Coordination, Organization, Institutions and Norms
in Agent Systems

COIN

12th International Workshop

COIN@AAMAS2011

May 3, 2010

Taipei, Taiwan

Stephen Cranefield

Pablo Noriega (Eds.)

COIN Steering Committee

Alex Artikis (National Centre for Scientific Research "Demokritos", Greece)
Nicoletta Fornara (University of Lugano, Switzerland)
Eric Matson (Purdue University, USA)
Julian Padget (University of Bath, UK)
Jeremy Pitt (Imperial College London, UK)
Vivian Torres da Silva (Fluminense Federal University, Brazil)
Wamberto Vasconcelos (University of Aberdeen, UK)
Javier Vazquez (Technical University of Catalonia, Spain)
George Vouros (University of the Aegean, Greece)

COIN@AAMAS 2011 Co-Chairs

Stephen Cranefield (University of Otago, New Zealand)
Pablo Noriega (IIIA-CSIC, Spain)

Programme Committee

Alexander Artikis (National Centre for Scientific Research "Demokritos", Greece)
Guido Boella (University of Torino, Italy)
Olivier Boissier (ENS Mines Saint-Etienne, France)
Cristiano Castelfranchi (ISTC/CNR, Italy)
Antonio Carlos da Rocha Costa (UCPEL, Brazil)
Virginia Dignum (University of Utrecht, Netherlands)
Marc Esteva (IIIA-CSIC, Spain)
Nicoletta Fornara (University of Lugano, Switzerland)
Jomi-Fred Hubner (University of Blumenau, Brazil)
Catholijn Jonker (Delft University of Technology)
Christian Lemaitre (Metropolitan Autonomous University, Mexico)
Victor Lesser (University of Massachusetts Amherst, USA)
Eric Matson (Purdue University, USA)
John-Jules Meyer (Utrecht University, Netherlands)
Simon Miles (Kings College London, UK)
Eugenio Oliveira (University of Porto, Portugal)
Andrea Omicini (University of Bologna, Italy)
Sascha Ossowski (University Rey Juan Carlos, Spain)
Julian Padget (University of Bath, UK)
Alessandro Ricci (University of Bologna, Italy)
Juan Antonio Rodriguez-Aguilar (IIIA-CSIC, Spain)
Tony Savarimuthu (University of Otago, New Zealand)
Christophe Sibertin-Blanc (IRIT, France)
Jaime Sichman (University of Sao Paulo, Brazil)
Viviane Torres da Silva (Fluminense Federal University, Brazil)
Munindar Singh (North Carolina State University, USA)
Catherine Tessier (ONERA, France)
Leon van der Torre (University of Luxembourg)
Wamberto Vasconcelos (University of Aberdeen, UK)
Javier Vazquez-Salceda (Universitat Politecnica de Catalunya, Spain)
Harko Verhagen (Stockholm University, Sweden)
Marina de Vos (University of Bath, UK)
George Vouros (University of the Aegean, Greece)
Pinar Yolum (Bogazici University, Turkey)

Additional Reviewers

Luciano Coutinho (University of Sao Paulo, Brazil)
Henirque Lopes Cardoso (Universidade do Porto, Portugal)
Ozgur Kafali (Boğaziçi University, Turkey)

Table of Contents

Operationalization of the Sanctioning Process in Hedonic Artificial Societies

Tina Balke[1] and Daniel Villatoro[2]

[1] University of Bayreuth
Chair of Information Systems Management
Bayreuth, Germany
tina.balke@uni-bayreuth.de
[2] Artificial Intelligence Research Institute (IIIA)
Spanish Scientific Research Council (CSIC)
Bellatera, Barcelona, Spain
dvillatoro@iiia.csic.es

Abstract. With the advent of highly distributed and populated artificial societies where centralized coordination is unfeasible, normative multiagent systems have moved into the focus of attention – as they are promising for improving agent interactions and minimize social frictions. However, an important point that deserves to be studied in detail is what happens when agents behave egoistically and possibly violate the norms they should comply with. The objective of this work is to present an integrated view of the *Sanctioning Process* and analyze each of its phases with regard to its operationalization in artificial societies. Moreover we review the sanctioning mechanisms presented in the multiagent literature and examine them in the context of our proposed process.

1 Introduction

Crime and punishment have been a constant in human affairs for as long as love and greed, or feed and fight; and have moved the will of men, and men found the need to control that will for the benefit of society. Likewise, in order to function well, artificial and technologically enabled societies are finding it necessary to establish regulatory frameworks and ways to enforce them.

Enforcement is concerned with trying to ask (and answer) what happens if an agent does not comply with the obligations or prohibitions of a normative system as well as how to react to that. As stated by Ågotnes et al. [1] or Coleman [2], for this reason *compliance* is one of the most important issues associated with normative systems. In this paper we will particularly focus in detail on one means of enforcement: we will analyze sanctioning[3] as one device to motivate normative compliance[4].

The main motivation of this work is to build a complete map of the sanctioning process. Thus, although many papers have dealt with sanctioning already (e.g. see the

[3] Despite one of the authors elsewhere [3] differentiated between sanction and punishment, for the scope of this work both words are considered synonyms and refer to the same process.

[4] For us, enforcement is a process and sanctioning is one possible specification of this enforcement process.

works mentioned in Section 4), to the best of our knowledge, the so far existing works do not attempt to give an overview of the design options available in the sanctioning process. These works normally fix the restrictions necessary for their environment and focus on a particular mechanism or framework. These restrictions of the environment and the assumptions made about the assignment of roles, as well as the different foci of the analyzed works, make the information on these design options very dispersed. As a consequence it is difficult to get an overview of the choices that can be made when designing a sanctioning mechanism.

We try to close this gap by presenting a comprehensive overview on the design options and roles relevant when talking about sanctioning.

We do not intend to reinvent the wheel. On the contrary, our first aim is to to elucidate the relevant facets of sanctioning in this context, establish pertinent conceptual distinctions and eventually provide a coherent framework so that available complementary developments may be brought together and missing pieces become evident. Moreover, as we indicated above, we have a second aim which is to make practical use of sanctioning as a motivational device for normative compliance in actual MAS. Thus we intend to shape our framework so that MAS development platforms may readily incorporate sanctioning as a resource.

As mentioned before, to us sanctioning, being one means of enforcement, is a process. These process is composed of different phases that – although interlinked – can be designed with a degree of independence. That is why, in this paper we will analyze the *sanctioning process* and the phases it consists of. By identifying all the features in each of the phases that compose the sanctioning process, we aim at obtaining a general view of the overall process.

In order to derive at our overall sanctioning process, in the next section, an introduction to the general state of the question and the literature associated with enforcement and sanctioning will be given. Furthermore, we delineate our object of study making our simplifying assumptions explicit and setting our project within the field. One of the aasumptions is that we think about hedonistic societies in the utility sense. Thus, emotions, despite playing a significant role in the santioning process, will not be investigated at this stage. In Section 3 we then study the sanctioning process as a sequence of phases and particular components that we believe deserve serious ulterior formal treatment. Next in Section 4, we take a step towards our second aim and do a first matching exercise against five well documented MAS platforms for regulated open MAS. We close this paper with an outline of our current and future work.

2 State of the question

Literature associated to the notion of sanctioning is overwhelming. Two traditional approaches, moral and legal, over centuries of study and practice, have established not only a rich conceptual universe around the notion of sanctioning but innumerable devices to address its practical aspects as well. We will draw inspiration from those sources, however we will base our proposal on works that are closer to the MAS world.

A crude classification of sanctioning-related MAS literature may end up with three major blocks.

There is a first group of Economics-inspired works that have studied sanctioning as incentive for rational behavior of utility-based agents. In this area, sanctioning is seen as an amount taken from an agent's benefits and the effectiveness of sanctioning is usually measured against system equilibria, in line with the theory of Becker [4]. The topic has been framed mostly in terms of mechanism design and the issues that economists have studied more thoroughly are the information about infraction and sanctioning [5], as well as the amount and pervasiveness of sanctioning [6]. Methodology has been either game-theoretic (see [2] for example) or experimental (e.g. [7]). In this latter case classical experimental economics methods have been enriched with agent-based simulation and thus explore situations that are difficult to examine with human subjects-only set-ups [8].

A second block is inspired by formal normative concerns, as they apply to agents and MAS. Some works deal with sanctioning, and incentives in general, as a component of the notion of norm and thus study the structural relationships of those components. For instance the relationship between target and victims[5], the syntax of activation and deactivation conditions or links between infractions and reparatory actions [9]. Others are interested in the dynamics of norm-compliance thus deal with sanctioning as events triggered when an infraction occurs [10]. Finally there are works that take sanctioning as a feature that depends on the type of norm (conventions, social, regimented, functional,... [11]). These are interested, for instance, in the operational semantics of compliance and enforceability [12], or in the class of incentives most naturally associated with different norm types [3]. Works in this block range from the strictly formal to the mostly computational approaches.

The third block is composed of works that are concerned with the cognitive aspects of punishment. Some works are interested in the role of sanctioning in the adoption or internalization of norms [13], others in the role or modes of learning and adaptation that may be associated with sanctioning [14], and yet others in the way different ways of sanctioning being more or less effective depending on the cognitive "make-up" of agents [3].

Our work will be based on knowledge from the three blocks and for the purpose of this paper – since our purpose is merely exploratory here – we will use that common knowledge without specific reference.

Our first task will be to state our subject matter with some precision. That is impose some simplifications to the sanctioning process in open MAS, so we will state our intended meaning of some terms and narrow down the scope of the discussion.

2.1 Regulated open MAS

Objective MAS Technically speaking we assume the existence of a MAS environment –observable as an objective entity– with fixed ontology, dynamic state, regimented constitutive conventions and participation of capable and entitled agents.[6]

[5] Detailed description of our understanding of the terms is given in Sec. 2.3

[6] That is (i) There exists a *world* that may be populated by (ii) *agents* whose activity may involve other agents and possibly other entities –like speed limits, traffic lights, or fines. (iii) At any point in time, that world is in a *state* which is represented by a set of variables whose values

Open MAS The decision-making processes of agents may be not under the control of the MAS. Agents may respond to their own motivations and be owned by different owners. Agents may enter or leave the MAS at will, as long as they are capable and entitled to be in the MAS.

Regulated MAS Assuming: (i) Interactions are regulated by norms. (ii) Agents may decide whether they comply with those norms or not. (iii) Only observable behavior is subject to be regulated. (iv) Norms are intended to be enforced.

2.2 Norms

Norms are artificial restrictions on agent behavior. In principle, norms dictate what actions are permitted, prohibited or obligatory under given conditions as well as the effects of complying or not with those norms.[7]

Norm structures are the components that define a norm for the purpose of punishment. For the moment we assume the following: the *antecedent* determines the activation and termination conditions; the *consequence* specifying the intended behavior; *target* agents that should abide by the norm; *goal* indicating the purpose and beneficiaries; *associated norms* whose activation / deactivation is dependent on activation / deactivation this norm; the *triggered actions* explaining its effects and consequences (mostly sanctions and reparatory actions).

Normative system is formed by (i) a regulated open MAS, (ii) the set of norms that regulate it, (iii) the conventions for the creation, issuance, diffusion and change of norms, (iv) a social structure of the MAS involving roles and relationships among roles, (v) appropriate governance mechanisms for norm-enforcement, (vi) performance criteria that determine the quality of the normative system.

Normative context is (i) a normative system, (ii) a set of participating agents and (iii) a non-empty set of states of the MAS.

Roles are a way of defining the tasks that differentiate groups of participants, thus may be seen as entitlements and obligations of any agent that may execute a given collection of tasks. Hence, a normative system includes norms that define whether an agent may change roles or perform more than one role at a given time, what roles may be compatible or incompatible, and whether roles may have hierarchies and of what types.

Norm types in terms of sanctioning (see [15] for example): (i) Constitutive and regimented norms are impossible to violate (e.g., traffic lights have three colors). (ii) Conventions, should be observed —because otherwise some action would not be effective— but are not directly enforced and have no direct sanction associated with them (e.g.,

---

may change only due to the actions of agents. (iv) Out of the many conceivable actions that, in principle, agents may attempt, the only ones that are deemed *admissible* in that world may have any effect in the world. (v) All actions are subject to preconditions (on the state of the world) that make them feasible and post-conditions that determine their intended effect in the world. (vi) Only agents that are capable of performing admissible actions and have the proper authorization may exist and act in the world.

[7] In general, norms are *explicit*, although norms may indeed be an emergent feature of a normative context and sanctioning a means for turning implicit disposition of individuals into norms to be observed in society as a whole.

salute protocols). (iii) Functional, or regulative/enforceable norms should be observed and if violated some corrective action may ensue.

2.3 Governance

These are devices to make norm compliance feasible and enforceable.

Sanctions and punishment. Non-compliance may involve some penalty on transgressors and, compliance some rewards. We will only refer to sanctioning in this paper, although *mutatis mutandis*, the framework applies to both –to rewards and to sanctions– in general. Only prescribed and perceivable behavior may be sanctioned.

Type of sanctions. Following [16], we recognize two: *direct* and *indirect.* Direct sanctions affect the agent immediately and are noticeable directly (like bans, fines and physical punishment). Indirect sanctioning affects only the agent's future actions and may be ostensible or not (e.g. warnings, reputation). One interesting feature of indirect sanctions, such as reputation, is the subsumtion of roles involved: by spreading rumors about a *violator*, every agent becomes a *judge* (as any agent can reinterpret the rumor however it wants) as well as an *executor* (any agent may use the rumor in what ever way it wants, and redistribute it to whom ever it wants). Hence, indirect sanctions might be irreversible, uncontrollable and unmeasurable obtaining (in a worst case scenario) an infinite sanction against the *violator*.

Objective of sanctions. While the general purpose of sanctioning is to motivate norm-compliance, we may distinguish different effects such sanctions are intended to have on future behavior. Customary objectives are deterrence, compensation, retaliation and, finally, exemplification and learning.

Utilitarian agents will be used in this paper for illustration purposes. In the tradition of Hedonism we assume that all agents try to maximize their own utility – hence we refer to hedonic artificial societies in the title. As a result, for the agents, we assume that agents' goals, motivations and decision-making processes take into account many issues that may be amalgamated into a single value —*utility*— which is computable through an "utility function". Thus any sanction will be associated to an increment or a decrement of the utility of transgressor and victims. Furthermore, we will assume that utility functions may be different for different individuals plus the existence of a (general) utility of the system. For utilitarian agents, therefore, sanctions are always financial penalties. In their case, the value of indirect penalties, like reputation, usually cannot be calculated.

Non-compliance qualifications allow the possibility of distinguishing degrees and gravity of non-compliance of some norms and, consequently, modes and intensity of their sanctions.

Governance roles involve entitlements and capabilities of MAS participants to detect and evaluate transgressions, and to impose sanctions and enact reparatory actions and, in some normative systems the possibility of adding or modifying norms (e.g. *legislator*). Some normative systems assume complete institutional governance (all governance is in the power of the objective MAS), while others delegate governance in specialized agents. Some systems are mostly self-regulated and still others may have only the governance that each participant may grant to its own actions.

Two basic roles (with regard to norms) have already been identified [2]: the beneficiaries and the targets. *Targets* are the actors for whom the norm is specified for. *Beneficiaries* are those actors who benefit from the norm and potentially hold the norm. However, some roles are still missing with regard to governance: the violators, the victims, the profiteers, the observers, the judges, the executors, the controllers and the legislators. *Violators* are the actors that have taken a different action that the one specified by the norm. *Victims* are the actors who suffer the consequences of a norm violation. *Profiteers* are the actors who benefit from the consequences of a norm violation. *Observers* are those actors that identify a violator. *Judges* are those actors that given the information about the violation are able to calculate the sanction to be applied to the violator. *Executors* are the actors that apply the sanction to the violators. *Controllers* are the actors that –after the sanction has been applied– control that the sanction had the desired effect. *Legislators* are the actors that observe the efficiency of the system with respect to the fulfillment of norms and application of sanctions.

In some cases an actor might assume more than one of these roles. The combination of roles that each actor posses is completely related with the type of society agents are located, and vice-versa. A society where all the roles are played by different agents is completely different to a society where all the roles are played by all agents.

