms. When an event type is produced inside one or more spaces of the environment, provided that a specific norm is linked to all these spaces, the norm will apply to all of them. This property of situated norms is fundamental when there exist different norms, may be also conflicting, situated in different spaces, which influence the behaviour of agents that participate in all these spaces.

In addition, when situated in an agent environment, norms can be used as computational entities which could be manipulated towards changing their content at run-time [46]. Since a norm is perceivable by many agents who share the same context, the agents through different schemas (e.g. voting) could decide to modify the norm. Similarly, the environment, while monitoring the effects of the norms to the agents' behaviour, detects how the norms affect the state of the system and can modify norms to preserve the stability of the system. An important challenge that should be addressed for making it possible to fully exploit, from the software engineer point of

view, existing environment frameworks [5, 39] for representing, managing, and enforcing situated norms is mainly due to the feasibility of easily extend the functionalities/services provided by environment frameworks for adding those required for norms management.

6 Environment for Organizations

Organizations in Multi-Agent Systems represent complex entities where a multitude of agents interact, within a structured environment aiming at some global purpose [8]. Concepts such as roles (or functions, positions), groups, tasks (or activities), interaction protocols (or dialogue structures), as well as norms or rules, are key elements for modeling, designing and implementing agent organizations.

In the Agreement Technologies domain, the Organization concept can be employed for two different purposes: (i) as a *facilitator* for reaching an agreement; or (ii) as a *result* of reaching an agreement. As facilitator, organizational structures can be employed for restricting the way agreements are reached by fixing the capabilities of the roles that agents can play inside the organization and the relationships among these roles (e.g. power, authority, compatibility, etc.). In this sense, the organizational structure imposes limits to which agents can negotiate with others, and which weight or importance will have their decisions inside the argumentation processes. For example, authority relationships in a hierarchical structure give more decision power to some roles than others, depending on their position in the organization in a negotiation process for reaching an agreement (e.g. supervisor vs. subordinate). As another example, in a hierarchical structure two subordinate agents cannot negotiate directly between them to reach an agreement on a salary increment, since this negotiation must be done with their supervisor. However, in a social cooperative structure like a federation or a team, their members would be able to reach a common agreement for a salary increment or benefit distribution.

As said above, an organization can also be the result of reaching an agreement. In this case, the agreement will determine which are the roles that can be played by agents, the specific relationships between these roles (i.e. social/power/authority relationships), the norms or restrictions that affect them, the capabilities of the roles, etc. For example, when two companies merge, they need to reach an agreement on all the terms of this merging, and the result of the agreement will be a new organizational structure, with new roles and relationships that might integrate the entities existing in the original companies.

To understand how organizations behave, the environment is fundamental. In human organizations, the environment encompasses all that is outside the organization: its suppliers, customers, competitors and institutions that regulate the behaviour of the members of the organization. Economic, geographic and political conditions are also part of the environment [47]. Therefore, the environment is the source of the resources required for the organization to survive [18], as it provides the materials, technology and members that the organization needs to develop its products and services as well as enough customers who consume these products and provide benefits

to the organization. Therefore, every opportunity to succeed and every threat to its existence comes from the environment [18]. In Multi-Agent Systems, the environment is mainly associated with the resources and applications that agents make use of, as well as those stakeholders or entities outside the organization that benefit, are dependent or facilitate the existence of the organization. This group includes suppliers, customers and users of the organization.

When modelling agent organization, the environment model normally provides constructs to represent the collection of resources in the space of the agent organization formed by non-autonomous entities that can be perceived and acted upon (manipulated, consumed, produced, etc.) by the component agents. AGRE [13], MASML [6], and AOM [44] are examples of organizational models (modelling techniques) that provide constructs to represent organization environment elements.

Recently, some approaches provide elements for helping the organization to make better profit of its environment. These approaches are mainly based on the Agents & Artifacts (A&A) conceptual framework [34], that models the environment around three main concepts:

- agents, the proactive entities of the system;
- artifacts, passive entities that help agents to reach their objectives by means of their provided functionality, and
- workspaces, used to model the topology of the environment.

