Ballroom etiquette: a case study for norm-governed multi-agent systems

Dorian Gaertner*, Keith Clark, and Marek Sergot

Imperial College London, SW7 2AZ, United Kingdom

Abstract. We present a case study which describes a ballroom as a social institution with autonomous dancer agents together with sets of norms and conventions that coordinate the behaviour of the participants. We provide a representation for the interaction protocols as finite state machines and a new way of formalising the associated norms in a logic programming language. Furthermore, we report on recent and ongoing work on an architecture for normative systems of this kind which allows agents to dynamically download interaction protocols and operational norms to guide their behaviour. Finally, we outline an alternative approach for representing the *institutional state* in a virtual, distributed fashion in the agents' private belief stores.

1 Introduction

We present the main elements of a norm-governed multi-agent system which simulates a ballroom for social dancing. It is intended to provide a case study for exploring the specification and implementation of a wide class of norm-governed multi-agent systems generally. Agents in the ballroom form commitments by negotiation according to specified protocols and conventions. The fulfillment of these commitments, and other aspects of the agents' interactions, are further guided and constrained by norms of conduct expressing what behaviours are socially acceptable (or 'legal'). These protocols and norms together constitute the ballroom *etiquette*[1]. Our assumption is that all ballroom etiquette has general common features, and a common ontology in terms of which the norms can be expressed, though the details will vary from one specific ballroom to another. Our aim is to provide an implementation in which agents joining a computational society (here, a ballroom) are provided with an executable representation of the applicable norms which they use in their internal decision-making procedures.

Clark and McCabe [8] have used the ballroom example to demonstrate features of the agent programming language `Go!`, and its support for multi-threaded agents with inter-agent communication and coordination via messages. Although the agents are quite simple, this scenario encompasses key behavioural features

---

* The first author undertook part of this work while at the Artificial Intelligence Research Institute (IIIA) in Bellaterra, Spain.

[1] *Etiquette:* the customs or rules governing behaviour regarded as correct or acceptable in social or official life. [Collins dictionary]

of agents: autonomy, adaptability and responsibility. However, in that implementation the norms governing the dancers' interactions — the protocols used to negotiate commitments, and the conventions that govern how commitments are fulfilled — are implicit in the code. We have constructed a version in which these norms are *explicit*, providing a case study for specifying and implementing a multi-agent system (MAS) in which agents take account of and deliberate about the norms that regulate their behaviour. Like Artikis et al. [2, 4, 3], Ossowski [16], and many others, we accept that organisational and legal elements of open agent systems, the semantics of agent communication, and social and normative relations generally, must be externally visible and not embedded in the internal state of individual member agents. However, we also want to address how individual agents can take into account the existence of norms in their internal decision making procedures. While this aspect has been analysed thoroughly for individual normative agents by Boella and Lesmo [5] and Castelfranchi et al. [6], the institutional view of norms for MAS has received comparatively little attention (but see e.g. [14, 17]).

In the present version we assume that agents always fulfil their commitments and comply with any other applicable norms, as in the original implementation by Clark and McCabe. In future versions we will remove this assumption, so as to explore mechanisms for enforcing and encouraging norm compliance, sanctioning and other reparational procedures, and the associated auditing infrastructure. These topics will not be covered in this paper.

In the next section, we describe the ballroom simulation in more detail. We provide examples of the protocols and norms, their representation, and requirements for their implementation in Sections 3 and 4. In Section 5 we outline an agent architecture in which agents can download interaction protocols together with the applicable norms and conventions expressed in a common ontology when they join the ballroom. Section 6 then introduces a way of modelling the institutional state in a virtual fashion, in which each agent maintains its own representation of the relevant state of the institution as part of its internal beliefs. Section 7 presents related work and Section 8 concludes.

2   The simulation

As a springboard for this research we are using a ballroom simulation devised by Clark and McCabe [7, 8] that consists of a dance floor and a bar area. In this simulation, male and female agents participate in social dancing and negotiate over which partner to dance with for the next dance of a certain kind. The agents are multi-threaded in that their reactive and deliberative behaviours are executed concurrently. Agents communicate and coordinate their behaviour using their beliefs, intentions and desires which are modelled using dynamic memory stores. The band, represented by an additional agent, plays one of six different types of dance deciding randomly which dance to play next. Once a negotiation between two dancers concludes in an agreement between them to dance the next polka, say, both dancers are committed to indeed perform the next polka together.