Normative system dynamics. While the manifest purpose of sanctioning is to motivate compliance, a collateral effect is the adaptation of behavior of individuals to the normative context and in a deeper level, the adaptation of the normative system to an evolving normative context. Thus the cost and effectiveness of sanctioning may induce changes in the sanctioning process in order to better achieve its goals, and may eventually call for a modification of the whole normative system. Both situations should be contemplated in the governance devices of the regulated MAS and, depending on the system design, they may be embedded in the system or handled off-line.

3 The Sanctioning Process

As shown in Figure 1 we conceptualize sanctioning as a four-stage process. The first three stages correspond roughly to the conventional processes of arrest, trial and conviction of transgressors, while the fourth is the process of learning and adaptation that ensues. Each stage involves distinctive activities whose performers are agents playing particular roles. Although in general terms most activities and roles are present in every regulated MAS that includes enforceable norms, they need to be adapted to the particularities of the normative context where violations take place. Here we will sketch the general contents but limit occasional comments to utilitarian agent sanctioning.

3.1 Detecting Non-Compliance

This process has two goals: the ascertainment of a violation and the identification of agents involved. Two obvious roles take part in this phase: *violator* and *observer*. One should note that, depending on the structure of the norm that is violated, the observer may need to gather enough evidence to ascertain what violation actually took place befor any further punitive actions can take place. Hence, in order to bring about charges,

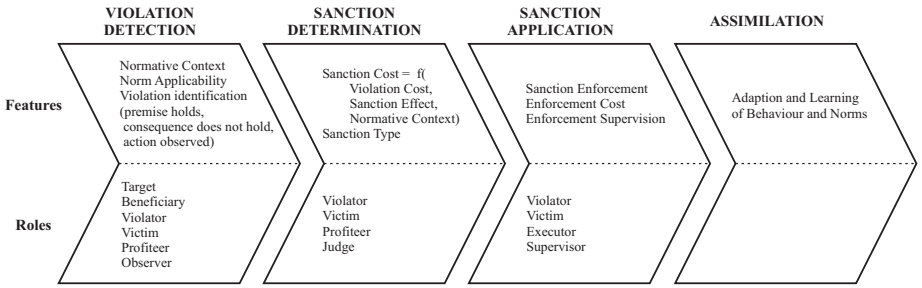


Fig. 1. The Sanctioning Process

observers may need to assess *damages*, to assign *blame* and to identify *victims* and *profiteers* that may be affected by the non-compliance.

Each of these roles may be performed by more than one agent and, in totally self-regulated MAS, even performed by the same agent. In the particular case of utilitarian agents, the observer role may take different forms:

- A *first-party observer* who controls its own compliance. Violations may be purposeful or accidental but the observer needs to realize that it misbehaved.
- A *second-party observer* who observes a misbehavior of a transaction partner(s). An example would be the buyer on eBay, who has bought and paid a product, but the seller does not send it to him. If the product was to be send insured, she can recognize a violation by the seller if she does not receive a parcel on time.
- *Trusted third-party observers* that supervise the behaviour of other agents in the system. Axelrod's Norms Game [17] has a good example where all participants are observers. The scenario is an exam in a classroom. Agents (students) can detect other agents cheating and accuse them of doing so. Therefore, all agents also act as observers.
- *Institutional watchmen* are agents whose decision-making and action is controlled by the MAS and have as goal the detection of transgressions and their proper assessment. Each one of these may have limited observation capabilities, as for example, the Second Life Governance Team, where a team of authority representatives patrol the grid, ensuring the correct behaviour of the citizens. If they are not present at the scene of a norm violation, then, they cannot detect it. Evidently, full observability may be achieved placing watchmen in every scene, as the case of staff agents in electronic institutions [18], who control every interaction.
- *Institutional enforcement* another way of achieving full observability of violations is embedding it as a functionality of the MAS as a whole [12, 19], however especially in open distributed MAS this approach is unfeasible due to the nature of the system.

The question whether all violations may be detected or not may determine strategic decisions of agents with respect to norm fulfillment. In scenarios where every violation is observed, agents would violate norms only if they consider that the benefits of a

violation are higher than the penalty. On the other hand, when violations might pass unnoticed, agent strategic behavior is also possible but then benefits are compared against expected (uncertain) penalties.

3.2 Sanction Evaluation

This phase involves the appraisal of the applicability of the norm within the normative context of the non-compliant action and, if applicable, activation of the normative consequences of infringement and determination of the actual sanction to be imposed. In this stage, *observers* bring charges to a *judge* who should decide if the violator deserves a sanction and then determined the appropriate sanction. The judge may also command the execution of reparatory actions as a consequence of the infringement. In some regulated MAS, violators and victims may be party to argumentation processes to establish applicability and severity of sanctions.

Sanctions are usually calculated as a function of the violator, the victims, the effects produced by the violation, and the normative context where the violation was detected. This sanction calculation might imply a cost which is absorbed by the *judge*. Some may need to be enacted either automatically or commissioned by the judge.

During sanction calculation for utilitarian agents, one should not assume that all agents are regimented by the same utility function. Even though taking this factor into account when calculating a sanction is crucial, this is unfortunately impossible, as normally the utility function of each agent is private (i.e. only known by the respective agent). In case this utility function was accessible, the sanctioning entities would be able to calculate a perfect sanction to decrement the utility function of the violator, even if they do not share that function. Consequently, to ensure the intended effect of a sanction we have to be aware of the agents' utility function, or at least, the *judges* and the *executors* agents do use a sanction that affects the utility of the *violator*.

One final decision has to be made before applying the sanction, and this is how the sanction should applied. The way a sanction is applied can be decided by the *judge* or be imposed by the normative context.

3.3 Sanction Application

The Sanction application phase is composed of the actual execution of a sanction and the assessment of its proper application.

The outcome of the previous stage is a sentence to be carried out, and the role that does it is the *executor*. Executors may be [20]:

- *A first-party victim*: depending on the types of norms that are controlling the society, one agent can result directly harmed by another agent's norm violation. If this is the situation, the victim can act as the observer, calculator and applicant of the sanction to the violator.
- *A third-party observer*: if an agent is seen violating a norm, the observing agent can have the right to apply a sanction to the violator, even if this agent did not suffer from the violation.

– a *A group of third-party observers*: the act of sanctioning can be distributed amongst a group of agents. This type of sanctioning act is often used in indirect sanctions such as reputation (which has an aggregated effect, the more agents use it, the more powerful).
– *Institutional enforcers* such as authority agents who do not observe all actions but only the ones within their vicinity, and do have the power (designated by the institution that they all belong to) to apply the sanction.
– *the violator* himself, who after violating a norm unintentionally, it might want to repay the damage done (maybe to avoid a loss in reputation).

Sanctions may have a cost associated to its application that may be bore by *executors* or the regulated MAS. Sometimes the cost are directly associated with the sanction. A straight forward example of *costly sanction* is imprisonment, where the state has to support jailmates. However, the application of other sanctions imply an unknown cost to the *executors* but still impinging damage to the *violator*. Reputation is normally one of the most effective *relativized-cost* sanctioning mechanism. Transmission of (bad) information regarding an agent has a relative cost (depending on the degree of resposability of the information transmitted and the retaliation level of the members of the society) to the agent transmitting the information, however, it might also affect the target of the rumor.

After the *executor* has acted, the *supervisor* is in charge of controlling that the *violator* has indeed received the sanction, and that the sanction has served the purpose for which it was designed. The *supervisor* is also responsible for ensuring that other actions that are associated with the infringed norm are properly triggered and carried out. In case of compensational sanctions, in particular, the *supervisor* will ensure that the *victims* are compensated.

3.4 Assimilation

Assimilation is the processes through which individuals or the normative system itself take advantage from a sanction to modify ulterior behavior.

As we have seen, by performing the corresponding roles, violators, observers, victims, judges, executors come in contact with information about norm-compliance that they could ideally incorporate in their decision-making baggage and may hence shape their own future behavior. The normative system however, may specify ways that information about the punishment process that is not the one that participants obtain directly through interactions, may become available to participants. By so doing, norms about punishment may give shape to a space for individual evolution where specific aspects of compliance are given more relevance than others and therefore facilitate evolution of the system along different lines.

While punishment is intended as a motivation for actions and the sanctions of specific specific norms may have an ostensible objective (compensation, retaliation, deterrence, exemplarity), the actual effect of punishment in an agent's motivations is a private matter of convicts, thus directly unobservable for the MAS. Nevertheless, the ulterior behavior of agents is observable, hence the MAS as a whole, or its *legislators*, may use evolution of the behavior of individuals as input for *purposeful* evolution of the

normative system. Autonomic MAS would use evolution of individuals' behavior for *self-adaptation*. In either case, modifications of behavior need to be observable somehow. The natural means are probes and indicators that must be aligned with performance parameters accessible to legislators or the system dynamic features.

This way, the choice and balance of those conventions that determine availability and form of information about punishment and norm-compliance have significant effects in the overall collective behavior. Collective behavior that results from the aggregate behavior of informed individuals, as well as collective behavior determined by new norms that result from legislative or autonomic adaptation.

4 Contrasting the Process

Having presented our sanctioning process in the last section, in this section we review some of the existing MAS sanctioning approaches in the context of our process (due to space constraints this review is not complete, although we believe is representative).

4.1 eInstitutions [18, 10]

The eInstitutions model consist of formal semantics for the notion of an institution and its components (abstract and concrete norms, empowerment of agents, roles) and defines a formal relation between institutions and organizational structures. In the original design of eInstitutions (eI) the sanctioning was done with the help of regimentation mechanisms. In eI, all actions are triggered with the help of messages. These messages that trigger actions by the individual agents are observed by staff agents (that are implemented in the middleware of the system). The staff agents check every message and do not forward it further in case it violates a norm. This way normative compliance was unavoidable. As a consequence of this rigor non-compliance identification, very little options for the remaining two phases exist. However, recent work [10] has started to include norm violations considerations, although these violations are detected by the institution, which is ready to apply the sanction to the violator.

4.2 OperA [21]

The OperA model is a framework for agent societies with the goal to legitimate the concept of autonomy between agent goals and society requirements. The model makes use of contracts that predefine the norms and their corresponding sanctions in case of violations. All the parameters of the contract are negotiated, becoming the transaction partners the *targets*, *beneficiaries*, potential *violators*, and *judges*. Hence in the calculation phase of our model, existing well defined sanctions are assumed and do not have to be reasoned about. In order to verify that agents fulfill the contracts Dignum introduces the idea of Trusted Third Parties (TTPs) that verify the compliance at "run-time". These TTPs are called monitoring agents. Even though in theory one can sign a contract without a TTP, this is rarely done. TTP observe the system and in case of violations induce the further sanctioning process playing the role of *observer*.

4.3 López y López et al. [22]

The work by López y López et al. has its roots in the SMART agent framework. This framework integrates our sanctioning process idea although no specification of the phases is given. Moreover, a much more abstract (compared to our hedonic) concept of agents is considered. In the work by López y López et al. agents follow norms to obtain their normative goals, without considering how these goals are fulfilled. More specifically, authors make a distinction between primary and secondary level norms, and their relation with the concept of interlocking. When a primary norm is violated, it activates (as they are interlocked) a second level norm in charge of sanctioning.

The responsibles for the sanctioning on this platform are the norm *defenders*. In our role classification, they act as *observers*, *executors* and *controllers*. The role of *judge* is not clear by whom is played, as the second level (sanctioning) norm already specifies the sanction and different situations with conflicting norms that might influence the calculation of this sanction are considered.

One very interesting point of the work by López y López et al. is the formalization of an autonomous normative reasoning process. This process is composed of other three that allow agents to decide whether to adopt a norm, whether to comply with the norm, and to update the goal of agents. This process is very important in open distributed systems as it allows adaptation and dynamicity against this rapidly changeable environments.

4.4 $\mathcal{MOISE}^{Ins}$ [19]

The $\mathcal{MOISE}^{Ins}$ framework follows a similar structure as the eInstitutions. It employs generic supervisor agents, aiming at controlling and enforcing the rights and duties of autonomous "domain" agents operating in an normative organization expressed with $\mathcal{MOISE}^{Ins}$. Whereas supervisor agents (who play the roles of *observers*, *judges* and *executor*) are dedicated to the control of the system at the normative and sanctioning level, the domain agents implement the functionalities of the application level. Hence, $\mathcal{MOISE}^{Ins}$ envisions agents to be enforced to comply with norms (although they have the freedom of violation) because the violations are being detectable by the system. One aspect of the model that is of special interest to the sanctioning process is the discussion of the normative learning and adaption by the society as a result of the sanctioning act presented by the authors. Thus, in $\mathcal{MOISE}^{Ins}$ agents may decide to adapt and change the organization in a bottom-up process if they feel that the current normative system is not suitable, installing a new normative pattern/structure. The phases of our process are also very well identified from the observation by the *supervisor agents*, passing through the calculation of the sanction (done during the design of the system), and the application of the sanction by the supervisor again.

4.5 EMIL-I-A [14]

EMIL-I-A is an agent architecture able to process the normative cues in the social environment where agents interact. This architecture allow agents to self-organize and

regulate their own norm-compliance within the system in a decentralized manner, without the necessity of a central authority. Agents' decision of whether or not following the norms is orchestrated by the internalization mechanism agents are endowed with, and by the norms' salience, as the baton that orchestrates the intelligent norm compliance. Following the philosophy of decentralization, agents assume all the roles, working as *observers*, *judges* and *executor*. Because of this unification of roles and adaptability of the EMIL-I-A architecture, agents have a plastic notion of norms, making the terms of compliance and violation also dynamic and dependent on agents' perception of the environment. Authors present a running example of their architecture using a simulation framework that approaches the establishment the of cooperation norm. All the phases discussed in this work are perfectly recognizable in their architecture, although the sanction calculation is imposed by the system.

5 Closing Remarks

This paper is a toddling start towards an organization of features akin to sanctioning, to facilitate their use in the design of regulated open MAS. We chose the unconventional perspective of sanctioning as a process, in order to plot a field where available resources are easily harvested while unexplored opportunities are more crisply revealed. Thus, we outlined the bounds of the punitive process and advanced core conceptual distinctions. Our quick exploration of these elements in the context of five MAS platforms suggests our proposal is valid and points out obvious topics to address next. In this last section we will touch upon the not so obvious.

We have sidestepped the fundamental question of why sanctioning works. If addressed from the assumption that sanctioning not only motivates but *modifies* behavior, it entails, first, an examination of tenants about an agent's rationality; in particular, in relation with the agent's motivations and in the context of the agent's conformity to a society. It also entails an analysis of the type of modification that is desired or achievable through sanctioning, those factors that bear upon behavior modification and the way behavior modification is achieved and ascertained. Utilitarianism on one hand, and on the other BDI architectures of different flavors, are standard position to avoid many of the problematic aspects of rationality. With respect to punishment, however, even these simplifications still leave hard questions open. At any rate, our description of the four-stage process should have made clear that a designer of sanctioning mechanisms for a given MAS, may have to consider many different facets of the mechanism, even assuming absolute control over the cognitive capabilities of agents.

A similar type of concerns, likely as rich, may be risen with respect to the "assimilating sanctioning" stage of the punitive process. We hinted at two foci of analysis: individuals and system's, hinted at the pertinence of the abundant work on agent and MAS evolution and, in particular, that of emergence and immersion of norms, but there is more. The process of sanctioning may be understood, also, as a resource to modulate those dynamics thus making evolution faster, or more sensitive to the success of sanctioning, or the other way around take the intended evolution as the modulator of sanctioning. In both directions, the study of how to determine the efficiency of a normative system and its sensitivity to changes in sanctioning is paramount. Of special

significance is the interplay of these matters with the moral and cognitive disposition of agents.

Our last comment is on implementational aspects of the punitive process. The distinctions we outlined in this paper are but a fraction of the number of design features that are amenable for use in actual MAS. Still more work is needed to come up with a concise framework for a design methodology and the development and assembly of tools that allow for a convenient use of a sanctioning process on top of generic MAS development platforms. The first steps should be to start with platforms like the five we examined and make sanctioning available as a service on top.