For example, *artifacts for organizational mechanisms* [11, 12] provide several mechanisms for: helping agents to obtain information about organizations of their environment; modifying the global behaviour of the system by changing the incentive system of the organization; and updating the action space of agents. Another example is the set of *organizational artifacts* implemented in ORA4MAS [19], aimed at managing the current state of the organization in terms of groups, social schemes, and normative state. They provide actions for agents to proactively take part in an organization (adopting/leaving roles, committing to goals) and actions for organizational agents to manage the organization.

The environment can thus provide several mechanisms to allow agents know which are the organizations in the system and how to participate inside them. In fact, situated organizations may allow agents to perceive their state and properties in order for the agents to join an organization. And situated organizations can also simplify the discovering and solving process for interdependencies between organizations [45]. Furthermore, we can employ physical objects as mediators for the access to some organizations, so then using these physical objects may count as endorsing some role in the organizations [37]. This can help bridging the conceptual gap between the elements available at the organizational level, normally defined in terms of norms, roles, resources, global goals, stakeholders, etc.; and the notions and constructs adopted at the agent level, for example with mentalistic notions such as beliefs, desires, intentions of a BDI-like agent model.

As a result, social structures can be explicitly expressed and shaped through organizational concepts, which can be perceived and managed by means of elements of the environment, such as artifacts (e.g. the above mentioned “organizational arti-

facts" or the "artifacts for organizational mechanisms"). These social structures can then be exploited by agents to cope with the difficulties of solving complex tasks in a coherent and efficient manner.

7 Environment for Argumentation & Negotiation

Instantaneous agreement between multiple humans or machines is quite rare. Different people have different beliefs about the world, different preferences and goals, and different decision constraints, such as deadlines or available processing resources. Accordingly, before agreement between multiple people or machines can be reached on some matter, there is often a requirement for the identification and possible resolution of these differences. Agreement may be possible even without resolution of differences, as when two parties to an economic transaction both believe they have done better out of the deal than did the other party. Indeed, some technologies and methods for reaching agreement disregard differences between participants. For example, simple voting mechanisms pool the varied preferences of the participants without trying to resolve or even identify the various preferences. Similarly, auction protocols seek to find an agreement between potential buyers and sellers of some item without trying to explain or resolve the different beliefs of the participants over the item's value.

Many human interactions, however, do use methods for identifying and possibly resolving differences in the beliefs, preferences, or objectives of the participants. Chief among these methods are those based on argumentation — the requesting and giving of reasons for statements — and negotiation — the attempt to decide between mutually-incompatible goals. Human beings have been engaged in argumentation and negotiation since our earliest days as a species. It is not surprising therefore that we should seek to vest software agents with the same capabilities to argue, to negotiate, and to jointly deliberate about what to believe and what to do [27]. Given a system comprising multiple autonomous software agents, jointly interacting via argumentation to achieve their individual or shared objectives, there are several resources that the system environment may provide to the participating agents. These resources include:

- Shared ontologies and semantics for communications between different agents.
- Libraries of protocols for agent interaction, possibly including associated metadata and annotations, such as statements of possible protocol outcomes and statements and proofs of protocol properties.
- Mechanisms for monitoring adherence to legitimate agreements reached between agents engaged in argument, along with appropriate enforcement, punishment and compensation procedures. These mechanisms have clear links to the norms discussed in Section 5.
- Third-party escrow accounts, which may hold money or other items in transit between negotiating agents. For example, an agent A who contracts to purchase a service from agent B may not pay directly, but, with B's agreement, may put the money in an escrow account to be held until the service is performed, whereupon the money is transferred to B.

- Mechanisms for third-party certification, as happens with professional groups (such as doctors) and artisans in many countries.
- Record-keeping, for example, acting as an independent register of all legitimate agreements made. For some types of interactions, a copy of the discussions leading to agreements may also have to be stored; this is the case, for example, in safety-critical domains such as medicine or the design of aircraft.
- Mechanisms for mediation and intervention in agent interactions, e.g. to suggest potential compromise agreements to agents engaged in negotiation or conflict.

There has been research on some of these topics. For example, Oren and colleagues [35] have considered policing and enforcement mechanisms for multi-agent systems. Miller and McBurney [28,29] have considered efficient means of storage and retrieval of protocols stored in libraries. And, considerable work in argumentation has looked at intelligent mediation between conflicting agents, for example [17, 23, 33],

In the reverse direction, argumentation and negotiation activities can usefully be applied to the design and selection of various features of the Environment. For example, participants may negotiate a shared ontology, as in [3, 38], for use in subsequent interactions. Similarly, participants may argue over the choice of a protocol, prior to commencing an interaction using the agreed protocol [20]. Clearly, there are close links between these issues and the topics of semantics, norms, and trust.