Dancer agents can arrive and leave the dance hall at any time — while the band is playing or during the negotiation phase. When a new dancer agent arrives, it registers with the directory server (which is a facilitator used by the dancer agents to discover potential partners) its gender and a list of its desires. These include (amongst others) the desire to dance n_1 dances of type X_1, n_2 dances of type X_2, and so on, and to go to the bar when dances of type X_3 are played. The directory server then announces the arrival of a new dancer agent to the others and informs the new dancer about what were the *initial* desires of the other agents that are present. This information might not coincide with the current desires of the other dancers due to the fact that it is the dancers themselves, rather than the directory server, which keep track of (their beliefs about) other dancers' changing desires. For example, while agent Bob may have been informed that agent Alice initially desired to dance two tangos, he is generally unaware that she may already have fulfilled these desires (unless it was he himself who has danced the tango with her twice, in which case he will have updated his beliefs accordingly). The prototype application is written in `Go!` [7], a multi-paradigm programming language for agent applications. `Go!` has logic programming features such as relations, is higher-order in a functional sense and also has action procedure definitions for imperative programming [8]. It is multi-threaded and uses asynchronous message-passing for inter-agent communication. Threads within the same agent can also communicate using Prolog-style dynamic relations as common memory.

The simulation cycles through two phases. In *phase one*, the band plays a tune of type X (randomly chosen from the set of dance types). In the `Go!` implementation, the band announces that it is about to play a dance of type X by broadcasting a start message. Dancers who have a joint commitment to dance together when this type of dance is played, then exchange messages as part of a hand-shake protocol and begin to dance. Dancers who have a joint commitment to go to the bar together when this type of dance is played, exchange messages and go to the bar. The end of the dance is signalled when the band announces (again by way of message broadcast) that it has stopped playing.

In *phase two* which takes place between dances, the dancer agents negotiate to form commitments to dance the next time the band plays a dance tune of type X or to go to the bar area the next time a dance tune of type Y is played. At the next dance interval the beliefs, desires and intentions of the dancer will almost certainly have changed. Even if they are the same, a re-negotiation with the same female may now have a different outcome because of changes in her mental state. The negotiations in phase two follow fixed protocols, the details of which will vary from one dance hall to another and will be looked at in the next section.

3 Negotiation protocols

In order to coordinate what to dance, when, and with whom, agents need to negotiate. In the simplest form, this involves one agent inviting another agent to dance, who then agrees. In general however, negotiations can be much more complex and have to follow certain protocols which define valid interactions.

Example variations One could restrict the ability to negotiate to times when the band is not playing. This is not necessarily a fixed feature of all dance halls. In some dance halls one may want to allow negotiations to take place at any time, or perhaps anywhere when the band is not playing but only in the bar when it is playing.

Another characteristic of a negotiation is the initiator. In the original version of the ballroom the female never took the initiative. A female dancer had to wait for an initial proposal from a male and thereafter could accept, reject, or make counter proposals about a different dance or to go to the bar, depending on her current desires and commitments. We now support a wider range of negotiation protocols, according to the type of dance hall. In a *ladies' choice discotheque*, for example, the female dancer can and should take the initiative. This is easily implemented by changing the protocol. Other possible variations involve disallowing counter proposals, or requiring a dancer to accept any proposal to dance, irrespective of its desires, as long as it does not conflict with existing commitments. One could also imagine a ballroom, in which dancers must negotiate via some intermediary. As mentioned before, a dancer agent can participate in many simultaneous negotiations by spawning auxiliary negotiation threads, subject to the 'good faith' principle outlined below. We may wish to impose further restrictions, so that agents can negotiate with only one other agent at a time, for example.