6 Acknowledgments

This work was supported by the Spanish Education and Science Ministry [Engineering Self-* Virtually-Embedded Systems (EVE) project, TIN2009-14702-C02-01]; Proyecto Intramural de Frontera MacNorms [PIFCOO-08-00017] and the Generalitat de Catalunya [2005-SGR-00093]. Daniel Villatoro is supported by a CSIC predoctoral fellowship under JAE program. Tina Balke was supported by a scholarship of the German Academic Exchange Service. Both authors thank Pablo Noriega for his valuable discussion of and comments on the paper.

References

1. Ågotnes, T., van der Hoek, W., Tennenholtz, M., Wooldridge, M.: Power in normative systems. In: AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2009) 145–152
2. Coleman, J.S.: Foundations of social theory. (August 1998)
3. Andrighetto, G., Villatoro, D., Conte, R., Sabater Mir, J.: Simulating the relative effects of punishment and sanction in the achievement of cooperation. In: Proceedings of 8th European Workshop on Multi-Agent Systems. (2010)
4. Becker, G.S.: Crime and punishment: An economic approach. The Journal of Political Economy **76**(2) (1968) 169–217
5. Fehr, E., Gächter, S.: Cooperation and punishment in public goods experiments. The American Economic Review **90**(4) (2000) 980–994
6. Dreber, A., Rand, D., Fudenberg, D., Nowak, M.: Winners don't punish. Nature **452** (2008) 348–351
7. Gurerk, O., Irlenbusch, B., Rockenbach, B.: The competitive advantage of sanctioning institutions. Science **312**(5770) (April 2006) 108–111
8. Carpenter, J.P.: Punishing free-riders: How group size affects mutual monitoring and the provision of public goods. Games and Economic Behavior **60**(1) (July 2007) 31–51
9. Perreau De Pinninck, A., Sierra, C., Schorlemmer, M.: A multiagent network for peer norm enforcement. Autonomous Agents and Multi-Agent Systems **21** (November 2010) 397–424
10. García-Camino, A.: Normative regulation of open multi-agent systems. PhD thesis, Artificial Intelligence Research Institute (IIIA) (2009)
11. López y López, F., Luck, M.: Modelling norms for autonomous agents. In Chavez, E., Favela, J., Mejia, M., Oliart, A., eds.: Fourth Mexican International Conference on Computer Science, IEEE Computer Society (2003) 238–245

12. Grossi, D., Aldewereld, H.M., Dignum, F.: Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, M., Fornara, N., Matson, E., eds.: Coordination, Organizations, Institutions, and Norms in Agent Systems II, Springer (2007) 101–114
13. Conte, R., Andrighetto, G., Campenni, M.: Internalizing norms. a cognitive model of (social) norms' internalization. The International Journal of Agent Technologies and Systems (IJATS) **2**(1) (2010) 63–73
14. Andrighetto, G., Villatoro, D., Conte, R.: Norm internalization in artificial societies. AI Communications **23** (2010) 325–339
15. Villatoro, D., Sen, S., Sabater-Mir, J.: Of social norms and sanctioning: A game theoretical overview. International Journal of Agent Technologies and Systems **2** (2010) 1–15
16. Vázquez-Salceda, J., Aldewereld, H., Dignum., F.: Norms in multiagent systems: from theory to practice. International Journal of Computer Systems Science & Engineering **20**(4) (2004) 95–114
17. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review **80**(4) (1986) 1095–1111
18. Esteva, M.: Electronic Institutions: from specification to development. PhD thesis, Artificial Intelligence Research Institute (IIIA) (2003)
19. Boissier, O., Gâteau, B.: Normative multi-agent organizations: Modeling, support and control, draft version. In Boella, G., van der Torre, L., Verhagen, H., eds.: Normative Multi-agent Systems. Number 07122 in Dagstuhl Seminar Proceedings (2007) Internationales Begegnungs– und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany
20. Balke, T.: A taxonomy for ensuring institutional compliance in utility computing. In Boella, G., Noriega, P., Pigozzi, G., Verhagen, H., eds.: Normative Multi-Agent Systems. Number 09121 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany (2009)
21. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. PhD thesis, Utrecht University (2003)
22. López y López, F., Luck, M., d'Inverno, M.: A normative framework for agent-based systems. In Boella, G., van der Torre, L., Verhagen, H., eds.: Normative Multi-agent Systems. Number 07122 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2007)

Organizationally Adept Agents

Daniel D. Corkill,[1] Edmund H. Durfee,[2] Victor R. Lesser,[1] Huzaifa Zafar[1], and Chongjie Zhang[1]

[1] University of Massachusetts Amherst, Amherst MA 01003, USA
[2] University of Michigan, Ann Arbor MI 48109, USA

Abstract. An *organizationally adept* agent (OAA) is not only aware that it is part of an agent organization and about its role(s) in that organization, but it can also assess how well it is fulfilling its organizational responsibilities and can proactively adapt its behaviors to meet organizational needs better. OAAs evaluate their behaviors based not only on their (agent-centric) self-interests, but also on their (organization-centric) responsibilities to each other and their (social-centric) willingness to perform activities requested by other agents.

Agent organizations designed for less adept agents must specify detailed guidelines that, when blindly followed, will influence individual agents to work together in the expected environment. However, if the environment deviates from expectations, such detailed organizational guidelines can mislead agents into counterproductive or even catastrophic behaviors. Organizations designed for OAAs, on the other hand, can assume that agents will reason about organizational expectations, and will adjust their behaviors when the nominal guidelines misalign with those expectations.

We describe an extended BDI reasoning architecture for an OAA that balances organizational, social, and agent-centric interests and can adjust this balance when appropriate. Then we show how we used the reasoning architecture with agents operating in RoboCup Rescue scenarios.

Keywords: organizationally situated agents; multi-agent organizations; adaptive agent behavior; emergent agent organizations

1 Introduction: Organizational Structuring and Control

Despite creating highly capable software agents, widespread deployment of large multi-agent applications has not occurred, in large part because coordinating many individual agents' activities to achieve collective benefit in open, dynamic environments is hard. Statistical approaches (like those inspired by insect colonies) allow miscoordination so long as aggregate behavior of the population is acceptable [2], but can waste valuable agent resources. More effective coordination can arise when each agent reasons about the past, present, and future activities of all agents, given the evolving environment state, to make informed decisions. This can work well for small teams and coalitions [17], but quickly

becomes intractable in larger agent populations. Avoiding intractability in large-scale settings by pre-specifying what activities each agent should perform cannot cope with an environment that changes unpredictably.

A variety of techniques for scalable coordination have been investigated. In general, these try to reduce the awareness an agent needs of others. For example, agents can exploit *locality* to only be aware of nearby agents, such as where an agent responsible for piloting a robotic truck needs only model other nearby vehicles. Or agents can utilize *aggregation* to model many other agents with a small set of parameters, such as how a product's price quote from an auction can summarize the bidding behaviors of all the buyers and sellers. Underpinning these and other scalable coordination techniques, however, are more stable structuring decisions about permissible interactions (e.g., the "rules of the road") and communication patterns. These structuring decisions in turn can require considerable designer effort and insight to ensure that operational coordination mechanisms (e.g., negotiation protocols and auctions) can perform effectively and tractably. Furthermore, the resulting structures cannot adapt themselves to changing circumstances without human intervention, and might rely instead on compensating (over)use of operational coordination techniques which will either yield lower-quality coordination decisions, incur greater amounts of overhead, or both.

These limitations can be confronted by treating organizational control organically within a multi-agent system rather than relying on an external (human) designer. *Organizational control* is a multi-level approach to coordination in which organizational goals are identified and used in determining agent roles and responsibilities [4]. These roles and responsibilities provide a context for making detailed *operational control* decisions by the individual agents. Organizational control is distinct from operational control in both time span and level of detail. Organizational roles and responsibilities specify general, long-term guidelines that are subject to ongoing elaboration and revision by the agents [6]. Operational control, on the other hand, involves specific short-term agreements among agents to perform specific activities at specific times [12]. Organizational guidelines reduce the complexity of operational decision making, lower the cost of distributed resource allocation and agent coordination, help limit inappropriate agent behavior, and reduce unnecessary communication and agent activities.

Corkill and Lander [5] note that effective multi-agent organization becomes increasingly important with: 1) increases in the number of agents and their diversity; 2) increases in the duration of agent activities; 3) increases in the repetitiveness of agent activities; 4) increases in resource sharing; 5) increases in agent collaboration and the complexity of interactions; 6) increases in agent specialization; 7) decreases in agent capability; and 8) decreases in resource slack (or increases in performance requirements). As agent-based applications become more widespread and complex, effective organizational coordination will become an important aspect of system performance. We are just reaching the point of creating multi-agent applications where organization makes a meaningful difference, and soon it will become the critical difference.

In the next section, we discuss organizationally adept agents in detail, focusing on the reasoning activities that they must perform. In Section 3 we describe a BDI-based architecture that we have developed to enable an agent to perform organizationally adept reasoning. Then we show how we used the reasoning architecture with agents operating in RoboCup Rescue [11] scenarios.

2 Organizationally Adept Agents

Central to effective organizational control is the creation of *organizationally adept* agents (OAAs) that can reorient their local activities based on their interpretation of the organizational intent, allowing emergent organizational behavior within designed organizations. Emergent behavior in multi-agent systems has been the topic of considerable interest and research [3, 14, 1]. We are primarily interested in emergent *organizational* behavior where individual agents identify interaction and local control decisions that have been effective in the past and give preference to similar decisions in the future. Gasser and Ishida were among the first to consider emergent organizational behavior as a means of organization self-design [7]. In their work, each agent used what it knew about how the organization was performing in its local neighborhood to make unilateral changes in its organizational behavior. Kamboj and Decker extended the Gasser and Ishida work to more complex multi-agent environments [10]. We, and other researchers, have used learning and diagnosis techniques to enable agents to determine emergent behavior [16].

An OAA must be able to: a) adapt its behavior given explicit expectation-annotated organizational directives, b) determine appropriate emergent organizational behaviors on its own (perhaps within organizationally delimited boundaries) when organizational expectations are not met or when externally designed directives are absent, and c) report deviations from organizational expectations as well as new emergent organizational behaviors to other agents.

OAAs are especially beneficial in settings where either the system objectives or the task-environment characteristics are not fully known. In addition, OAAs provide mechanisms to adapt (both short and long-term) to changing environmental conditions. However, the benefits provided by OAAs diminishes when the objectives of the system are not identified at all, when an agent cannot generate expectations of its own task performance or measure if it is achieving those expectations, and when agents are unaware of their neighbors and expectations of their performance. The importance of agent expectations (and the OAA approach) does not depend on whether the agents are assembled by negotiated agreements between self-interested agents or whether they are inherently inclined to achieve common social objectives.

2.1 Functional OAA Architecture

Fig. 1 presents the basic functional architecture of an OAA. The agent's organizational behavior derives from a combination of annotated organizational

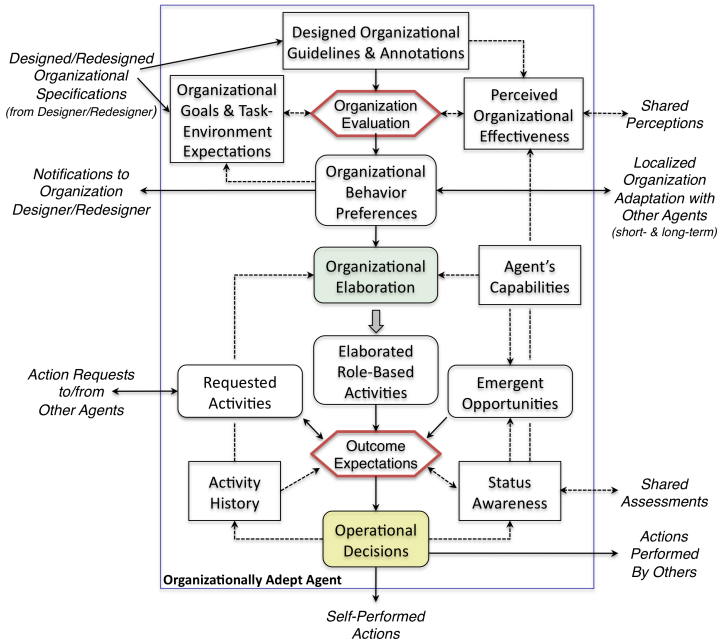


Fig. 1. Organizationally Adept Agent

guidelines that are disambiguated and elaborated into specific actions by the OAA and from actions identified from the agent's local view of its opportunities (which may not align with the organizational guidelines).

The activities shown in the boxes at the top of the figure are associated with organizational control aspects of the OAA. An organization designer provides the OAA with organizational guidelines and performance expectations that are elaborated into specific role-based (organization-centric) operational activity possibilities. Activities to be performed by an OAA can also be requested by other agents, and these social-centric activity possibilities are shown at the lower left side of the figure. On the lower right side are emergent (agent-centric) opportunities stemming from the agent's awareness of its environment, capabilities, and status. Decisions about what organizational, social, and self-centric possibilities should be performed by the agent (or should be requested to be performed by other agents) are made by the "Operational Decisions" function at the bottom center of the figure. This reasoning activity is at the heart of the OAA approach and, after presenting an example of the type of agent behavior that we expect an OAA to perform, we describe a BDI-based implementation of this core OAA functionality.

2.2 An Example of OAA Behavior

Consider an initial organization, shown in Fig. 2, for the RoboCup Rescue domain. As part of this organization, the city is divided into four regions. A *center*

is assigned the role of managing each region. The activities of a center include: maintaining information about buildings in its region, collecting status reports (such as location, water levels, etc.) from each fire-brigade agent within the region, determining the number of fire brigades that are needed to extinguish a building on fire, determining which fire brigade agents to assign to which buildings, and requesting help from neighboring centers when required. Each center is provided with organizational guidelines that include the fire-brigade agents that the center manages[3] and parameters for the center's managerial role (for example, the boundary of the region it manages). Additionally, the center's organizational guidelines are annotated with task-environment expectations provided by the organization designer.[4] For example, annotations might specify that the center should typically have to manage at any given time no more than: one medium task (requiring two fire brigades to extinguish) and one small task (requiring one fire brigade); three small tasks; or one large task (requiring 3 fire brigades). Assuming that organization was designed for a city where fires are evenly distributed across regions, an equal number of fire brigades have been assigned to each region.

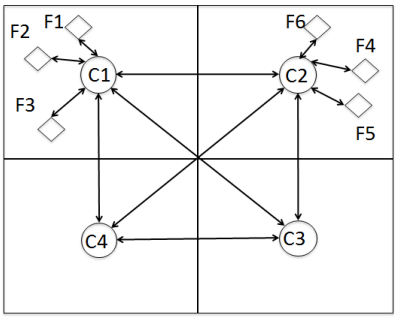
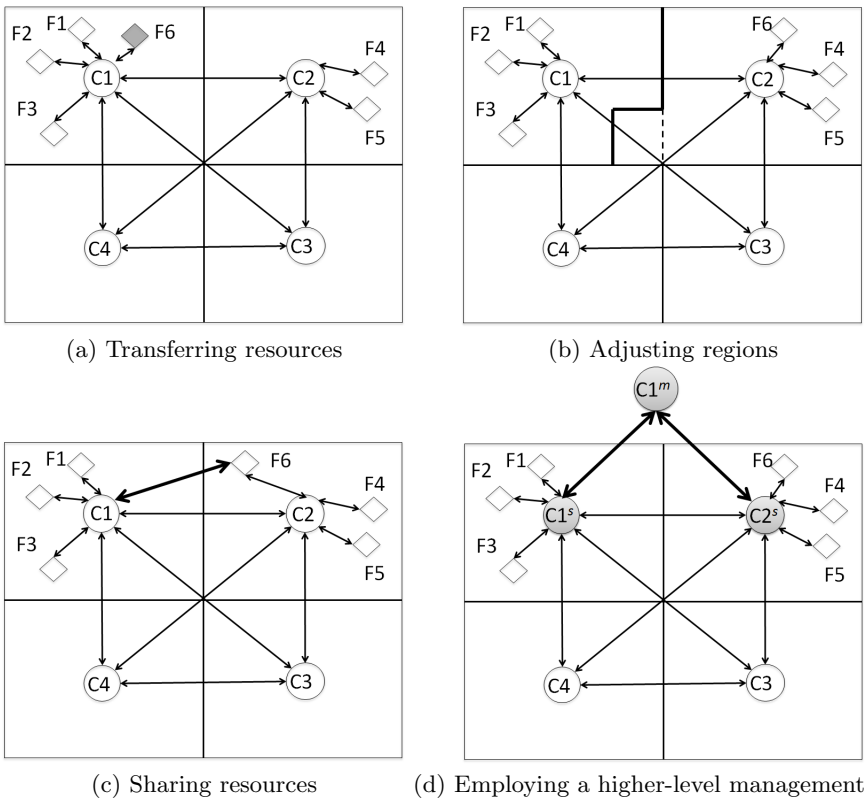


Fig. 2. An example RoboCup Rescue organization where the city is divided into four regions, with a manager (centers C1, C2, C3, and C4) responsible for each region. Agents assigned the role of extinguishing fires (fire brigades F1 to F6) are also shown for the C1 and C2 regions.