Implicit in the study of argumentation and negotiation by agreement technologists is a belief that these approaches should enhance the likelihood of agreements being reached, or of deals being struck, between the different participants. There has only been limited research on this question, but what has been done supports this implicit assumption — for instance, the work of Karunatillake [21] demonstrates the beneficial impacts of argumentation in reaching agreements over conflicting uses of resources. Much research remains to be done to understand better and to quantify these benefits.

8 Environment for Trust

On the contrary of other Agreement Technologies like negotiation or argumentation that are directly linked to the idea of agreement, trust and reputation are not agreement generator mechanisms. However, without trust and reputation a great amount of agreements would not be possible or rather difficult. Any agreement process has associated certain level of uncertainty and trust and reputation mechanisms help the individual to deal with (at least part of) that uncertainty. For example, in an argumentation process some of the arguments can be based on information that cannot be confirmed and that its acceptance depends only on the reliability of the sender. The reputation of that sender can make up for that lack of direct knowledge about the arguments, becoming the key to finally reach the agreement.

The relation between trust and reputation as Agreement Technologies and the environment can be analysed from two perspectives. First looking at the environment as a *support* for trust and reputation mechanisms (and in general as a support for any Agreement Technology) and second looking at the trust and reputation systems as

modifiers of the environment (again, this role of environment modifier is not exclusive of trust and reputation but can be applied also to other Agreement Technology).

In general for the Agreement Technologies, and in particular for trust and reputation, the environment is the substratum that supports them.

One of the main sources to build trust on others is the result of direct interactions. These interactions are the consequence of a necessity from one of the agents (the trustor) that "trusts" the other (the trustee) to perform an action. After agreeing on the commitment, the environment will become the place where the action will be executed and more importantly, where the result of the action will be made evident. The observation of the performance and its subsequent evaluation is one of the most important inputs used by the trustor to modify the trust on the trustee. Therefore the environment, on top of being the place where the trustee performs the action, is also the *channel* that allows the trustor to perceive the performance of the trustee.

In the case of reputation something similar happens. We can define reputation as "what a social entity says about a target regarding his/her behaviour". The definition states the fact that reputation cannot exist without communication. Reputation is based on information circulating among individuals. Therefore, we can see how the environment is acting again as a support by allowing this communication.

In both cases it is clear that the environment can be an obstacle or can facilitate the functioning of trust and reputation mechanisms (and Agreement Technologies in general). An environment that allows an objective observation of the individuals' behaviour and facilitates communication is favouring and making more effective the use of trust and reputation mechanisms (and the other way around).

The environment is the support element of trust and reputation but at the same time, trust and reputation mechanisms modify the environment. Trust and reputation have a dual nature: as local mechanisms used by individuals to select their partners and, from a societal perspective, as social instruments for social order. As mechanisms for social order, their task is to influence the behaviour so to enforce some kind of conduct in detriment of another. Social relations among individuals are part of the environment. Reputation as one of the sources for trust, and trust as a mechanism that defines in a big extend the relations among individuals, have an important influence in the intensity, creation and destruction of these social relations. Apart from this direct influence of trust and reputation in the environment, more than often societies also modify the environment to improve trust among their members and facilitate reputation mechanisms to detect wrongdoers (for instance by adding communication mechanisms to allow reputation spread, by creating physical and social spaces to facilitate trust relations among companies, etc.).

9 Discussion and Concluding Remarks

Given the characterization of agent environment done in Section 2 and the analysis of its role with respect to the different dimensions of Agreement Technologies in the other sections, the benefits of agent environment for Agent-Based Agreement Technologies can be broadly framed at two different (but related) levels:

- *conceptual level* where the use of environment improves the modeling and design of the strategies and mechanisms that allow for achieving agreements, conceiving solutions that don't rely necessarily only on agents and message passing as unique abstractions,
- *practical level* where the use of agent environment improves the separation of concerns, the modularity, reusability, openness and extensibility in designing and implementing agent-based agreement technologies.