Commitments We assume that agents do not dance/go to the bar unless they have a commitment to do so. Furthermore, in the present version of the system, dancers always fulfil their commitments, i.e. they will always dance/go to the bar, if they have a commitment to do so. From this it follows (in a sense we leave informal in this paper), that dancer agents never make commitments concerning the next dance of type X with more than one partner at a time. It also follows that each agent has at most n dance commitments at any given time, where n is the number of different dance types.

Good faith principle Dancers can participate in many negotiations concurrently, but only in accordance with a *principle of good faith* whereby (a) they never initiate, accept or counter-propose a new negotiation concerning the next dance of type X when already participating in a negotiation about dance type X with another agent, and (b) they never propose or accept a commitment concerning dance type X when they already have a commitment regarding dance

type X with another agent. There are several assumptions about the nature of the negotiation protocol that are implicit in this good faith principle: one aim of the case study is to make them explicit.

Representation Negotiations follow a particular protocol which is specified by the system designer when formalising the interaction of agents. These protocols are represented as *Finite-State-Machines* and instantiated by the agents when required. One example protocol and the simplified messages passed between the negotiation participants is given in Figure 1.

Note that `Dance`, `Dance2` and `Dance3` are variables instantiated to distinct dance types and that in this example, only female dancers can suggest drinking at the bar. After the negotiations are finished and the band has started to play a certain song (a polka, say) *phase two* is executed (as explained in Section 2). Agents who have a joint commitment to dance the next polka enter another communication. There, another protocol regulates the interaction between male and female dancers similar to the notion of a hand-shake. The male reminds the female of their mutual commitment and the female acknowledges him and the dance begins. If their agreement was to go drinking at the bar when the next polka is played, then their interaction will vary accordingly. Once the band stops playing, the dancing or drinking stops and another round of negotiations begins. Finally, the band announces the end of the evening and agents follow a protocol that governs how they bid farewell to each other.
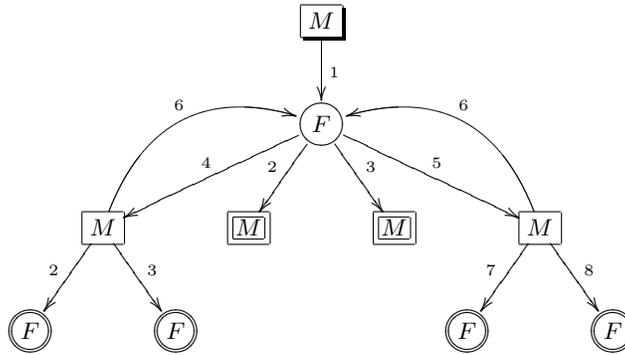

Fig. 1. Simplified example protocol which allows agents to counter-propose

(1) `willyouDance(Dance)`	(5) `barWhen(Dance2)`
(2) `okDance()`	(6) `willyouDance(Dance3)`
(3) `sorryDance()`	(7) `okBar()`
(4) `willyouDance(Dance2)`	(8) `sorryBar()`

To summarise, there are three stages in the simulation that are controlled by some protocol (negotiation, hand-shake and farewell) but these protocols are not hard-wired into the agents' code. Instead they are downloaded dynamically, in a sense that will become clear in Section 5.

4   Norms

For the purposes of this paper, we will follow a commonly accepted usage and classify norms broadly into two general categories: *constitutive norms* and *behavioural norms* (or norms of conduct).

Constitutive norms specify (1) the conventional-institutional meaning of messages and other communicative acts within a given institution (that a message 'Will you dance the tango?' is an expression of an 'offer to dance', that a message 'Yes' is an expression of an 'acceptance to dance' depending on context, and so on), (2) the protocols and procedures which define what kinds of acts are meaningful or 'valid' according to context (that an 'acceptance' following a (valid) 'offer' forms a 'commitment', that an 'offer' by one party can be followed by a 'counter-offer' by the other party in a negotiation, that the parties in a negotiation must take turns in exchanging messages, and so on), and (3) other more general forms of what are sometimes called 'qualification norms' which define how institutional facts, such as 'band is playing', are determined from observable facts ('brute facts' in Searle's terminology) such as a 'started playing' message having been broadcasted by the band agent and no 'stopped playing' message having been broadcasted in the meantime. A wide variety of formalisms for expressing constitutive norms have been reported in the literature. We discuss the choices we have made in our current implementation in the following section.