Now, consider center C1 operating in the Fig. 2 structure. As new fires are discovered, C1 begins allocating resources (assigning fire brigades the task of extinguishing buildings on fire) to fires in its region in accordance with the parametrized role assigned to it—even though it could potentially perform other activities (such as allocating resources to fires outside its region). Suppose C1 discovers new fires in its region that require more resources than C1 has available. What should C1 do as an OAA? First of all, C1 needs to make operational decisions about how it should address the immediate problem. Should C1 allow new fires to stack up until it can assign its fire-brigade resources to them (hoping that the arrival rate will drop back to manageable levels quickly)? If C1 delays responding to fires too long, buildings might burn down or, worse, the fires might spread to neighboring buildings, compounding the problem. Given this, should C1 delay

[3] Those fire-brigade agents also having been assigned the role of extinguishing fires as directed by the center.

[4] In this paper, we do not care if the agent organization is designed by a human or by an automated (potentially distributed) designer process [15, 9]. In either case, we assume that annotations sharing the designer's expectations are provided along with an agent's organizational guidelines to help an OAA recognize when the assumptions used by the designer become invalid.



(a) Transferring resources (b) Adjusting regions

(c) Sharing resources (d) Employing a higher-level management role

Fig. 3. Adjustments to the organization (from Fig. 2)

dealing with the newly discovered fires? Or, should C1 assign a smaller number of resources to these fires with the goal of preventing them from spreading, rather than extinguishing them? Should C1 ask a neighboring center for help, and if so, which center? If C1 decides to seek help from a neighboring center, should it simply borrow additional resources (and incur an added span-of-control challenge in managing the additional brigades) or should it transfer the responsibility of handling the fire to the neighboring center?

Furthermore, from an organizational-control perspective, C1 needs to consider whether the overload situation is: 1) a transient burst whose effects are so brief that they lie within the environmental expectations that were assumed when the organization was designed; 2) a longer duration event that is outside the organizational expectations (but will return to normal eventually); or 3) a change in the environmental situation from the original assumptions, and the design is no longer appropriate.

In deciding what to do, C1 must generate candidate operational-action choices and consider potential longer-term changes to its organizational behavior. Consider how C1 might proceed under case 2, above, where the longer-term

volume of fires in its region exceeds both C1's resource constraints and its organizational expectations. Assume that this situation has become recurring, and that each time it has occurred C1 has addressed it operationally by borrowing resources from a neighboring center. Most of the time, C1 has borrowed brigades from C2, and C1 is now approaching C2 first in order to reduce the operational cost of seeking resources from other centers. The next step in addressing this recurring situation is to move from repeated operational "fixes" to a long-term agreement between C1 and C2.

An OAA makes localized organizational-behavior modifications by obtaining long-term agreements with other OAAs. In doing so, OAAs assess, implement, and evaluate the effect of these behavior modifications locally, and then, if appropriate, report them to the organization designer as information to be used in modifying the overall organization design, if appropriate. As shown in Fig. 3, C1 and C2 can form one of several local long-term agreements:

Transferring resources—If C2 perceives that it generally has a spare fire brigade even when operating near the limit of the task-environment expectations associated with its organizational guidelines, C1 and C2 can agree to permanently transfer control of a fire brigade (e.g., F6) to C1. To accomplish this agreement, C1, C2, and F6 need to form a long-term agreement among themselves and annotate it with the corresponding environment expectations.

Adjusting Regions—If C2 perceives that the long-term task volume is lower than its local processing capability and organizational expectation but sometimes it needs all three fire brigades, C1 and C2 can mutually decide to adjust the region boundaries so that the expected task distribution between the two new regions is in line with organizational expectations.

Sharing Regions—Regions can also be adjusted so that they overlap. Fires in buildings in the overlap area can be handled by either center (providing increased resource flexibility), but at the complexity of having the centers determine who will handle each fire in the overlap area. For example, C1 and C2 could decide to negotiate each time, or simply allow the center with more available resources to take charge.

Sharing Resources—If C2 perceives that the long-term task volume is in line with its local processing capability and organizational expectations and that C1 and C2 rarely require simultaneous use of three or more fire brigades, C1 and C2 can agree to share a fire brigade (e.g., F6). That is, both C1 and C2 can direct F6 and F6 reports its status information to both centers. The agreement with F6 needs to specify how it will deal with conflicts, such as if C1 and C2 request it to fight different fires at the same time. For example, F6 could be directed to always prefer requests from C2 over requests from C1, or to seek resolution of the conflict from the centers, or F6 could be allowed to decide for itself which fire to fight.

Employing a Higher-Level Management Role—If C2 perceives that the long-term task volume is in line with or exceeds its local processing capability and organizational expectations (but rarely do C1 and C2 require three or more fire brigades simultaneously), C1 and C2 can share their resources more efficiently

by agreeing to enable a new high-level super-manager role, C^m, to coordinate decision making of C1 and C2. This new role can be performed by either C1 or C2. Assume C1 undertakes this new role, $C1^m$. When either $C1^s$ or $C2^s$ (the original C1 and C2 roles are replaced with "subordinate" center roles that have diminished responsibilities due to the enabling of the C^m role) is overloaded with fires to extinguish, $C1^m$ requests (as part of its organizational responsibilities) status information of $C2^s$'s fire brigades, makes a more coordinated allocation decision, and notifies $C2^s$ to execute the part of the decision that requires its fire brigades.

3 An Extended BDI Architecture for an OAA

Our OAA architecture is built on a variant of the classic BDI model [13]. To perform the decision-making required by an OAA, we introduce organizational-level reasoning that affects the normal operational-level reasoning by incorporating organizational guidelines as beliefs and modifying the goal-evaluation mechanism used for deciding which operational goals to pursue at any moment. In addition to problem-domain beliefs, goals, and plans, we explicitly represent beliefs, goals, and plans that support organizational-level reasoning. Conventional BDI agents make decisions based on their local interests. For an OAA, however, organizational guidelines and requests from other agents also affect the agent's operational decisions. To enable the OAA to balance the potentially conflicting interests of organization, others, and self, we structure the evaluation function of operational goals to include three weighted components that explicitly represent these three interests. We will discuss these extensions after we briefly describe the basic BDI architecture.

A BDI agent has a *belief set*, a set of *primitive actions*, and a *goal library*. Its belief set is composed of *beliefs* that represent the agent's knowledge about the world, including its local state, the surrounding environment, and in an OAA, the organization, and the interests of other agents. A belief is defined by a set of variables and associated values. Primitive actions are executable, which provide the capabilities of the agent. The goal library contains a set of generic goals. A generic goal can be instantiated when its precondition is satisfied. A generic goal is specified by the following components:

Type—The label that indicates the type of the goal.

Precondition—A logical condition that must be true in order to instantiate the goal. The condition is verified on the agent's belief set. This condition does not have to remain true during the execution of the goal's plan.

Postcondition—A logical condition whose value becomes true once the goal is achieved. This condition is also verified on the agent's belief set.

Plan algorithm—The method that generates a plan to achieve the goal based on the belief set.

Utility function—The function that computes the utility of the goal given the belief set.

Algorithm 1: The BDI-Agent Reasoning Process

input: Goal library GL and initial belief set B_0

$B \leftarrow B_0$; /* B_0 is the initial belief set*/
$cg \leftarrow nil$; /* Initialize with no committed goal */
repeat
  Get percept p;
  $B \leftarrow updateBeliefSet(B, p)$;
  $G \leftarrow options(B, GL, cg)$; /* Instantiate top-level goals */
  $cg \leftarrow filter(B, G, cg)$; /* Select and commit a goal from goal set G */
  $\pi \leftarrow plan(B, cg)$;
  $executePlan(B, GL, cg, cg, \pi)$; /* recursively execute the plan */
until *the process is terminated*

Algorithm 1 shows the reasoning process of a BDI agent. When the agent perceives the environment, receives messages from other agents, or notes a change in the status of one of its ongoing actions, it updates its belief set. Based on the updated belief set, the agent instantiates and evaluates goals (which are also called *desires*). The agent then determines the goal(s) with the highest utility and commits to achieving them. A committed goal is also called an *intention*. Once committed to a goal, the agent will select and instantiate a *plan* from its plan library (or generate a plan by using a predefined algorithm) in order to achieve the goal. A plan for a goal consists of a sequence of primitive actions or subgoals. Goals and plans form a hierarchy, called a *goal-plan tree*. Note that committed goals only refer to top-level goals in the hierarchy.

Algorithm 2 illustrates how an agent executes a plan to achieve a top-level committed goal. The execution of a plan is generally sequential. But if a plan contains subgoals, it will be executed recursively. During plan execution, the agent will constantly update its belief set based on its perceptions and potentially instantiate new top-level goals. If a newly instantiated goal has a higher utility than a currently committed goal (and the agent cannot execute plans for them both), the method $reconsider(B, cg, G)$ will return true and the execution of the plan of that currently committed goal will then exit. Without considering the uncertainty of the environment, the satisfaction of a top-level goal requires the satisfaction of all subgoals. If the environment is uncertain or an agent has only partial observability, it is possible that the goal is achieved without achieving all subgoals of its plan or the goal is not achieved even when all subgoals of its plan are achieved. Given uncertainty of the environment and of the effects of the actions, the agent needs to check at every time step whether the goal is achieved and, if not, that its plan remains sound. This verification can be done by checking if the precondition of the first primitive action or subgoal of the plan is satisfied. If it is not, the agent needs to create a new plan for the goal.

An OAA reasons at two different levels: organizational and operational. At the organizational level, the agent reasons about whether current organizational guidelines are effective and, if not, how it can improve them by either performing a local organization adaptation or requesting a non-local adaptation from the organization designer. At the operational level, the agent decides which operational goals to pursue and which plans to choose for achieving committed goals. These two types of reasoning focus on goals at different temporal scales.

Algorithm 2: Function $executePlan(B, GL, cg, g, \pi)$

input : Belief B, Goal library GL, committed goal cg, current goal g, plan π_g
output: return $false$ if committed goal cg needs to be reconsidered; otherwise, return $true$

```
1  while not (empty(π) or succeeded(g, B) or impossible(g, B)) do
2      α ← first(π);
3      π ← rest(π);
4      if isPrimitiveAction(α) then
5          do(α);
6      else
7          π_α ← plan(B, α);
8          if executePlan(B, GL, cg, α, π_α) then
9              return false;      /* Committed goal cg needs to be reconsidered */
10         end
11     end
12     Get percept p;
13     B ← updateBeliefSet(B, p);
14     G ← options(B, GL, cg);      /* Instantiate top-level goals */
15     if reconsider(B, cg, G) then
16         return false;
17     end
18     if not sound(π, B, g) then
19         π ← replan(B, g);
20     end
21 end
22 return true;
```

Organizational-level reasoning is intended to achieve long-term organizational goals (organizational objectives), while operational-level reasoning focuses on immediate operational goals. However, these two reasoning levels interact. Organizational guidelines (reasoned about at the organizational level) provide preferences or constraints for an OAA's choice of operational goals and plans. These guidelines specify what roles the agent should perform, priorities for these roles, and how these roles are parametrized (e.g., under what conditions these roles should be activated, who the agent should communicate with, etc.). In turn, the effectiveness of organizational guidelines is reflected in how well operational goals (weighted by their importance) are accomplished by the agent. Thus, OAA reasoning needs to include beliefs about the status of operational goals.

The top-level goal of organizational-level reasoning is to assess and maintain the effectiveness of an agent's organizational guidelines. Each OAA has three generic plans to achieve this goal that can be invoked in different situations: using existing organizational guidelines, performing localized organizational adaptation, and requesting non-local adaptation from the organizational designer. For example, if current guidelines perform as expected, the agent needs to follow them for making decisions at the operational level; otherwise, it needs to decide whether to adapt locally or more globally. Its decision-making relies on its knowledge about the current situation, i.e., its belief set. At the organizational level, the agent perceives messages from the organizational designer or other agents that perform organizational adaptation. Its belief set contains current

organizational guidelines, performance annotations, and environmental expectations. To evaluate the effectiveness of the current organizational guidelines, the OAA needs to monitor the results of its operational goals over time, updating beliefs about its own performance (e.g., the failure frequency of operational goals). In addition, its belief set also needs to reflect environment changes to determine whether the changes are within its expectations, a transient deviation from its expectations, a frequent, long-duration deviation from its expectations, or a permanent deviation from its expectations [8].

At the operational level, the agent's perceptions, belief set, and goals are usually domain-specific, which is similar to the implementation of conventional BDI agent architectures. However, as agents interact and perform within an organization, the decision-making of each OAA needs to balance the interests of itself, the organization, and other agents. As we discussed above, in our OAA architecture, the goal evaluation function largely determines the agent's decision-making process. Therefore, we structure the utility function of goals to reflect explicit separation of organizational, social, and self-interests. In our current OAA implementation, the utility function $U(g, B)$ for operational goal g with belief set B is defined as the following:

$$U(g, B) = v(g)(w_s p_s(g, B) + w_o p_o(g, B) + w_r p_r(g, B)),$$

where $v(g)$ is the objective value of goal g, $p_s(g, B)$, $p_o(g, B)$, and $p_r(g, B)$ are the preferences for goal g of the agent itself, the organization, and other agents, whose values are in the range $[-1, 1]$, and w_s, w_o, and w_r are weights, whose values are in the range $[0, 1]$ and are normalized to sum to 1. These weights are also part of an agent's beliefs. An agent can change these weights through updating its belief set based on its assessment of the effectiveness of organizational guidelines, its own experiences, and requests of other agents. The local preference $p_s(g, B)$ reflects the agent's local interest for this goal. The preference $p_o(g, B)$ is determined using organizational guidelines contained in the agent's belief set B. The value $p_r(g, B)$ summarizes the agent's belief of the interest that other agents have expressed (via sent requests) for goal g. Let $p_{r1}(g, B), \ldots, p_{rk}(g, B)$ be the interests of other agents for goal g. In our current implementation, the preference $p_r(g, B)$ is defined as the following:

$$p_r(g, B) = \frac{2}{1 + \alpha e^{(-\beta \sum_{i=1}^{k} p_{ri}(g,B))}} - 1,$$

where $\alpha > 0$ and $\beta > 0$ are constants (i.e., $\alpha = 50$ and $\beta = 8$). It is possible that, for some goal, there are no organizational guidelines relevant to it or no requests from other agents. If this happens, we set the corresponding weights to zero and normalize the weights so that they sum to 1.