Let's consider for instance the semantic dimension to illustrate those points. When we consider heterogeneous agent-based agreement technologies, benefits of environment at the conceptual level and its implication at the practical level are clearly visible: the environment is an unavoidable mediating entity between the semantics which are hard-wired into the internal reasoning of the different agents (see sec. 4). The above sections show other impacts in relation with norms, organizations, argumentation and negotiation, trust and reputation.

Going a step further in the characterization of agent-environment for agent-based agreement technologies, it appears that the *interaction mediation-support layer* of the agent environment as shown in the Fig. 1 shift to an *agreement mediation layer*. This layer implements the high level constructs of a *coordination metamodel* which is involved in the building of agreements. Moreover, this layer ensures also that the actual specification of a particular instance of the coordination conventions is properly implemented and the corresponding conventions dully enforced [9]. It may thus be seen and understood as the "institutional layer" as defined in [9]) in the sense that:

1. a virtual agreement space—or "institutional", in the sense of Searle [42])— where only certain messages are deemed acceptable and thus processed, is created (at run-time) and
2. that the admission of messages and their subsequent processing in the environment comply with those conventions that have been specified.

To achieve this double purpose of translating the specification into a run-time agreement space and enforcing the coordination conventions within that agreement space, that layer needs to include appropriate data structures, operations, and governance functionalities. The data structures mirror the conceptual coordination devices and the operations apply to those data structures Governance functionalities ensure that the actual interaction flow complies with the coordination conventions that have been specified for the particular socio-technical system. Thus, in particular, the layer would need to mirror the coordination metamodel properly by providing support to basic services such as:

- Time keeping, and consequently control of time-outs, synchronization and interaction clock (if part of the metamodel)
- Interface with the external environment (depending on the metamodel: services, activation platform, other organizations)
- Primitive means of decomposition of complex activities (e.g., scenes and transitions in the case of electronic institutions)

It also has to provide support to elaborated services in relation to agreement technologies such as:

- Operations that implement interactions between agents and environment: enter or leaving the agreement space, moving form one subactivity to another, performing an atomic action, ...
- Operations that the environment needs to perform in order to keep activity flowing in the agreement space: create and terminate the agreement space and its subactivities, controlling the changes of roles attempted by agents, managing the movements of agents between activities and most importantly, guaranteeing that atomic actions comply with stated norms, ...
- Data structures needed to keep track of agent activity, namely, the state of the local and global contexts as interactions proceed
- Representation and handling of atomic interactions among agents, handling of interaction flow

Besides these basic and common features, the paper has shown that elaborated services in relation to agreement technologies need to be considered and added in that layer. This raises different challenges and multiple and interesting research directions for supporting and developping Agreement Technologies. For instance, considering semantics, it is required elaborated services to:

- represent environmental aspects at different levels of support: either using a specific representation framework, or using any combination of formal and informal representation means, agents need to explicitly represent knowledge about all the environment aspects,
- combine knowledge in large-scale open settings & reconcile subjective views: if we need to deal with open and large scale settings, then establishing common and coherent views of the shared environment is unavoidable for agents to coordinate and collaborate effectively. However, establishing semantic agreements between any pair of distant agents in large and open communities of agents is a challenge. The environment dimension has not been thoroughly investigated towards reaching this end,
- help in the learning the semantics of everything, out of cases: being able to represent everything in the environment and understanding the representations used, is one thing. Inspecting and exploiting the interactions of others with the environment (within specific contexts of interaction) agents must be able to learn the meaning of representations used.
- support computing emergent, commonly agreed languages (symbols, syntax and semantics) for interaction. In open and large-scale settings, agents may use any formal or informal representation language. Computing commonly agreed representation means, is a great challenge.

Considering the norms, there are important needs of defining and representing in a standard way the events and actions that happen in an environment, as well as the context of the interactions in terms of properties of resources and their value in order to be used in the definition of norms that should regulate those events and actions. There is also the need of elaborated services to add in this agreement mediation layer such as:

- general mechanisms for contextualizing abstract norms defined at design-time into norms situated in specific spaces,

– extendable functionalities/services provided by the environment for adding those required for norms management.