Behavioural norms specify what actions are permitted and obligatory. They may be further classified according to whether we take a "bird's eye" perspective from the system designer's point of view or whether we take a genuinely agent-centric perspective. We will not elaborate further on that distinction here. As already mentioned, in the present version of our system we assume for simplicity that all agents comply with the applicable behavioural norms: agents do not perform actions that are not permitted, and always perform actions that are obligatory. We plan to remove this simplification in later versions.

Although apparently simple, the ballroom example has a very rich and varied set of possible norms, of both kinds. Some are straightforwardly constitutive, some are clearly behavioural, and some, in their natural language formulation, can be interpreted in different ways. We list here some examples with a brief discussion in each case.

Some constitutive norms:

– Dance partner (and negotiation partner) must be of opposite sex — this may not apply in certain dance halls. It may be that a male agent sends a message to another male agent offering to dance but that message is not a valid offer according to the constitutive norms in force in the ballroom. We might want to add a further behavioural norm to say that it is not *permitted* for a male dancer to send a message that offers to dance to another male agent, but that is a *separate* level of specification.
– A female must wait to be approached by a male dancer: only a male agent can (has the institutional *power* to) initiate a negotiation to dance — this can also vary according to the dance hall. Again, it is possible that a female agent sends a message attempting to initiate a negotiation but this is not a meaningful message according to the constitutive rules in force. As a separate level of specification, we might add another norm to the effect that females are not permitted to send such messages.
– Negotiations only take place when the band is not playing. In other dance halls, negotiations can take place at any time, or perhaps only at the bar when the band is playing. Messages offering to dance can be sent while the band is playing but they are not offers to dance according to the constitutive norms. It is a separate question whether the dance hall permits such messages to be sent.

Some behavioural norms:

– An agent must fulfill all its commitments to dance/go to the bar (if it has a chance to do so).
– An agent should not leave the dance hall while it has unfulfilled commitments. In future versions we will allow agents to negotiate about the release from commitments but we do not support that refinement yet.
– Always accept a request for a dance if it does not conflict with existing commitments. (This does not apply in all dance halls.) Here there is an obligation on an agent to exercise its powers, as defined in the constitutive norms, in a particular way.
– Do not dance more than three consecutive dances with the same partner. (In the present version, agents do not maintain an explicit history of all previous dances but this can be easily added by extending the belief state of an agent.)
– At a wedding dance, all male agents must dance at least once with their mother-in-law. (In the present version, agents negotiate only about the next dance of a particular type. In future versions we want to introduce an element of planning.)

In the present version, agents always fulfill their commitments, and comply with all obligations. In future versions, we plan to distinguish between *deliberate* violation of an obligation (such as when an agent chooses not to fulfill its commitment to dance) and *practically unavoidable* violations (such as when an agent is prevented from fulfilling its commitment to dance, for example because of physical restrictions on the size of the dance floor, inability to complete the required hand-shaking protocol in the time available, and so on). This is future work.

Some norms, in their natural language formulation, can be interpreted either as constitutive norms, or as behavioural norms, or as a combination of the two. For example: the principle of 'good faith' introduced in Section 3 requires that an agent X does not offer to dance a tango with Y when it already has a commitment to dance a tango, or is already negotiating to dance a tango, with another agent Y'. Should this be represented as a constitutive norm or as a behavioural norm? If we view it as a constitutive norm, we are saying that a message from X to Y offering to dance a tango is not a valid offer when X has a (potentially) conflicting commitment to dance with Y'. Depending on what other norms are in force, the recipient Y may have obligations to accept offers from X, or to send counter-proposals, or rejections, according to its other commitments. But how is the recipient Y to determine whether X's offer is valid? In order to do that, it would need to know what other commitments X has made, and even what other negotiations X is currently engaged in. This is clearly impractical, unless we have some kind of central server which records all messages exchanged and to which agents can refer to determine what is currently valid and what is not. This is something we want to avoid. The alternative is to say that offers from X are valid offers (as long as they are correctly formed) whether or not X is currently committed to dancing the tango or is engaged in negotiations to do so. Instead, we say that there is a behavioural norm which forbids X from sending (valid) offers to dance the tango in these circumstances. Y can proceed in its decision-making without access to X's other commitments and negotiations; the obligation is on X to ensure that the 'good faith' principle is complied with.