4 Evaluation

We are only beginning development of a complete OAA architecture that provides the complete adaptive capabilities described in Section 2. However, we believe that the basic ability to incorporate and balance organizational, social,

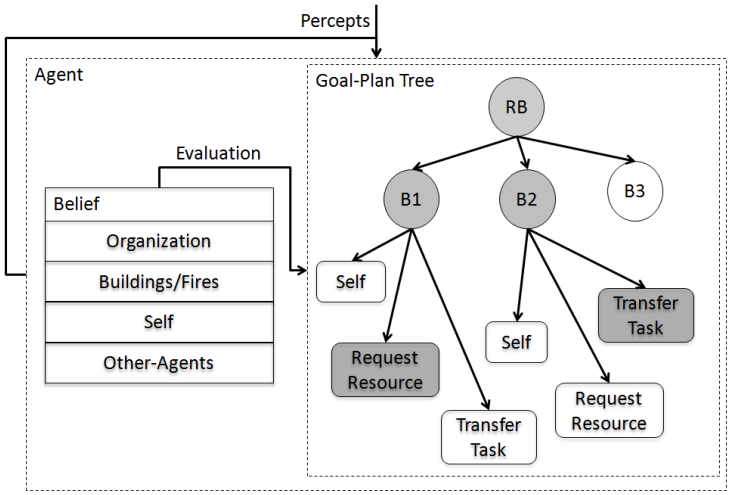


Fig. 4. Center OAA architecture (with an example goal-plan tree). Shown are a general Rescue Buildings (RB) goal and three goals specific to buildings on fire (B1, B2, and B3). Three plans are shown: extinguish building using the center's own resources, request resources from neighboring centers, and transfer the extinguish task to neighboring centers. Selected plans and goals are highlighted.

and self-interests in making operational decisions is a crucial requirement for OAAs that can perform reasonably when there is uncertainty and inappropriateness from the perspective of a single interest. Furthermore, the ability for an OAA to adjust the influence it grants each of these perspectives over time is important to agent and organizational robustness and effectiveness. For example, an OAA must be able to function without organizational guidelines (by increasing the weight of self and social influences) if no guidelines have been supplied or if they become inappropriate. In the next section, we discuss how we used the OAA reasoning architecture as the operational decision engine for agents extinguishing fires in RoboCup Rescue.

4.1 Using the OAA Architecture in RoboCup Rescue

We implemented the OAA architecture for centers in RoboCup Rescue (Fig. 4) Each agent receives percepts, from itself (characterizing the internal status of the agent), from external factors in the environment (from the RoboCup Rescue simulator), and from other agents. These percepts include organizational guidelines received from an automated designer[5] (e.g. the geographic region the center is managing), tasks (e.g. fires in the region, task-transfer requests), and resource statuses (e.g. water levels of fire-brigades the center is controlling, resource-transfer requests). Based on these percepts, each agent updates the following types of beliefs:

[5] Not discussed in this paper.

Organization Beliefs (OB)—represent the organizational guidelines given an agent including the roles it is responsible for and specific parameters of those roles. For example, a role that a center performs is task allocator. The guidelines parametrize that role with the fire-brigade agents that the center has authority over and the geographical region it is responsible for managing.

Building/Fire Beliefs (BB)—represent the various buildings in the environment, whether they are on fire or not, the importance of the buildings, and the utility received on extinguishing fires in those buildings.

Resource Beliefs (RB)—status information regarding fire-brigades agents directed by the center, including their water level, health, and location.

Other Center Beliefs (CB)—assessment of the willingness of other centers to help this center and of the likelihood of other centers to request help from this center based on past experiences.

As beliefs are updated, each agent forms a local goal-plan tree, where each goal and plan is evaluated based the agent's beliefs. Each center starts with one inherent goal, to extinguish buildings that are on fire. As a new fire percept is received, the agent instantiates a subgoal (B1 for example) for the particular building. For each building on fire, the agent can execute one of three plans: 1) it could try to extinguish the fire using only its own brigades (Self in Fig. 4), 2) it could augment its own resources by requesting brigades from neighboring centers (Request Resources), or 3) it could transfer the task of extinguishing the fire to a neighboring center (Transfer Task). The utility of a goal is calculated using the following function:

$$U(goal, beliefset) = f(goal, BB)(w_s f_s(goal, RB) + w_o f_o(goal, OB) + w_r f_r(goal, CB))$$

Thus, if a center is highly inclined to follow its organizational guidelines, its w_o will be high (e.g., 0.8), while w_s and w_r will be low (e.g., 0.1 each).[6] If a center is strongly self-motivated, w_s will be high. If a center greatly prefers assisting other agents, w_r will be high. These weights are normalized so that $w_s + w_o + w_r = 1$. Note that there are two things operating here. For each goal, its objective utility ($f(goal, BB)$) is boosted by how much the goal agrees with the center's organizational guidelines ($f_O(goal, OB)$), its beliefs about itself ($f_S(goal, RB)$), and its beliefs about its neighboring agents ($f_R(goal, CB)$). On the other hand, the weights influence the center's inclination for goals that agree with its organization-centric, social-centric, and self-motivated interests.

4.2 Demonstrations

We illustrate OAA behavior differences given different (fixed) weights for w_s, w_o, and w_r using RoboCup Rescue scenarios. Consider center C2 and two buildings B1 and B2, both on fire (Fig. 5).

[6] Although we wanted to show the different behavior of OAAs that were heavily weighted toward one of the three interest perspectives, we did not want to eliminate entirely the ability to choose activities stemming from the other two perspectives.

Initially, $w_s = w_o = w_r = \frac{1}{3}$. When the fires are discovered, C2 generates a goal to extinguish each fire (goals B1 and B2). For goal B1, $f_S = \frac{1}{d}$, $f_O = 1$ (since building B1 is within the region of C2's organizational responsibility), and $f_R = 0$ (since extinguishing building B1 has not been requested of C2 by another agent). For goal B2, $f_S = \frac{1}{d}$, $f_O = 0$ (since building B2 is outside the responsibility region that is specified in C2's organizational guidelines), and $f_R = 0$. Because of the added organizational inclination, C2 will prefer accomplishing goal B1 over B2, unless B2 is a significantly more important building than B1 (i.e $f_{B1} << f_{B2}$). Once C2 assigns resources to accomplish goal B1, and if $U(B2, B)$ is greater than a certain threshold, C2 will generate plans for requesting help from neighboring centers for accomplishing B2. Values for f_s, f_o, f_r will be calculated for the goals associated with each of the plans, allowing for C2 to select one of them.

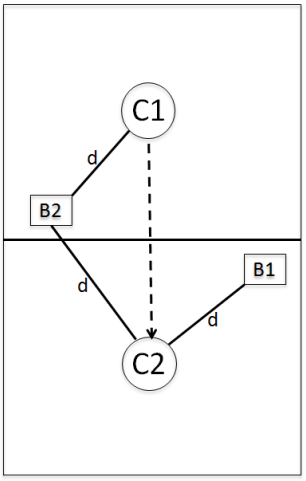


Fig. 5. Center C2 discovers two buildings on fire, B1 and B2 at an equal distance, d, from the center of the region. It has enough resources to handle only one of them. Center C1 discovers fire B2 and sends a request to C2 for help extinguishing it

Extending this scenario further, suppose C1 discovers B2. It will generate a corresponding goal for it and assign values $f_S = \frac{1}{d}$, $f_O = 1$, and $f_R = 0$. Assume C1 does not have enough resources to fight fire B2, and from its perspective $U(B2, B)$ is greater than its threshold. C1 will generate plans for requesting help from neighboring agents (just as C2 did previously). Suppose C1 makes a request of C2 to extinguish this fire. This means C2 will increase its f_R value for goal B2 based on the local importance of handling requests from C1. Now, C2 must make an operational decision between accomplishing goals B1 and B2. If rather than the weights being $\frac{1}{3}$ each, $w_r > w_o$, C2 will assign its resources towards accomplishing goal B2. If $w_r < w_o$, C2 will assign its resources to B1. If both C1 and C2 were to favor self-interests, they both might try to request resources for B2 (because the organizational interest is not accounted for), incurring both the operational cost of making those duplicate requests, as well as the potential cost of over-assigning resources to B2.

These initial demonstrations show that our RoboCup Rescue OAAs not only behave differently when strongly weighted toward different interests (organization, social, self), but that they perform ineffectively when the interest they are emphasizing does not fit their beliefs and environment. More importantly, these demonstrations show the potential for OAAs that maintain a balance between these three important interests and adjust that balance when long-term environmental conditions and agent behaviors warrant. Our OAAs can follow organizational guidelines when they are available, yet operate in their absence or ignore them if they are inappropriate.

Acknowledgement. This material is based in part upon work supported by the National Science Foundation under Award No. IIS-0964590. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. C. Bernon, V. Chevrier, V. Hilaire, and P. Marrow. Applications of self-organising multi-agent systems: An initial framework for comparison. *Informatica*, 30:73–82, 2006.
2. E. Bonabeau, M. Dorigo, and G. Théraulaz. *Swarm Intelligence: From natural to artificial systems*. Oxford University Press, 1999.
3. C. H. Brooks and E. H. Durfee. Congregation formation in multiagent systems. *Journal of Autonomous Agents and Multiagent Systems*, 7:145–170, 2003.
4. D. D. Corkill. *A Framework for Organizational Self-Design in Distributed Problem-Solving Networks*. PhD thesis, University of Massachusetts Amherst, Feb. 1983.
5. D. D. Corkill and S. E. Lander. Agent organizations. *Object Magazine*, 8(4):41–47, Apr. 1998.
6. E. H. Durfee and Y. pa So. The effects of runtime coordination strategies within static organizations. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 612–618, Nagoya, Japan, Aug. 1997.
7. L. Gasser and T. Ishida. A dynamic organizational architecture for adaptive problem solving. In *Proceedings of the National Conference on Artificial Intelligence*, pages 185–190, Anaheim, California, July 1991.
8. B. Horling, B. Benyo, and V. Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 529–536, Montreal, Canada, June 2001.
9. B. Horling and V. Lesser. Using quantitative models to search for appropriate organizational designs. *Autonomous Agents and Multi-Agent Systems*, 16(2):95–149, 2008.
10. S. Kamboj and K. S. Decker. Organizational self-design in semi-dynamic environments. In *Proceedings of the 2007 IJCAI workshop on Agent Organizations: Models and Simulations* (AOMS-07), pages 335–337, Jan. 2007.
11. H. Kitano and S. Tadokoro. RoboCup-Rescue: A grand challenge for multi-agent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.
12. J. G. March and H. A. Simon. *Organizations*. John Wiley & Sons, 1958.
13. A. S. Rao and M. P. Georgeff. Bdi agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems* (ICMAS'95), pages 312–319, San Francisco, California, June 1995.
14. G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos. Self-organisation and emergence in MAS: An overview. *Informatica*, 30:1–11, 2006.
15. M. Sims, D. Corkill, and V. Lesser. Automated organization design for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 16(2):151–185, Apr. 2008.
16. T. Sugawara and V. Lesser. Learning to improve coordinated actions in cooperative distributed problem-solving environments. *Machine Learning*, 33(2-3):129–153, Nov. 1998.
17. M. Tambe, J. Adibi, Y. Alonaizon, A. Erdem, G. Kaminka, S. Marsella, and I. Muslea. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, 110:215–240, 1999.

Modelling and monitoring interdependent expectations

Stephen Cranefield[1], Michael Winikoff[1], and Wamberto Vasconcelos[2]

[1] Department of Information Science, University of Otago, Dunedin 9054, New Zealand
{scranefield,mwinikoff}@infoscience.otago.ac.nz
[2] Department of Computing Science, University of Aberdeen, AB24 3UE, Aberdeen, UK
wvasconcelos@acm.org

Abstract. Previous research on modelling and monitoring norms, contracts and commitments has studied the semantics of concepts such as obligation, permission, prohibition and commitment; languages for expressing behavioural constraints (such as norms or contracts) to be followed by agents in specific contexts; and mechanisms for run-time monitoring of fulfilment and violation of these constraints. However, there has been little work that provided all of these features while also allowing the current expectations of agents, and the fulfilment and violation of these expectations to be expressed as first-class constructs in the language. This paper demonstrates the benefits of providing this capability by considering a variety of use cases and demonstrating how these can be addressed as applications of a previously defined temporal logic of expectations and an associated monitoring technique.

1 Introduction

Much research in multi-agent systems has been influenced by organisational principles from human society, and in particular social concepts such as norms and commitments have been extensively studied due to their potential to enable the efficient specification and management of agent interaction in open societies of autonomous agents.

Previous research on modelling and monitoring norms, contracts and commitments has studied the semantics of concepts such as obligation, permission, prohibition and commitment; languages for expressing behavioural constraints (such as norms or contracts) to be followed by agents in specific contexts; and mechanisms for run-time monitoring of fulfilment and violation of these constraints. However, there has been little work that provided all of these features while also allowing the existence, fulfilment and violation of obligations and commitments to be expressed as first-class constructs in the language. We believe that the ability to directly express statements about these features of an agent's social context is important as it allows the investigation of richer types of norms and contracts that are interdependent. Our aim in this paper is to demonstrate that this is a capability that is desirable but not adequately addressed to date, and show how a logic and monitoring technique developed in our previous work can meet our requirements.

In this paper, we are not concerned with distinctions between norms and commitments, and generalise both concepts to the notion of *expectations* on future world states, events and/or agent actions, while ignoring social issues such as where these expectations come from (e.g. mandated by authorities, inferred through observation and experience, or requested and accepted via agent messaging) and how they are embedded

in the relationships that exist between agents. In our view these issues can be largely decoupled from the questions of what it means to have an expectation that is active, fulfilled or violated, and how these expectations change from one state to the next.

The structure of this paper is as follows. In Section 2 we present a survey of a range of approaches to modelling and monitoring various types of expectations. Section 3 provides an overview of our previously defined logic of expectations (for which we have built an associated model checker for monitoring expectations), and explains why previously imposed restrictions on the nesting of expectation-related modalities can be lifted. Some use cases illustrating the utility of modelling interdependent expectations are presented in Section 4, and the paper is concluded in Section 5.

2 Previous work

A wide variety of approaches have been investigated for modelling and monitoring constraints on agents' future behaviour in the context of electronic institutions, normative multi-agent systems and commitment-based semantics for agent communication. Early work in electronic institutions (e.g. [15, 19, 25]) focused on the development of middleware that can directly interpret an institution specification provided by a designer and ensure that agents follow the norms, and for this reason considered norm representations that have a procedural rather than declarative flavour, giving rise to the so-called "protocol-based norms". Work in the related field of normative multi-agent systems (e.g. [9]) has tended to focus on higher-level declarative representations of norms. Research on commitment-based semantics for agent communication (e.g. [23, 28]) aims to explain the individual speech acts and/or complete dialogs exchanged between agents in terms of the commitments requested and made by one agent towards another.

There are strong links between these research fields with much work crossing the boundaries between them, e.g. the design of a norm representation language that includes operational details such as violation checks and repair strategies alongside a declarative norm [26], the extension of e-institution middleware to handle rule-based norms as well as protocol-based norms [19], and an institution specification language that models both norms and agent communications in terms of commitments [17].

Below we discuss work in these areas, focusing on the formalisms used, whether concepts such as expectation, fulfilment and violation are expressible in those formalisms, and whether (and how) the monitoring of expectations has been addressed. The approaches discussed range from high-level logical models, investigated mainly to gain semantic understanding of norms, commitments or general expectations, to operational models that can be directly executed and are therefore amenable to run-time monitoring. However, there has been no work that provides a good semantic account of the activation, fulfilment and violation of expectations of any sort, allowing these concepts to be explicitly represented, and also providing a technique for monitoring expectations, except for the work of Governatori and Rotolo [21], which addresses only the recovery from violations via contrary-to-duty norms. In Section 3 we show how our prior work on modelling and monitoring expectations can be extended to provide all three features, and argue why this ability opens up a new range of interesting use cases in expectation modelling and monitoring.

2.1 Logical approaches

Dignum and colleagues investigated the extension of dynamic [13] and temporal [10] logics with deontic concepts to allow the expression of obligations involving deadlines. The obligations studied address either the performance of specific actions [13] or the fulfilment of (atemporal) propositions [10] by a deadline. In one approach [13], the semantics of formulae were defined relative to a state and a trace so that "the history (i.e. the trace) of an ideal world might differ from the history of the present world", and this feature was used to define the notion of ideality represented by obligations. Later work used simpler semantics in which models of the logic are assumed to include a propositional constant $Viol$[3]. Broersen et al. [10] also used an ideality proposition Idl to allow a more subtle account of deadline obligations. These propositions have no semantics of their own—they are given semantics indirectly via the definitions of the obligation operators, which constrain the states in which $Viol$ and Idl should hold.