Moving to the organization, elaborated services, situated in the agent environment and accessible to the agents, are needed to:

– provide facilities to enter or exit a given organization to allow run-time recruitment of new members as well as voluntary desertion and/or expulsion of members,
– support on-demand creation, deletion and modification of organizations,
– give support to the institutional components of an organization, i.e. norms, powers, agreements.

Considering the argumentation dimension, besides scaling up the existing work, which typically considers single interactions between two or a small number of agents, elaborated services have to be considered to complement the basic services dedicated to interaction management. These elaborated services will provide the possibility to manage libraries and database of ontologies, protocols and agreements. Finally, once abstractions such as ontologies and interaction protocols are stored and accessible in the environment, participating agents in a system then need the ability to reason about these abstractions and to invoke them as required. Some work in this direction has been undertaken, e.g. [30], but much remains to be done.

Finally considering Trust and Reputation, challenges and elaborated services need to be considered in order to exploit the existing strong link between trust and reputation and the environment in electronic environments. Computational trust and reputation models [36] till recently have been designed as isolated components, barely linked to the context and the rest of the agent elements. Therefore they have little capacity to change the environment to facilitate its functioning as described before. One improvement in this sense would be giving the trust and reputation system the capacity to influence the actions of the agent to modify the environment so it becomes more trust and reputation "friendly". For example the trust and reputation mechanism could suggest actions to the agent in order to improve the communication capabilities the environment has or suggest the formation of new social relations to enforce already present trust relationships. So instead of relying on the possibilities the environment offers "for free" to facilitate trust and reputation, the agent (by means of its trust and reputation mechanism) should be proactive and modify the environment to make more robust and reliable the two social artifacts.

Acknowledgements

This paper is the synthesis of the discussions from the workshop and panel about Environment in Agreement Technologies, held during the COST Action IC0801 Agreement Technologies meeting June 1 and 2, 2012, organized by O. Boissier, C. Carrascosa, A. Ricci, M. Schumacher. We would like to thank the COST Action IC0801 for the support.