5   Architecture

In this section, we will outline an agent architecture that supports normative agents of a heterogeneous kind. While we assume that, in order to participate in the ballroom, dancers have an understanding of concepts from an underlying ontology (as outlined in [9]), the agents are not required to have the norms hardwired into their code. Instead, they *download* the interaction protocols together with explicit norms and conventions when they join the ballroom. The choice of ontology language is not important in our present discussion.

In order to understand the protocol annotation, agents need to know the semantics of *role names* and the institutional meaning of messages sent and received. The interpretation of a message 'Will you dance the tango?' as expressing an offer to dance, and the specific conditions under which it is a valid offer to

dance, are part of the (constitutive) norms of the ballroom etiquette. In this paper we will refer to the institutional meaning of a message as its *purpose*. We are aware that this term is rather overloaded, but it seems natural and intuitive in the present context. Since each transition in the protocol graph corresponds to the sending of a message, we annotate the protocol arc with the corresponding 'purpose'. Furthermore, not every agent can use every transition, so role labels are used to restrict certain transitions.

When an agent decides, depending on its current goal, to invoke a communication with another agent, the protocol tells it what messages to send and subject to what norms. It can only send the message if it has the correct role and fulfils all the norms associated with the transition in question. These norms are expressed using an ontology of relations and the interface to this ontology needs to be understood by each dancer.
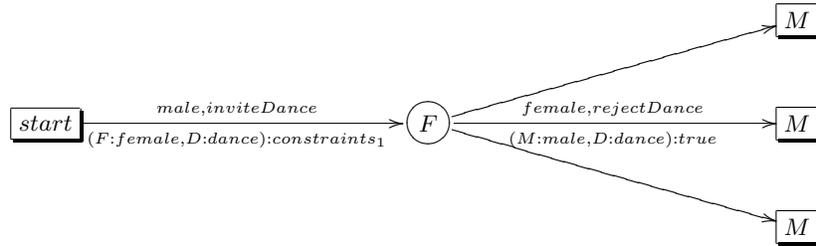

Fig. 2. Part of a proposal protocol with annotated arcs

An example could be a norm that limits the number of dances a dancer can dance with the same partner to six. It uses the predicate `dancesPerPartner(P,N)`, which has to be implemented by each agent and provides access to its mental state. This particular predicate unifies `N` with the number of dances that the agent has already danced with partner `P`.

```
canStillDanceWith(P) :-
        dancesPerPartner(P, N),
        N < 6.
```

The norm `canStillDanceWith(P)` is not violated, if this number is less than six. In a different ballroom, this number may vary or indeed the norm may not be present at all.

The predicate `dancesPerPartner` is an example of the relations we termed *introspective predicates*. These are part of the current mental state of the agent. Each participating agent is required to provide definitions for each of these predicates. Another example predicate is `bandResting`. In the absence of a central authority that logs when the band starts and stops to play, each agent needs to

keep track of this itself and the predicate is true, if the agent currently believes that the band is not playing.

In Figure 2, we illustrate a part of the protocol from Figure 1. The arcs between states of the protocol are annotated with a role, a purpose and a set of constraints. The introspective predicates can be used here directly to constrain a transition or they can be used in the description of an explicit norm that has been downloaded by the dancer.

For example, an agent can go from the start state to the next state if it plays the role *male* by sending a message which has the purpose *inviteDance*. The actual illocution used by the agent is provided by the protocol so that agents do not have to know them as long as they understand the *purposes*. An agent obtains this illocution by providing two things: a binding for the variable F (of type female) and a binding for the variable D (of type dance). The constraints denoted by *contraints1* in Figure 2 further restrict the applicability of this transition. They are made up of introspective predicates and some of the downloaded conventions. In the example, *constraints1* could be:

```
canStillDanceWith(P), bandResting, notCommittedNorm(D)
```

A different transition that leaves the state annotated with F can only be traversed by a *female* agent who intends to reject an offer. The female dancer needs to provide the identity of the male who approached her and the dance in question in order to form a valid rejection message (using the appropriate illocution provided by the protocol). However, there are no further constraints associated with this transition (indicated by the *true* constraint).