The works cited above did not address the modelling of obligations dependent on the fulfilment or violation of other obligations, except for the simple case of the violation or fulfilment of single actions. Also, in the examples of interdependent obligations considered in this [13] and earlier work [12], rather than explicitly using $Viol$ and Idl predicates as conditions of norms, predicates directly expressing the occurrence or lack of occurrence of the specific desired action are used. While this demonstrates how, for specific examples, an obligation can be made conditional on a predicate that happens to correspond to fufilment or violation of another obligation, there is no systematic treatment of nested violation and fulfilment operators within obligations. This is reasonable given the restricted setting (obligations to perform a given action), but this approach leaves open the question of how inter-related norms with more complex temporal structure could be expressed.

Alberti et al. [1] describe a means to perform run-time protocol compliance monitoring based on logical constraints expressing positive and negative expectations as the consequences of observed actions. At run time, agent messages are detected and asserted as facts, and abductive inference is used to keep track of pending, fulfilled and violated expectations. However, this information about the state of expectations cannot be expressed using constraints, so interdependent expectations cannot be modelled.

Verdicchio and Colombetti [27] use a variant of CTL* to provide axioms defining the lifecycle of commitments based on their making and cancelling as well as requests for them, which come about through the exchange of messages. The language includes predicates to represent a commitment being made and whether it is fulfilled, violated or pending. It seems that these predicates could appear within the content of commitments. There is no discussion of how the language could be used for practical reasoning.

Bentahar et al. [6] define model-theoretic semantics for their Commitment and Argument Network (CAN) formalism that models agent communication in terms of social commitments and argumentation. Their logical language can express the creation of commitments of various sorts and requests for commitments to be made, as well as the satisfaction and violation of commitments. It appears that the satisfaction and violation

[3] In some work it is noted that this propositional constant could be qualified, e.g. with a norm index [12], so that different types of violation can be distinguished.

operators can be nested within the content of a commitment. Their discussion of pragmatic aspects of their formalism [5] does not address the monitoring of commitments.

Singh [23] provides model-theoretic semantics for commitments, with two modalities defining practical and dialectical commitments between a debtor x and creditor y. The language allows these modalities to be nested. The paper discusses possible reasoning postulates and their soundness and completeness, but there is no discussion of how this logic could be used in practice. It is, however, claimed that the approach provides a basis for specifying precisely how commitments arise in a context and can be manipulated. Modalities corresponding to fulfilment and violation are not discussed.

Governatori and Rotolo [21] propose a technique for design-time checking of a set of rules (specifying some process) against a set of normative rules regulating it. This is in contrast with the run-time checking of actual behaviour that is the focus of this paper, and is therefore contingent on the process descriptions to be checked being available for this purpose. The normative language used is based on defeasible logic and has a special focus on "contrary to duty" norms. It thus has an implicit notion of violation of a norm expressible in the language.

Cranefield and Winikoff [11] define an extension of propositional linear temporal logic that includes temporal operators stating that an expectation currently exists, is fulfilled, or is violated as a result of a particular conditional rule of expectation. In contrast to the work discussed above, the concepts of violation and fulfilment of expectations are given their own first class semantics. The logic, as described previously, did not allow formulae representing existence, fulfilment or violation of an expectation to appear nested within a rule of expectation; for example, a rule could not be triggered by the violation of another rule. A model checking procedure allows the truth of these formulae to be determined either off-line (e.g., when checking an audit trail) or incrementally as new states become available. This logic is the basis of the discussion in this paper, and a modified version is described in Section 3.

2.2 Rule languages

García-Camino et al. [19] present a language for defining conditional norms and the sanctions or rewards associated when norms are fulfilled or violated. Norms control the utterance of speech acts within particular periods (specified in terms of dates or relative to other speech acts). Sanctions can modify attributes of an agent, such as its credit. The language is given an operational semantics in terms of the Jess expert system shell, and this allows norm fulfilments and violations to be detected at run-time and sanctions to be applied. However, the occurrence of fulfilments and violations cannot be expressed within the normative rules themselves.

Boella and van der Torre [8] propose a normative systems approach for defining document access control policies in distributed virtual communities. Community and individual policies are represented using beliefs, desires and goals described using rules of the form $l_1 \wedge \cdots \wedge l_n \rightarrow l$, where l and each l_i are literals built from a set of propositional decision variables and system parameters. Decision variables may be defined to represent the situation where a given agent is considered to be in a state of violation if some specific variable is *true*, and this allows nested violations to be represented. For example, agent a_2 may be deemed to be in violation if it does not consider the state

of affairs represented by variable q to be a violation by agent a_1. There is no underlying semantics for violation—the determination that a violation exists is local to an agent based on its rules. This approach allows the negotation of agents for access to information to be characterised using game-theoretic techniques.

García-Camino et al. [20] define an expressive rule language with constraints for specifying conditional norms and explicitly tracking the normative state of a multi-agent system as agents exchange messages. Rules may refer to norms, so it is possible to define rules stating, for example, that one obligation triggers another. Although the rules track the normative state of the multi-agent system and therefore detect violations of norms, these cannot be represented using the proposed set of predicates for representing normative states. As the rule language is not dependent on the predicates used to model states, additional fulfilment and violation predicates could be added. However, the only semantics for predicates are any operational ones defined by rules.

Fornara et al. [17] describe an approach for specifying institutions in which agents communicate. The content of commitments comprise an action, proposition or referential expression existentially or universally bound to an interval of time. Norms are event-driven rules to create, update or cancel commitments. Although the authors advocate the suitability of an operational approach to checking agent norm-compliance, they do not give details as to how this could be done, and their language cannot express fulfilments and violations of commitments and norms.

Aldewereld et al. [2] have considered the use of "counts-as" predicates to link normative (abstract) events with real-world (concrete) events. In particular, obligation and prohibition norms in deontic logic are operationalised as "counts-as" statements: an obligation (respectively prohibition) with content ϕ maps to the statement that $\neg\phi$ (respectively ϕ) counts as a norm violation. Norms can have a maintenance condition (once active, the norm is deemed violated if this condition evaluates to false), and this allows a limited degree of temporal expressiveness. It is not clear if the violations of different norms can be distinguished and there is no discussion of whether the norm violation and fulfilment conditions can be used within the content of norms. The approach is implemented using the DROOLS forward-chaining engine.

2.3 Action description languages

Artikis and Sergot [3] use the event calculus for specifying and tracking normative states of multi-agents systems based on the concepts of obligation, power and permission. Their approach specifies how the actions agents perform affect the values of fluents (dynamic properties) encoding the state of the domain and the powers, permission and obligations of the actors. Obligations represent actions that agents should perform (rather than states of the world they should bring about). A violation fluent is used to declare that an action causes a violation, but this has no special semantics. Farrell et al. [16] present a similar approach for modelling and monitoring the state of contracts.

Commitment machines [28] define agent interaction protocols by specifying the preconditions and effects of the agent actions in terms of commitments that exist between participants. A set of protocol states are defined in terms of the propositions and commitments that hold in them and domain actions are defined in terms of the propositions and commitments they cause to hold (their "effects"). Actions cause transitions

between states if the new state is a logical consequence of the original state and the action's effects. Agents interpret commitment machines at run time to determine a desired path through the protocol, or they can execute a finite state machine compiled from the commitment machine. There is no notion of violation of a commitment in this formalism—an execution of the protocol either reaches a state in which a desired commitment exists, or it does not. Commitments can be conditional on the existence of other commitments, but these represent instances of conditional commitments between agents that were created during the protocol execution, not general rules that one commitment should always create another.

2.4 Automata-based approaches

Spoletini and Verdicchio [24] developed an automata-based technique for monitoring commitments expressed in a propositional temporal logic with both past and future operators. The monitoring problem is modelled as a word recognition problem over an alphabet comprising propositions representing the contents of "sniffed" agent messages and the values of past-oriented subformulae of the formula to be monitored. The formula is preprocessed using Gabbay's rules [18] to separate out any future operators nested within past operators. The values of subformulae formed from past operators with no nested future operators are recognised dynamically by deterministic Büchi automata, and these subformulae are replaced by special propositions representing the outputs of the automata. The resulting formula is then translated into an alternating modulo counting automata. In this approach, fulfilment and violation are represented by the operational condition of the automaton reaching an acceptance or non-acceptance state—there is no representation of fulfilment or violation within the language used for representing commitments.

Modgil et al. [22] model norms with augmented transition networks (ATNs), comprising three states representing the norm being inactive, active and either fulfilled or violated. ATNs are processed via an architecture in which observer agents send messages to monitors, which trigger transitions in the ATNs and notify a manager agent of norm fulfilments and violations. The norms could, in principle, include messages announcing fulfilments and violations in arc labels, with the manager having the responsibility of sending these, but this extension is not proposed in the paper. The approach is defined in terms of a highly procedural account of the architecture and the interaction between its components, and it is difficult to relate it to more declarative approaches.

3 A temporal logic of expectation

The logic used in this paper is based on an extension of propositional linear temporal logic proposed by Cranefield and Winikoff [11]. However, in this paper we introduce some changes[4] from the original presentation and omit some features of the language

[4] The syntax of the previous version of the logic did not include the four-argument versions of Exp, Fulf, Viol nor the Trunc_S and Progress operators. However, these operators were defined semantically and used in the definitions of the two-argument versions of Exp, Fulf and Viol (which we have renamed here from their original names ExistsExp, ExistsFulf and ExistsViol). Here, to allow a concise presentation, we include all these operators in the syntax.

that are not relevant to the discussion in this paper. The syntax of the logic is described by the following grammar:

$$\begin{aligned}\phi ::= {} & \mathsf{Exp}(\phi,\phi,n,\phi) \mid \mathsf{Fulf}(\phi,\phi,n,\phi) \mid \mathsf{Viol}(\phi,\phi,n,\phi) \mid \\ & \mathsf{Exp}(\phi,\phi) \mid \mathsf{Fulf}(\phi,\phi) \mid \mathsf{Viol}(\phi,\phi) \mid \\ & p \mid \neg\phi \mid \phi \wedge \phi \mid \bigcirc\phi \mid \ominus\phi \mid \phi \,\mathsf{U}\, \phi; \mid \phi \,\mathsf{S}\, \phi \mid n \mid \mathsf{Trunc}_{\mathsf{S}} \mid \mathsf{Progress}(\phi,\phi)\end{aligned}$$

where p is a proposition, $\bigcirc$ is the standard temporal "next" operator, $\ominus$ is the standard temporal "previous" operator, U is the standard temporal "until", S ("since") is a backwards-looking version of until, and n is a nominal: a proposition that is constrained to be true in exactly one state in the model. We assume that the model contains at least one nominal for each state, as these are used in the semantics to identify the states in which "rules of expectation" fire and introduce new expectations. Nominals are a feature of hybrid logic [7], and the original version of the logic [11] contained other hybrid logic constructs. However, only nominals are needed in this paper. The $\mathsf{Trunc}_{\mathsf{S}}$ and Progress operators are explained below.

We assume the propositions include $\top$ (true) and $\bot$ (false), with their usual meanings, and define as abbreviations the Boolean connectives $\vee$ and $\rightarrow$, the derived temporal operators "eventually ϕ" ($\Diamond\phi \equiv \top \,\mathsf{U}\, \phi$), and "always ϕ" ($\Box\phi \equiv \neg\Diamond\neg\phi$), and similar backwards-looking versions $\diamondminus\phi \equiv \top \,\mathsf{S}\, \phi$ and $\boxminus\phi \equiv \neg\diamondminus\neg\phi$.

The semantics determine the truth of a formulae at a given state in a model comprising a finite or infinite sequence of states together with a valuation function specifying the propositions that hold in each state. In the case of a finite model, either *strong* or *weak* semantics can be used to evaluate the $\bigcirc$ and U operators [14]. The strong semantics assume a formula is false if the model does not include enough states to evaluate a formula, while the weak semantics assume a formula is true in this situation. Thus, in the final state of a finite model, $\bigcirc p$ is false under the strong semantics and true under the weak semantics. The operator $\mathsf{Trunc}_{\mathsf{S}}$ is a simplified form of an operator defined by Eisner et al. [14], and its semantics truncate the model at the current state and use the strong semantics to evaluate its argument formula. Essentially this means to determine whether the argument formula can be known to be true without using any information in future states. Formally, $\mathsf{Trunc}_{\mathsf{S}}\,\phi$ is true in state i of a model $\mathcal{M}$ if and only if ϕ is strongly true ($\models^{+}$) in a truncated model $\mathcal{M}^i$ where all states after i have been removed:

$$\mathcal{M}, i \models \mathsf{Trunc}_{\mathsf{S}}\,\phi \quad \text{iff} \quad \mathcal{M}^i, i \models^{+} \phi$$

3.1 Expectation Operators

The first two arguments, of the Exp, Fulf and Viol operators represent a conditional rule of expectation. Although the condition and expectation of a rule always appear as separate arguments of an operator in our logic, for convenience we will write $\lambda \Rrightarrow \rho$ as shorthand[5] for "the rule given by the pair λ and ρ". The meaning of a rule $\lambda \Rrightarrow \rho$ is that if λ evaluates to *true* in any state, given the information in the model up to that state, then ρ is an expected constraint on the model at that state.

[5] Note that '$\Rrightarrow$' does not represent logical implication and is not formally part of our language.

Unlike most approaches to modelling norms and commitments, our expectations are not limited to propositions that describe a desired property of a single state (e.g. the performance of a given action by an agent) in conjunction with a simple deadline constraint. Instead we aim to study the fulfilment and violation of more general types of expectation, such as those that aren't brought about by agents' actions ("The sun will rise each morning") and those with complex temporal structure ("If I pay for a subscription then the publisher will send me a magazine issue each month for a year from the month after my payment is received"). Thus, λ and ρ can be any formula in our logic, although the semantics ensure that the rule can only fire if the condition λ can be evaluated without the use of information from future states [6]. The expectation ρ can be a formula expressing desired properties of the states up to the present and/or a constraint on the future sequence of states that should be monitored for fulfilment or violation.

A formula $\mathsf{Exp}(\lambda, \rho, n, \phi)$ means that the formula ϕ is an active expectation as a result of the rule $\lambda \Rrightarrow \rho$ having fired (i.e. its condition λ becoming true) in a (possibly prior) state specified by nominal n. If the rule fired in a prior state, but the expectation was not immediately fulfilled or violated, then the current form of the expectation ϕ may be different from the expectation ρ in the rule due to the use of *formula progression* (explained below) to carry forward an expectation from one state to the next.

The operators $\mathsf{Fulf}(\lambda, \rho, n, \phi)$ and $\mathsf{Viol}(\lambda, \rho, n, \phi)$ have the same argument structure as Exp, and mean that the rule $\lambda \Rrightarrow \rho$ firing in the state specified by n has resulted in an active expectation ϕ that is fulfilled or (respectively) violated in the current state. These three operators are defined as follows (where n is a nominal):

$$\begin{aligned}
\mathsf{Exp}(\lambda, \rho, n, \phi) \iff & (n \wedge \mathsf{Trunc_S}\ \lambda \wedge \phi = \rho) \vee \\
& \exists \psi \ominus (\mathsf{Exp}(\lambda, \rho, n, \psi) \wedge \\
& \quad \neg \mathsf{Trunc_S}\ \psi \wedge \neg \mathsf{Trunc_S}\ \neg\psi \wedge \mathsf{Progress}(\psi, \phi)) \\
\mathsf{Fulf}(\lambda, \rho, n, \phi) \iff & \mathsf{Exp}(\lambda, \rho, n, \phi) \wedge \mathsf{Trunc_S}\ \phi \\
\mathsf{Viol}(\lambda, \rho, n, \phi) \iff & \mathsf{Exp}(\lambda, \rho, n, \phi) \wedge \mathsf{Trunc_S}\ \neg\phi
\end{aligned}$$

The definition of Exp states that there are two ways for an expectation to result from a rule $\lambda \Rrightarrow \rho$: either λ holds in the current state (without recourse to future information) and therefore ρ is now expected (i.e. it is an expected constraint on the model), or some other formula ψ was expected in the previous state as a result of the rule, ψ was not known to be true or false in that state given the model up to that point, and thus a "progressed" form of ψ is now expected. $\mathsf{Progress}$ is a temporal operator corresponding to the progression function defined by Bacchus and Kabanza [4] for planning with "temporally extended goals". Details are beyond the scope of this paper, but essentially, progression transforms a temporal formula from the viewpoint of one state into the viewpoint of the next state. A formula that can be determined to be true (respectively false) without recourse to any future states progresses to $\top$ (respectively $\bot$). A formula that requires future information in order to be fully evaluated is partially evaluated using information from the model up to the current state and is then re-expressed

[6] Future states might be available in offline monitoring of expectations, such as the examination of an audit trail.

as an equivalent constraint in the context of the next state. For example, if p holds in the current state, then $p \wedge \bigcirc q$ progresses to q, expressed as $\mathsf{Progress}(p \wedge \bigcirc q, q)$.