References

1. J. L. Arcos, P. Noriega, J. A. Rodriguez-Aguilar, and C. Sierra. E4mas through electronic institutions. In D. Weyns, H. Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems III.*, number 4389 in Lecture Notes in Computer Science, pages 184–202. Springer, Berlin / Heidelberg, 08/05/2006 2007.
2. S. Bandini, S. Manzoni, and G. Vizzari. Agent based modeling and simulation: An informatics perspective. *J. Artificial Societies and Social Simulation*, 12(4), 2009.
3. R.-J. Beun and R. Eijk. A co-operative dialogue game for resolving ontological discrepancies. In F. Dignum, editor, *Advances in Agent Communication*, LNAI 2922, pages 349–363. Springer, Berlin, Germany, 2004.
4. S. Bromuri and K. Stathis. Situating Cognitive Agents in GOLEM. In D. Weyns, S. Brueckner, and Y. Demazeau, editors, *Engineering Environment-Mediated Multi-Agent Systems*, volume 5049 of *LNCS*, pages 115–134. Springer Berlin / Heidelberg, 2008.
5. S. Bromuri and K. Stathis. Distributed agent environments in the ambient event calculus. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 12:1–12:12, New York, NY, USA, 2009. ACM.
6. V. T. da Silva, R. Choren, and C. D. Lucena. MAS-ML: A Multi-Agent System Modeling Language. *International Journal of Agent-Oriented Software Engineering*, 2(4):381–421, 2008.
7. K. da Silva Figueiredo, V. Torres da Silva, and C. de Oliveira Braga. Modeling norms in multi-agent systems with normml. In M. De Vos, N. Fornara, J. Pitt, and G. Vouros, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, volume 6541 of *Lecture Notes in Computer Science*, pages 39–57. Springer Berlin / Heidelberg, 2011.
8. V. Dignum. *Handbook of Research on Multi-Agent Systems : Semantics and Dynamics of Organizational Models*, chapter The Role of Organization in Agent Systems, pages 1–16. Information Science Reference Publisher, 2009. ISBN 978-1-60566-256-5, DOI 10.4018/978-1-60566-256-5.ch001.
9. M. d'Inverno, M. Luck, P. Noriega, J. A. Rodriguez-Aguilar, and C. Sierra. Communicating open systems. *Artificial Intelligence*, 186(0):38 – 94, 2012.
10. C. T. dos Santos, P. Quaresma, R. Vieira, and A. Isaac. Comparing argumentation frameworks for composite ontology matching. In P. McBurney, I. Rahwan, S. Parsons, and N. Maudet, editors, *ArgMAS*, volume 6057 of *Lecture Notes in Computer Science*, pages 305–320. Springer, 2009.
11. S. Esparcia, E. Argente, R. Centeno, and R. Hermoso. Enhancing MAS Environments with Organizational Mechanisms. *International Journal on Artificial Intelligence Tools (IJAIT)*, 20(4):663–690, 2011.
12. S. Esparcia, R. Centeno, R. Hermoso, and E. Argente. Artifacting and Regulating the Environment of a Virtual Organization. In *23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2011)*. IEEE, 2011.
13. J. Ferber, F. Michel, and J. Baez. AGRE: Integrating environments with organizations. In *Environments for Multi-agent Systems*, volume 3374 of *LNCS*, pages 48–56. Springer, 2005.
14. N. Fornara and M. Colombetti. Specifying and enforcing norms in artificial institutions. In M. Baldoni, T. Son, M. van Riemsdijk, and M. Winikoff, editors, *Declarative Agent Languages and Technologies VI*, volume 5397 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg, 2009.
15. N. Fornara and C. Tampitsikas. Using OWL Artificial Institutions for dynamically creating Open Spaces of Interaction. In *Proceedings of the AT 2012 First International Conference on Agreement Technologies, October 15 - 16, 2012 in Dubrovnik, Croatia.*, page to appear, 2012.
16. A. Garca-Camino, P. Noriega, and J. A. Rodrguez-Aguilar. Implementing norms in electronic institutions. In S. T. Michal Pechoucek, Donald Steiner, editor, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*, pages 667–673, Utrecht, NL, 2005. ACM Press.
17. T. F. Gordon and N. Karacapilidis. The Zeno argumentation framework. In *Proc. Sixth International Conference on AI and Law*, pages 10–18, New York, NY, USA, 1997. ACM Press.
18. B. J. Hodge, W. Anthony, and L. Gales. *Organization Theory: A Strategic Approach*. Prentice Hall, 2002.
19. J. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20:369–400, 2010.
20. J. Hulstijn, M. Dastani, and L. van der Torre. Negotiation protocols and dialogue games. In *Proc. Belgian-Dutch AI Conference (BNAIC-2000)*, Kaatsheuvel, 2000.