Summarising, we can see that the path an agent takes through the protocol graph is influenced by its desires, biases and preferences as well as the constraints on the protocol and the norms and conventions it downloaded.

6   Virtual, distributed institutional state

In common with many previous works (see e.g. [3, 12] among others) we use the term *institutional state* to refer to the set of obligations, permissions, and other institutional facts (such as 'band is playing' or 'X can send an offer') current at any given time. An implementation of a MAS then provides some mechanism to maintain and evolve the institutional state and verify actions with respect to it. In many implementations, this is done by providing a server, centralised or physically distributed, to which agents can refer to obtain authoritative information about the current institutional state (see e.g. [1, 12]). Essentially the same idea has been used in areas outside MAS, such as the 'policy servers' that are used in the management of distributed computer systems. In García-Camino et al. [12] *governor agents* are used to intercept messages from external agents and only send them on to the institution if they conform with the current institutional state.

We are using the ballroom case study to develop an alternative approach, intended to be complementary to the methods summarised above, which we believe is more appropriate in some applications of norm-governed MAS. Like Robertson in [18], we argue that centralised control is not always needed. Instead of storing the institutional state explicitly, whether centralised or distributed, every agent in the system is responsible for maintaining its own partial view of the current institutional state. This is intended to provide a much lighter and more flexible mechanism for enforcing norm compliance. Conceptually, there is an institutional state, but it is not represented explicitly (except perhaps for auditing purposes which might be desirable in some applications).

Our ballroom scenario is one example of a society which can operate without central control or norm enforcement. If an institution has sanctions, compensations and/or reparatory mechanisms, then it needs some form of auditing to objectively test its participants' claims. The more it relies on such mechanisms, the more important it is to have some central authority. Conversely, societies that have more implicit norm enforcement (via reputation or trust) are less reliant on strict, authoritarian enforcement.

Since each agent is responsible for maintaining its own beliefs about the evolving institutional state, these beliefs may differ from the actual institutional state. We make two remarks about this. (1) In order to be effective, norms (both constitutive and behavioural) must be formulated in terms of facts that an agent can observe directly or via simple communication with other agents. We have already discussed one such example in previous sections, where we argued that the recipient of a message (an offer to dance the tango, say) could not be expected to have information about the sender's other commitments or ongoing negotiations. (2) We imagine it will be possible to verify formally that an agent's implementation of the required norm interpreter will yield correct beliefs about the institutional state (correct with respect to the conceptual institutional state), on the assumption that all relevant actions are monitored reliably. We believe a certification process of this kind can be made practicable, in some cases at least. Experimentation with other possible methods is one of the main topics of the next phase of the development.

We have several promising candidate formalisms for representing agents' beliefs about the institutional state. In particular we are investigating the use of an adapted version of the action language $(\mathcal{C}+)^{++}$ [20, 21] for this purpose.

7   Related Work

Electronic institutions have attracted a lot of attention amongst AI researchers recently. Thus far they have been used in the domain of e-Commerce, most notably in the form of auctions [11, 15, 19] and allocation processes [23, 24]. Frameworks have been developed which help with the specification of organisational requirements and verification of electronic institutions. Two examples are OMNI [24] and ISLANDER [15, 19].

We have used the ISLANDER formalism and associated specification tool to design the ballroom as an electronic institution. This work has been undertaken in collaboration with Sierra, Noriega, Rodríguez-Aguilar and García-Camino and will be reported elsewhere.

Grossi et al. [13] note that ontologies are used to relate the abstract concepts in which the norms of an institution are defined to the concrete application. They introduce the notion of contextual ontology and formally characterise it using a description logic. We avoid the need to translate from norms expressed in an abstract way to operational norms by making the conventions and norms downloadable at run-time. Agents still need to be aware of concepts from an ontology, but in our case these concepts are limited to role names, purposes and introspective predicates.