The Exp, Fulf and Viol operators defined above are rather specific in the information they express about a currently active, fulfilled or violated expectation: the third and fourth arguments record the state in which the rule's condition became true and the current form of the expectation. In many cases, it may be sufficient to know there is currently an active, fulfilled or violated expectation resulting from a given rule. We therefore overload these operators and define alternative versions in which the last two arguments are omitted due to an implicit existential quantification:

$$\mathsf{Exp}(\lambda, \rho) \iff \exists n,\phi\, \mathsf{Exp}(\lambda, \rho, n, \phi)$$
$$\mathsf{Fulf}(\lambda, \rho) \iff \exists n,\phi\, \mathsf{Fulf}(\lambda, \rho, n, \phi)$$
$$\mathsf{Viol}(\lambda, \rho) \iff \exists n,\phi\, \mathsf{Viol}(\lambda, \rho, n, \phi)$$

Using these operators and the model checker described previously [11] we can now analyse an observed execution trace to check for the activation, fulfilment or violation of expressive temporal rules of expectation, or, as special cases, more restricted representations used in prior work. For example, fulfilment of an obligation $O(\rho \leq \delta)$, stating that condition ρ must be brought about before deadline proposition δ becomes true [10], can be represented as $\mathsf{Fulf}(\top, \neg\delta\, \mathsf{U}\, (\rho \wedge \neg\delta)))$.

Note that we could also introduce additional versions of the Exp, Fulf and Viol operators that existentially quantify over only n or only ϕ; however in the remainder of this paper we will focus on the two-argument versions of the operators.

3.2 Nesting Expectation Operators

In the previous account of the logic, the Exp, Fulf and Viol operators could not contain nested occurrences of these operators. In this paper we allow this nesting, and explain why the previous restriction was unnecessary.

It follows from the definitions given above that the truth of the two versions of the Exp, Fulf and Viol formula do not depend on any future states in the model. This is because they depend only on the truth of a nominal (a special type of proposition) in the current state, Exp and Progress formulae in the prior state, and formulae prefixed by the $\mathsf{Trunc}_\mathsf{S}$ operator in the current and prior states. Formula progression, by definition, does not depend on future states, and the $\mathsf{Trunc}_\mathsf{S}$ operator eliminates them from consideration. Therefore it is meaningful for Exp, Fulf and Viol operators to appear within a condition of a rule (the first argument) inside one of these operators—the use of the $\mathsf{Trunc}_\mathsf{S}$ operator to evaluate rule conditions will work correctly. For example, suppose that a library application has the rule of expectation $book\_borrowed \Rrightarrow \bigcirc book\_returned$ (where each state represents a day). Suppose that we also have a contrary-to-duty rule of the form $\mathsf{Viol}(book\_borrowed, \bigcirc book\_returned) \Rrightarrow fine$, indicating that failure to return a book on time results in a fine (or, more precisely, in the expectation that a fine be imposed). In order to evaluate the formula $\mathsf{Exp}(\mathsf{Viol}(book\_borrowed, \bigcirc book\_returned), fine)$ we need to check whether in the current or any previous state a book was borrowed, and whether this book was returned on time or not. The key point is that in order

to evaluate the nested Viol formula, we never need to consult any future timepoints, due to the use of TruncS in the semantics of Exp and Viol.

Additionally, to appear as the expectation of a rule (the second argument) within one of these operators, a formula must be able to be progressed when required—see the second line of the definition of the four-argument Exp operator. The axioms defining the progression operation that were defined previously [11] include the following base cases[7] (adapted slightly here for simplicity of presentation):

$$\mathcal{M}, i \models \mathsf{Progress}(\phi, \top) \text{ if } \mathcal{M}^i, i \models^+ \phi$$
$$\mathcal{M}, i \models \mathsf{Progress}(\phi, \bot) \text{ if } \mathcal{M}^i, i \models^+ \neg\phi$$

where $\mathcal{M}$ is a model, i is the index of a state in the model, and $\mathcal{M}^i$ denotes the model with all states after index i removed.

As Exp, Fulf and Viol can be evaluated without using any future states in the model, then one of the two base cases above will apply, and formulae having these operators as their principal functor will progress to either $\top$ or $\bot$. Therefore, these operators can also appear nested within the second arguments of these three types of formula and the restriction on nesting Exp, Fulf and Viol imposed in our previous work is unnecessary. Finally, the model checking process described in our earlier work [11] can be easily extended to apply to nested expectations, and we have extended our tool to be able to do so (as we will demonstrate in Section 4.1).

4 Use cases for nested expectation operators

In the previous section we argued that the restriction in previous work which did not allow expectations to be nested was unnecessary, and that the semantics of nested expectations are well defined and unproblematic. We also argued that checking whether nested expectations hold, are fulfilled or are violated, can be easily done within the existing framework and tool [11].

In this section we argue that allowing for nested expectations allows for a range of scenarios to be easily specified. Since we are making the case that nested expectations provide additional expressivity that is useful in a broad range of cases, we provide a number of different use cases in which nested expectations are used to specify desired normative behaviour. Space limitations prevent us from developing each of the scenarios in detail, but the aim is not to provide details on any given case, but, rather, to argue that a wide range of scenarios exists where there is a benefit from allowing nested expectations in a declarative way.

Chained expectations. One use case scenario for nested expectations is to allow for causality relationships between expectations to be captured. In this case, we may want to specify that a certain expectation ω exists when some other expectation has been fulfilled. We can express this as follows:

$$\mathsf{Fulf}(\phi, \psi) \Rrightarrow \omega$$

[7] Other axioms for Progress (not shown here) define $\mathsf{Progress}(\phi, \psi)$ compositionally based on the principal functor of ϕ and involve recursive progression of the top-level subformulae of ϕ.

In other words, once rule $\phi \Rrightarrow \psi$ is fulfilled, ω is expected.
This sequential fulfilment of expectations could arise when the two commitments must be fulfilled in a certain order due to one setting up the conditions for the other to be attempted, or when an agent's responsibilities are escalated as a result of successful performance.

Fulfilment ends probationary period. Another scenario, which is complementary to the one above, is that an expectation ω exists (only) until another expectation is fulfilled:

$$\top \Rrightarrow \omega \ \mathsf{W}\, \mathsf{Fulf}(\phi, \psi)$$

where W is the "weak until" operator: $\alpha \mathsf{W} \beta \equiv (\alpha \mathsf{U} \beta) \vee \Box \alpha$. In other words, ω is (unconditionally) expected until rule $\phi \Rrightarrow \psi$ is fulfilled, or, if the rule is never fulfilled, it is always expected.
This encodes the situation where some condition applies (e.g. limited access to resources) until an agent ends a probationary period by fulfilling a certain expectation (such as passing a test).

"Contrary to duty" expectation or expectation to act on violation. Whereas the previous two cases dealt with the fulfilment of expectations, and how fulfilment may specify the termination or creation of another expectation, this rule deals with violation, and how it may result in the creation of another expectation:

$$\mathsf{Viol}(\phi, \psi) \Rrightarrow \omega$$

In other words, when rule $\phi \Rrightarrow \psi$ is *violated*, ω is expected.
This type of rule represents the well known concept of a contrary to duty expectation: if one expectation is violated an alternative expectation is created. If ω involves a different agent to the one that violated the first expectation, this can represent the requirement for that agent to respond to the violation. A concrete example of this case that we discussed earlier is the expectation that a fine be imposed should a library book not be returned on time.

Expectation handling priority. The next few scenarios show how nested expectations can be used to specify constraints on the *timing* of expectations. For instance, the following rule expresses a priority between two expectations:

$$\mathsf{Fulf}(\phi, \psi) \Rrightarrow \mathsf{Fulf}(\lambda, \rho)$$

In other words, when rule $\phi \Rrightarrow \psi$ is fulfilled, rule $\lambda \Rrightarrow \rho$ should have been fulfilled already. This could be used to express a policy for placing a priority on the order of fulfilment of rules.

Just-in-time expectation management. The following form of rule could be used to encode a policy that resources for fulfilling a given expectation are made available the moment that expectation becomes active.

$$\mathsf{Exp}(\phi, \psi) \Rrightarrow \omega$$

In other words, once rule $\phi \Rrightarrow \psi$ is triggered, ω is expected.

Delaying rule activation. This, and the next example, deal with constraints on the timing of an expectation relative to an arbitrary condition ω. The following rule expresses the policy to avoid the conditions that trigger an expectation until appropriate resources are in place for fulfilling it ("ω").

$$\top \Rrightarrow (\neg \mathsf{Exp}(\phi, \psi)) \,\mathsf{U}\, \omega$$

In other words, avoid triggering the rule $\phi \Rrightarrow \psi$ until ω is true.

Delaying rule fulfilment. Similar to the previous example, this is a constraint on the timing of an expectation relative to ω, but here we are specifying that the agent should not *fulfil* the expectation until ω. This may be desirable if, for example, an agent has a policy to not be over-diligent in fulfilling an expectation, e.g. it might only pay bills on the last possible day for payment.

$$\top \Rrightarrow (\neg \mathsf{Fulf}(\phi, \psi)) \,\mathsf{W}\, \omega$$

In other words, avoid fulfilling the rule $\phi \Rrightarrow \psi$ until (and only if[8]) ω becomes true.

Avoid violation between two states. Finally, this and the subsequent scenario deal with constraints over a time *interval*. Given two nominals, n_1 and n_2 we can specify that in the interval defined by the two end points a given condition must hold. For example, we may require that within a designated time interval a certain expectation should not be violated:

$$n_1 \Rrightarrow (\neg \mathsf{Viol}(\phi, \psi)) \,\mathsf{U}\, n_2$$

In other words, avoid violating the rule $\phi \Rrightarrow \psi$ between the states referenced by the nominals n_1 (inclusive) and n_2 (exclusive).
This example may be used in situations where an agent may be willing to risk violation of an expectation, but not during certain periods (e.g. when the boss is in the office).

Fulfil a rule sometime between two states. Similarly to the previous scenario, within a given time interval we can specify a condition, in this case that something (such as an expectation being fulfilled) *should* happen:

$$n_1 \Rrightarrow \neg n_2 \,\mathsf{U}\, (\neg n_2 \wedge \mathsf{Fulf}(\phi, \psi))$$

In other words, the rule $\phi \Rrightarrow \psi$ must be fulfilled sometime between the states referenced by the nominals n_1 (inclusive) and n_2 (exclusive). The formalisation can be paraphrased as once n_1 has occurred, n_2 cannot occur until after $\mathsf{Fulf}(\phi, \psi)$.
This example may be used in situations where an agent subject to an expectation may adopt the policy of fulfilling it in a more restricted period than was originally required, e.g. while the boss in the office and able to directly observe the fulfilment.

[8] As before, W is the "weak until" operator: $\alpha \mathsf{W} \beta \equiv (\alpha \mathsf{U} \beta) \vee \Box \alpha$.

4.1 Model checking example

In this section we briefly illustrate that the model checker, extended with the ability to progress expectation modalities, behaves as expected. We use the last use case above as an example. Consider the rule $p \Rrightarrow \Diamond q$ (once p holds, q is expected to hold eventually). We can encode the property that this rule will be fulfilled using our Python-based model checker as follows, where formulae are written in prefix form as nested tuples.

```
f = Formula(('Fulf', 'p', ('U', True, 'q')))
```

Now consider a model with four states $s_0, \ldots, s_3$ in which p holds in state s_1 and q holds in state s_3. We encode this as follows, where the `4` is the number of states in the model, and the second argument maps each proposition to a list of indices of states in which it holds. For instance, `'p':{1}` indicates that proposition `p` holds in state 1 (i.e. the second state, since the first state has index 0).

```
m = Model(4,{'p':{1}, 'q':{3}})
```

After invoking the model checker on this formula we can examine the value of property `f.labels` to find that the fulfilment formula `f` is only satisfied in state s_3 (we simplify the data structure to suppress details of the label structure not relevant to this paper):

```
{0: False, 1: False, 2: False, 3: True}
```

We now modify the formula so that our policy is to only fulfil the original rule from s_1 onwards and before s_3:

```
f = Formula(('Fulf', 's1',
               ('U', ('not', 's3'),
                 ('and', ('not', 's3'),
                   ('Fulf', 'p', ('U', True, 'q'))))))
```

This definition for `f` is simply the Python-based encoding of the formula:

$$\mathsf{Fulf}(s_1, \neg s_3 \,\mathsf{U}\, (\neg s_3 \wedge \mathsf{Fulf}(p, \Diamond q)))$$

which is the last use case scenario ("Fulfil a rule sometime between two states").

Invoking the labelling function again and examining `f.labels` we now find that the formula is false everywhere:

```
{0: False, 1: False, 2: False, 3: False}
```

This is the expected result as the new formula can only possibly be satisfied in states s_1 and s_2, and the original formula does not hold in those states.

5 Conclusion

In this paper we have considered the ability of approaches for modelling and monitoring techniques various sorts of expectations to represent the existence, fulfilment and violation of expectations as first-class entities, and to allow these to appear nested within

rules of expectation. Having found no work in the literature that fully meets these needs, we demonstrated how our existing approach to modelling and monitoring expectations extends readily to address this issue. We listed some use cases to show that the expectations expressible using this new modelling ability are of interest in practical settings.

At present our focus is on passive detection of expectation creation, fulfilment and violation (i.e. monitoring). However, as our use case included several examples of policies that agents might adopt, we need to investigate ways in which agents informed by these expectations can understand them and proactively adjust their behaviour to fulfil such policies. Specifically, an agent could use the model checker to reason about hypothetical extensions of the history so far. Given a history H, and an agent who is considering either action A_1 (resulting in state S_1) or action A_2 (resulting in state S_2), then we could use the model checker to label the extended history $H \oplus S_1$ (where "$\oplus$" denotes sequence concatenation) and the extended history $H \oplus S_2$, and use the results to guide decision making by the agent. More generally, an agent might realise via analysing traces that its current expectation doesn't meet some soft constraint (e.g. getting praised by the boss), and therefore follow a heuristic (or apply some reasoning technique) to restrict the period in which it aims to satisfy the expectation (e.g. so that the boss will witness it). To allow this we must also extend our framework to allow the content of expectations to identify which agents are responsible for particular expectations. Finally, the fulfilment operator (and other operators) could be extended to allow not just a single rule to be given as an argument, but a *set* of rules.