21. N. C. Karunatillake. *Argumentation-Based Negotiation in a Social Context*. Phd, School of Electronics and Computer Science, University of Southampton, Southampton, UK, 2006.
22. A. Koster, W. M. Schorlemmer, and J. Sabater-Mir. Engineering trust alignment: Theory, method and experimentation. *Int. J. Hum.-Comput. Stud.*, 70(6):450–473, 2012.
23. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreement through argumentation: a logical model and implementation. *Artificial Intelligence*, 104:1–69, 1998.
24. R. Krummenacher, E. P. B. Simperl, D. Cerizza, E. D. Valle, L. J. B. Nixon, and D. Foxvog. Enabling the European patient summary through triplespaces. *Computer Methods and Programs in Biomedicine*, 95(2-S1):33–43, 2009.
25. R. Kullerkupp. Analysis and preliminary design of a social networking system for supporting information exchange about medicines, 2012.
26. M. P. Locatelli and G. Vizzari. Awareness in collaborative ubiquitous environments: The multilayered multi-agent situated system approach. *TAAS*, 2(4), 2007.
27. P. McBurney and S. Parsons. Dialogue games for agent argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 13, pages 261–280. Springer, Berlin, Germany, 2009.
28. T. Miller and P. McBurney. Annotation and matching of first-class agent interaction protocols. In L. Padgham, D. Parkes, J. P. Mueller, and S. Parsons, editors, *Proc. Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*, New York, NY, USA, 2008. ACM Press.
29. T. Miller and P. McBurney. Characterising and matching iterative and recursive agent interaction protocols. In W. van der Hoek, G. Kaminka, M. Luck, and S. Sen, editors, *Proc. Ninth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010)*, pages 1207–1214, Toronto, Canada, 2010. IFAAMAS, ACM Press.
30. T. Miller and P. McBurney. Propositional dynamic logic for reasoning about first-class agent interaction protocols. *Computational Intelligence*, 27(3):422–457, August 2011.
31. J. Odell, H. V. D. Parunak, M. Fleischer, and S. Brueckner. Modeling agents and their environment. In *Agent-Oriented Software Engineering III*, volume 2585 of *LNCS*, pages 16–31. Springer Berlin / Heidelberg, 2003.
32. F. Y. Okuyama, R. H. Bordini, and A. C. da Rocha Costa. A distributed normative infrastructure for situated multi-agent organisations. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, AAMAS '08, pages 1501–1504, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
33. E. Oliva, P. McBurney, and A. Omicini. Co-argumentation artifact for agent societies. In I. Rahwan, C. Reed, and S. Parsons, editors, *Proc. Fourth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2007)*, pages 115–130, AAMAS 2007, Honolulu, Hawai'i, USA, 2007.
34. A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, Dec. 2008.
35. N. Oren, T. Norman, A. Preece, and S. Chalmers. Policing virtual organizations. In C. Ghidini, P. Giorgini, and W. Hoek, editors, *Proc. Second European Conference on Multi-Agent Systems (EUMAS 2004)*, pages 499–508, Barcelona, Spain, 2004. EUMAS.
36. I. Pinyol and J. Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, pages 1–25, 2012.
37. M. Piunti, A. Ricci, O. Boissier, and J. F. Hübner. Embodied Organisations in MAS Environments. In *Proceedings of the 7th German Conference on Multiagent System Technologies MATES'09*, volume 5774 of *LNAI*, pages 115–127. Springer-Verlag, 2009.
38. C. Reed, T. J. Norman, and N. R. Jennings. Negotiating the semantics of agent communications languages. *Computational Intelligence*, 18(2):229–252, 2002.
39. A. Ricci, M. Piunti, and M. Viroli. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, 23(2):158–192, Sept. 2011.
40. A. Ricci, M. Piunti, M. Viroli, and A. Omicini. Environment programming in cartago. In R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah-Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms and Applications, Vol. 2*, pages 259–288. Springer Berlin / Heidelberg, 2009.
41. J.-J. Sarrasin, M. Schumacher, C. Hay, and P. Richard. Health-identity : mobile services for consumers of medicines. In *EFMI Special Topic Conference "Seamless care - safe care : The challenges of interoperability"*, June 2010.
42. J. R. Searle. *The construction of social reality*. Free Press, 1995.
43. M. Sensoy, T. J. Norman, W. W. Vasconcelos, and K. P. Sycara. Owl-polar: A framework for semantic policy representation and reasoning. *J. Web Sem.*, 12:148–160, 2012.

44. L. Sterling and K. Taveter. *The Art of Agent-Oriented Modeling*. Massachusetts: The MIT Press, 2009.
45. C. Tampitsikas, S. Bromuri, N. Fornara, and M. I. Schumacher. Interdependent Artificial Institutions In Agent Environments. *Applied Artificial Intelligence*, 26(4):398–427, 2012.
46. N. Tinnemeier, M. Dastani, and J.-J. Meyer. Programming norm change. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, AAMAS '10, pages 957–964, Richland, SC, 2010. ACM.
47. J. Wagner and J. Hollenbeck. *Organizational Behavior*. South-Western Pub., 2001.
48. D. Weyns and T. Holvoet. Formal model for situated multiagent systems. *Fundamenta Informaticae*, 63(2–3):125–158, 2004.
49. D. Weyns, A. Omicini, and J. J. Odell. Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, Feb. 2007.
50. D. Weyns and H. V. D. Parunak, editors. *Special Issue on Environments for Multi-Agent Systems*, volume 14 (1) of *Autonomous Agents and Multi-Agent Systems*. Springer Netherlands, 2007.
51. D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems: State-of-the-art and research challenges. In D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber, editors, *Environment for Multi-Agent Systems*, volume 3374, pages 1–47. Springer Berlin / Heidelberg, 2005.
52. D. Weyns, K. Schelfthout, T. Holvoet, and T. Lefever. Decentralized control of E'GV transportation systems. In M. Pechoucek, D. Steiner, and S. G. Thompson, editors, *AAMAS Industrial Applications*, pages 67–74. ACM, 2005.