As we have pointed out in the introductory section, there has been a lot of research on normative agents, some of which pre-dates the work on the institutional view on norms (i.e. [5, 6, 10]). Our work attempts to take into account the institutional responsibilities as well as internal beliefs and desires when choosing a course of action for each agent. The agent architecture we outlined facilitates this by providing a way to combine external norms with internal mental states.

8 Summary and Future Work

The aim of this paper was to present a novel and rich case study for investigating normative behaviour in multi-agent systems. We described a ballroom simulation where interaction protocol templates can be downloaded at run-time together with a representation of behavioural norms and conventions associated with them. We outlined an agent architecture that incorporates these ideas and suggested that for some multi-agent systems no central regulatory authority is needed, and can be replaced by a virtual representation where each agent is responsible for maintaining its own partial view of the relevant institutional state. A main aim of our experiments with this example is to determine more precisely the relative merits and disadvantages of this approach, and to identify the classes of applications in which it can be used.

The fundamental issues we raised in this paper must be taken into account when trying to make norms operational in a distributed environment. We hope to stimulate scientific discussions with our thoughts and will continue to work with the ballroom scenario in a variety of ways. Below, we describe some of our ideas.

In order to achieve full norm-awareness, agents need to be able to reason about norms. In some cases, they will adopt a norm and in others they will not. This depends a lot on their current beliefs, desires and intentions. For example, a female agent, who believes strongly in emancipation is unlikely to adopt a norm that says 'A female has to wait to be approached by a male dancer'. We are working on relating a norm or potential commitment to the BDI modalities. The ballroom scenario can then be used to investigate norm adoption by simulating

different ballrooms with different conventions and allowing the agent to choose which ballroom to go to depending on its agreement with the conventions.

A second line of research is to investigate more formally the openness of the system. In a fully open system that makes as few assumptions about the agents' capabilities as possible, the conventions will need to be expressed in an XML-like ontology language (like OWL) for agents to read and reason about them (using an OWL reasoners like [22]). Queries such as *'Can I still dance with X?'* will need to be answered by an external, institutional service. The feasibility of this approach has to be contrasted to our de-centralised (but less open) approach where introspective predicates are used.

We are currently also working on extensions which include planning (allowing agents to commit further into the future), standing obligations (the mother-in-law example) and action histories (implemented using a variant of the Event Calculus as in [3]). All of these will clearly complicate the internal representation of an agent and lead to modifications of our system.

The ballroom example may seem at first sight to be unrealistically simple, a toy example that offers few practical insights. This is not so. As we have tried to indicate in the paper, the example exhibits a very wide and varied range of issues, of exactly the kind that are to be encountered in practical applications of norm-governed multi-agent systems. A multiple-auction application, for instance, shares many of the same features. What is deliberately simple in the ballroom example is the implementation of the agents' internal decision making procedures not connected to the representation of norms – in the example, how to dance a specific number of dances, and how to select potential partners. This allows us to focus on the primary question of interest, which is the representation and possible operationalisations of norms. We are confident that lessons learned from experiments with the ballroom example will be directly applicable in practical applications.

Acknowledgements

References

1. Alexander Artikis. *Executable Specification of Open Norm-Governed Computational Systems.* PhD thesis, Imperial Colllege London, 2003.
2. Alexander Artikis, L. Kamara, Jeremy V. Pitt, and Marek J. Sergot. A protocol for resource sharing in norm-governed ad hoc networks. In *Proc. 2nd Workshop on Declarative Agent Languages and Technologies (DALT'04), New York, July 2004*, 2004.

3. Alexander Artikis, Jeremy V. Pitt, and Marek J. Sergot. Animated specifications of computational societies. In Cristiano Castelfranchi and W. Lewis Johnson, editors, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS 2002)*, pages 1053–1061, New York, NY, USA, 2002. ACM Press.

4. Alexander Artikis, Marek J. Sergot, and Jeremy V. Pitt. Specifying electronic societies with the Causal Calculator. In Fausto Giunchiglia, James Odell, and Gerhard Weiss, editors, *Agent-Oriented Software Engineering III. Proc. 3rd International Workshop (AOSE 2002), Bologna, July 2002*, LNCS 2585, pages 1–15. Springer, 2003.

5. Guido Boella and Leonardo Lesmo. Deliberate normative agents. In R. Conte and C. Dellarocas, editors, *Social order in MAS*. Kluwer Academic Publishers, 2001.

6. Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur. Deliberative normative agents: Principles and architecture. In *Agent Theories, Architectures, and Languages*, pages 364–378, 1999.

7. Keith L. Clark and Francis G. McCabe. Go! for multi-threaded deliberative agents. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 964–965, New York, NY, USA, 2003. ACM Press.

8. Keith L. Clark and Francis G. McCabe. Go! - a multi-paradigm programming language for implementing multi-threaded agents. *Annals of Mathemathics and Artificial Intelligence*, 41(2-4):171–206, 2004.

9. Keith L. Clark and Francis G. McCabe. Ontology schema for an agent belief store. *International Journal Of Human Computer Studies (to appear)*, 2006.

10. Rosaria Conte, Rino Falcone, and Giovanni Sartor. Introduction: Agents and norms: How to fill the gap? *Artificial Intelligence and Law*, 7:1–15, 1999.

11. Marc Esteva. *Electronic Institutions: from specification to development.* PhD thesis, Institut d'Investigació en Intelligéncia Artificial, Bellaterra, Spain, 2003.

12. Andrés García-Camino, Juan Antonio Rodríguez-Aguilar, Carles Sierra, and Wamberto Vasconcelos. A distributed architecture for norm-aware agent societies. In *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. Declarative Agent Languages and Technologies workshop (DALT'05)*, 2005.

13. Davide Grossi, Huib Aldewereld, Javier Vázquez-Salceda, and Frank Dignum. Ontological aspects of the implementation of norms in agent-based electronic institutions. In *Proceedings of NorMAS'05, First International Symposium on Normative Multiagent Systems*, Hatfield, April 2005.

14. Fabiola Lopéz y Lopéz and Michael Luck. Modelling norms for autonomous agents. In M. Mejia E. Chavez, J. Favela and A. Oliart, editors, *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)*, pages 238–245. IEEE Computer Society, 2003.

15. Pablo Noriega. *Agent Mediated Auctions: The Fishmarket Metaphor.* PhD thesis, Universitat Autònoma de Barcelona, 1997.

16. Sascha Ossowski. *Coordination in Artificial Agent Societies: Social Structure and its Implications for Autonomous Problem-Solving Agents*, volume 1535 of *Lecture Notes on Artificial Intelligence*, chapter Distibuted Artificial Intelligence, pages 48–55. Springer Verlag, Berlin, 1999.

17. Pietro Panzarasa, Nick R. Jennings, and Tim J. Norman. Formalising collaborative decision-making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, 12(1):55–117, 2002.

18. David Robertson. A lightweight coordination calculus for agent systems. In João Alexandre Leite, Andrea Omicini, Paolo Torroni, and Pinar Yolum, editors, *Declarative Agent Languages and Technologies II, Second International Workshop, DALT 2004, Revised Selected Papers*, volume 3476 of *Lecture Notes in Computer Science*, pages 183–197, New York, NY, USA, 2004. Springer.

19. Juan Antonio Rodríguez-Aguilar. *On the Design and Construction of Agent-mediated Electronic Institutions*. PhD thesis, Institut d'Investigació en Intelligéncia Artificial, 2003.

20. Marek J. Sergot. $(\mathcal{C}+)^{++}$: An action language for modelling norms and institutions. Technical Report 2004/8, Department of Computing, Imperial College London, 2004.

21. Marek J. Sergot. Modelling unreliable and untrustworthy agent behaviour. In B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka, editors, *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, Advances in Soft Computing, pages 161–178. Springer-Verlag, 2005.

22. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. Submitted for publication to Journal of Web Semantics.

23. Javier Vázquez-Salceda. *The Role of Norms and Electronic Institutions in Multi-Agent Systems*. Birkhaeuser Verlag, 2004.

24. Javier Vazquez-Salceda, Virginia Dignum, and Frank Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, November 2005.