References

1. Alberti, M., Gavanelli, M., Lamma, E., Chesani, F., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based software tool. Applied Artificial Intelligence 20(2), 133–157 (2006)
2. Aldewereld, H., Álvarez-Napagao, S., Dignum, F., Vázquez-Salceda, J.: Making norms concrete. In: Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems. pp. 807–814. IFAAMAS (2010)
3. Artikis, A., Sergot, M.: Executable specification of open multi-agent systems. Logic Journal of the IGPL 18(1), 31–65 (2009)
4. Bacchus, F., Kabanza, F.: Using temporal logics to express search control knowledge for planning. Artificial Intelligence 116(1-2), 123–191 (2000)
5. Bentahar, J., Moulin, B., Chaib-draa, B.: Commitment and argument network: a new formalism for agent communication. In: Dignum, F. (ed.) Advances in Agent Communication, LNCS, vol. 2922, pp. 146–165. Springer (2004)
6. Bentahar, J., Moulin, B., Meyer, J.J.C., Lespérance, Y.: A new logical semantics for agent communication. In: Inoue, K., Satoh, K., Toni, F. (eds.) Computational Logic in Multi-Agent Systems, LNCS, vol. 4371, pp. 151–170. Springer (2007)
7. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
8. Boella, G., van der Torre, L.: Security policies for sharing knowledge in virtual communities. IEEE Transactions on Systems, Man and Cybernetics, Part A 36(3), 439–450 (2006)
9. Boella, G., van der Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. Autonomous Agents and Multi-Agent Systems 17(1), 1–10 (2008)
10. Broersen, J., Dignum, F., Dignum, V., , Meyer, J.J.C.: Designing a deontic logic of deadlines. In: Deontic Logic in Computer Science, LNAI, vol. 3065. Springer (2004)

11. Cranefield, S., Winikoff, M.: Verifying social expectations by model checking truncated paths. Journal of Logic and Computation (2010), advance access, doi: 10.1093/logcom/exq055
12. Dignum, F., Meyer, J.J.C., Wieringa, R.: A dynamic logic for reasoning about sub-ideal states. In: ECAI Workshop on Artificial Normative Reasoning. pp. 79–92 (1994)
13. Dignum, F., Weigand, H., Verharen, E.: Meeting the deadline: On the formal specification of temporal deontic constraints. In: Foundations of Intelligent Systems, LNAI, vol. 1079, pp. 243–252. Springer (1996)
14. Eisner, C., Fisman, D., Havlicek, J., Lustig, Y., McIsaac, A., Campenhout, D.V.: Reasoning with temporal logic on truncated paths. In: Computer Aided Verification, LNCS, vol. 2725, pp. 27–39. Springer (2003)
15. Esteva, M., Rosell, B., Rodríguez-Aguilar, J.A., Arcos, J.L.: AMELI: An agent-based middleware for electronic institutions. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems. vol. 1, pp. 236–243. IEEE Computer Society (2004)
16. Farrell, A.D.H., Sergot, M.J., Sallé, M., Bartolini, C.: Using the event calculus for tracking the normative state of contracts. International Journal of Cooperative Information Systems 14(2 & 3), 99–129 (2005)
17. Fornara, N., Viganò, F., Colombetti, M.: Agent communication and artificial institutions. Autonomous Agents and Multi-Agent Systems 14(2), 121–142 (2007)
18. Gabbay, D.M.: The declarative past and imperative future: Executable temporal logic for interactive systems. In: Temporal Logic in Specification, LNCS, vol. 398, pp. 409–448. Springer (1989)
19. García-Camino, A., Noriega, P., Rodríguez-Aguilar, J.A.: Implementing norms in electronic institutions. In: Proceedings of the 4nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005). pp. 667–673. ACM Press (2005)
20. García-Camino, A., Rodríguez-Aguilar, J.A., Sierra, C., Vasconcelos, W.W.: Constraint rule-based programming of norms for electronic institutions. Autonomous Agents and Multi-Agent Systems 18(1), 186–217 (2009)
21. Governatori, G., Rotolo, A.: How do agents comply with norms? Dagstuhl Seminar Proceedings 09121, http://drops.dagstuhl.de/opus/volltexte/2009/1909 (2009)
22. Modgil, S., Faci, N., Meneguzzi, F., Oren, N., Miles, S., Luck, M.: A framework for monitoring agent-based normative systems. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems. pp. 153–160. IFAAMAS, Richland, SC (2009)
23. Singh, M.P.: Semantical considerations on dialectical and practical commitments. In: Cohn, A. (ed.) Proceedings of the 23rd National Conference on Artificial Intelligence. vol. 1, pp. 176–181. AAAI Press (2008)
24. Spoletini, P., Verdicchio, M.: An automata-based monitoring technique for commitment-based multi-agent systems. In: Coordination, Organizations, Institutions and Norms in Agent Systems IV, LNAI, vol. 5428, pp. 172–187. Springer (2009)
25. Vázquez-Salceda, J.: The role of norms and electronic institutions in multi-agent systems applied to complex domains: The HARMONIA framework. AI Communications 16(3), 209–212 (2003)
26. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing norms in multiagent systems. In: Multiagent System Technologies, LNCS, vol. 3187, pp. 313–327. Springer (2004)
27. Verdicchio, M., Colombetti, M.: Communication languages for multiagent systems. Computational Intelligence 25(2), 136–159 (2009)
28. Yolum, P., Singh, M.P.: Commitment machines. In: Meyer, J.J.C., Tambe, M. (eds.) Intelligent Agents VIII, LNCS, vol. 2333, pp. 235–247. Springer (2002)

On the Analysis and Implementation of Normative Systems – Towards a Methodology

Andrew Jones[1,2], Jeremy Pitt[2], Alexander Artikis[2,3]

[1]Department of Informatics, King's College London, UK
[2]Department of Electrical & Electronic Engineering,
Imperial College London, UK
[3]Institute of Informatics & Telecommunications,
National Centre for Scientific Research "Demokritos", Athens 15310, Greece

Abstract. This paper outlines a methodology for the design of norm-governed multi-agent systems, emphasising the importance of a formally articulated characterisation of the principal normative concepts, the adequacy of which should be assessed independently of considerations of computational tractability and implementation. We use the concept of institutionalised power as the main example motivating our methodological recommendations.

Keywords: Institutionalised Power, Constitutive Norms, Synthetic Method, Communication.

1 Introduction

This paper is concerned with outlining a methodology for the analysis, design and implementation of norm-governed multi-agent systems, emphasising the importance of a formally articulated characterisation of the principal normative concepts, the adequacy of which should be assessed independently of considerations of computational tractability and implementation.

The paper is organised in the following way: in Section 2 we use the concept of institutionalised power as an illustrative example of the complexity of normative notions, with a view to motivating a methodology for the design of norm-governed multi-agent systems (MAS); we then outline that methodology, paying particular attention to what we choose to call the *middle-layer*: a formal model of the target domain, capable of articulating key conceptual distinctions, which will themselves inform the construction of lower-layer models that seek to provide a basis for system implementation. In Section 3 we then propose a generalisation of the methodology, expressed in terms of an extension of the Adapted Synthetic Method [1]. In Section 4 we provide two illustrative examples, both of which are concerned with the analysis of institutionalised power, and both of which serve to underline the importance for MAS design of a clearly articulated middle-layer conceptualisation. Section 5 concludes the paper with some indications of further lines of enquiry, and some conjectures about how our methodological proposals, if adopted, should influence the way in which prospective MAS researchers are trained.

2 Motivating the Approach

To motivate our approach, we discuss primarily the notion of institutionalised power – although a number of other central normative concepts might just as well have been employed for this purpose – and we then use that discussion as a point of departure for outlining the proposed methodology.

2.1 Institutionalised Power

Any reasonably comprehensive model, formal or informal, of norm-governed multi-agent systems must be able to accommodate norms pertaining to institutionalised power, in addition to those that express obligations and permissions. It is a commonplace feature of organisations that particular agents – be they individual or collective – are empowered to carry out actions the consequences of which have a significant bearing on the way the organisation is governed or administered, and thus on the way agents interact. So, for instance, given agents may be empowered to bring about particular types of normatively significant states of affairs, as when relevant public officials are empowered to create a state of marriage between two individuals, or to validate a will, or to appoint some other individuals to particular roles (including roles that themselves involve the exercise of institutionalised powers), or to create new obligations and permissions, or to delete existing norms. These are among the many examples that can be given to illustrate the prevalence and importance of institutionalised power in the day-to-day conduct of social life.

It is natural to see the concept of institutionalised power as belonging to the more general theory of rights. Indeed we often use the terms entitlement or right in describing the situation of empowered agents: the Database Administrator has the right to (is entitled to) grant access to the database but perhaps no other staff member has that right. In their work on norm-governed multi-agent systems, computer scientists would do well to keep in mind that that the theory of rights has a very long history in Social, Moral and Legal Philosophy, and that the vast body of literature in this area provides ample evidence of the complexity of the language of rights, as it is commonly employed in ordinary, everyday sets of rules and regulations.

By way of a fairly simple illustration consider the following sentence:

– The Head of Department can sign off staff claims for expenses.

In that sentence, the modal auxiliary 'can' may be given at least three distinct interpretations. The sentence might express the idea that the HoD is empowered to sign off such claims – the *can* of empowerment. Or it might express the HoD's permission to sign the claims the *can* of permission. Or, thirdly, it might express the fact that the HoD has the physical ability required to carry out the act of signing – the *can* of practical possibility. Nearly one hundred years ago this threefold ambiguity was recognised by Hohfeld [2], in work that was to provide the foundation for the modern formal-logical theory of rights

stemming from Kanger [3, 4]. And note that the three interpretations are indeed distinguishable: our HoD might be empowered but physically incapacitated, or physically capable but not empowered; similarly, the HoD might be permitted but physically incapacitated, or physically capable but not permitted. Furthermore, we may imagine circumstances in which the Dean of Faculty, say, just to humour the HoD, permits him to sign off claims, even though his signature does not validate them. Finally, for an instance of empowerment without permission, consider another example, from a situation that arose recently at King's College London: having been empowered by the UK Government to grant its own degrees, independently of the University of London, the College was nevertheless not allowed to exercise that power until such time as its own Academic Board had agreed to permit that exercise.

The ambiguity of 'can' is frequently also present in everyday expressions of norms and rules that employ the term 'authorised'. For instance, 'Only the Head of Department is authorised to assign teaching duties' would reasonably be understood to pertain to a power held only by the HoD, whereas 'The Head of Department is authorised to park his car in the quad' presumably means, essentially, that he is permitted to park there although it may also imply that relevant others (e.g., the gatekeeper) are obliged to let him drive into the quad for the purpose of parking his car. While there is good reason to suppose that a number of types of rights can be represented, in a formal-logical theory, by means of appropriate combinations of deontic modalities for obligation and permission with action modalities (see the work of Kanger referred to above, and, e.g., [5, 6]), the points just made about the interpretation of 'can' support the contention that this formal language will have to be further enriched if it is also to be able to capture rights of type empowerment. A key observation here, well-known from the work of Searle [7] among others, is that power-conferring norms standardly take the form of constitutive rules, typically expressed using 'counts-as conditionals' which indicate the means by which specific types of new situation may be brought into being. (In contrast to rules of the regulative type, which typically employ such modalities as 'ought', 'must', 'shall', 'may', to express norms that govern pre-existing practices.) So, for instance, the registrar's uttering of a particular sentence ("I pronounce you man and wife"), in a particular context (e.g., there must be witnesses present), counts (in a particular institution, e.g., a particular legal system) as a means of creating a state of marriage. Similarly, the signing of the expenses claim by the HoD counts as a means of creating a valid claim. The first attempt to provide a formal-logical analysis of the 'counts-as' conditional was [8], in which a modal conditional logic was employed. Several other theories followed, and these are well documented and analysed in the doctoral thesis of Davide Grossi [9]. (A recently completed handbook chapter [10] provides a critical summary and comparison of the various approaches, together with an indication of some yet-to-be-resolved issues.)

2.2 Towards a Methodology

Our aim here, however, is not to offer any further contribution to the formal theory of rights and powers, but – having now emphasised that ambiguities and nuances are commonplace, and that, accordingly, work on norm-governed multi-agent systems must be properly informed of them – we proceed to outline a methodology aimed at facilitating the combination of conceptual sensitivity with the demands of practical applications.

In essence, we wish to point to the need for a formal theory of the central normative notions, including rights, powers, obligations and permissions, and relevant action-theoretic concepts, sufficiently rich to provide a means of articulating significant ambiguities and complexities of interpretation of the types alluded to above. Preferably, this formal theory should be expressed in a logical language, with a properly specified semantics, to enable testing for relations of implication and consistency in sets of norms. Our central proposal is that a formal theory of this kind should form a middle layer between, on the upper level, statements of norms and policies expressed in natural language and, on the lower levels, formalisms employed in the actual design and implementation of norm-governed multi-agent systems. We emphasise that the function of the middle-layer theory is to provide an analytic tool for the expression of conceptual distinctions; considerations of computational tractability would here be irrelevant. It is when one moves down from the middle layer and into actual system design that computational considerations of course begin to come into play. (Of course, we do not rule out the possibility that a good, middle-layer theory might itself also be computationally tractable. Our point is, rather, that the criterion of adequacy that middle-layer theories must meet is that they serve as clear, conceptual-analytical tools.)

We suppose, further, that the construction of the lower layers would be very likely to involve simplifications. Not all of the nuances captured at the middle layer are likely to be of import for a given set of tasks, i.e., for a specific set of system requirements. (For instance, it may be that the distinction between permission and empowerment might be safely ignored for a particular practical application.) But one of the clear advantages of having a rich middle-layer theory would then be that the system designer has a much clearer picture of the simplifying assumptions he is making than he would have were he to move directly from natural-language policy formulations towards the level of implementation; a properly developed middle-layer theory will significantly reduce the risk of misinterpretation and over-simplification. (Below we discuss an example in which, we claim, these risks are clearly exhibited. The examples will show how – in the absence of a suitably rich conceptual analysis – things can so easily go wrong.)

A further observation should also be made about the envisaged relationship between the upper, natural-language layer and the middle-level theory. Ideally, one wants to be able to construct the conceptual theory in such a way that it is relatively easy - even for one not trained in its underlying formal semantics - to determine how to formally represent a given norm and, more particularly, how to

determine which formal structures should be chosen as the appropriate, distinct formalisations of an ambiguous natural-language sentence. Our own teaching experience provides some reason to believe that a multi-modal formal language, containing the basic 'building-block' modalities for obligation, permission, action, attempted action and the counts-as conditional connective [11], can be relatively easily learnt by students not familiar with the semantics of modal logic, to a level of proficiency at which the student can correctly formalise and formally disambiguate the modal aspects of natural-language policy statements. Furthermore, having first generated a set of formally-expressed normative positions, it is possible to move 'back up' from the logical forms towards more-or-less natural-language expressions, whilst preserving the logical structures encapsulated in the formal-logical statements themselves. It does not require training in the underlying formal-logical techniques to be able to decide on an appropriate fully natural-language rendering of these stilted, 'upper-middle level' representations. For an example applying a multi-modal logic in this kind of way, see [12], which examines some norms regulating the flow of information (e.g., concerning the right to be informed, the right to remain silent, etc.) from the perspective of a set of formally articulated normative positions.

In the non-modal domain, we note here also that one of the key advantages of the rule-representation techniques used in OPA (Oracle Policy Automation, currently in widespread use in, for instance, the design of knowledge-based systems for public administration), is that - although informed by an underlying logical representation - the rule-representations are 'pitched' at a level so close to natural language that it is relatively easy for the administrators – the domain experts – to determine whether the knowledge base correctly represents the administrative rules concerned.

In the next section, we make a more general methodological argument concerning the way the natural sciences and systems engineering may profitably interact in the design and development of multi-agent systems.

3 The Extended Adapted Synthetic Method

Nature has evolved mechanisms for solving some of the hardest engineering challenges. Unsurprisingly, engineers have turned to biologically-inspired ideas for possible solutions to particularly pressing challenges: for example, ant colony optimisation for network routing, immune systems for pattern classification, fungal colony optimisation for programmable networks, infant body babbling for robot learning, and so on.

This process was termed the Synthetic Method [13], where when the objective is to build an artificial system that exhibits the same capabilities as the natural system, the steps are first to construct a theory from observed phenomena, then engineer the artificial system from the theory, operate or animate the artificial system, and observe its performance. From observation the designer could justify/falsify the theory.

When the objective is to build an artificial system which provided a solution to an engineering problem rather than a testable model of a theory, [1] proposed the Adapted Synthetic Method, where the engineer creates a formal characterisation of the theory, and engineers the artificial system from the characterisation. Because information may be 'lost' in the characterisation or engineering steps, the observed performance may justify/falsify the characterisation, and does not necessarily have anything to say about the theory itself.

What we propose here is an extension of the Adapted Synthetic Method. The process of formal characterisation is the process of expressing a theory in a calculus, where by calculus we mean any system of calculation or computation based on symbolic representation and manipulation. This is the 'middle layer' indicated above, and note there may be just one calculus between the theory and the artificial system, or there may be many, depending on how computationally tractable the intervening representational calculi are, and we call the transformation between one calculus and another *principled operationalisation*.

For example, in [14] the observed phenomena are legal, social and organisational systems; the theory is a theory of norm-governed systems, and the formal characterisation is expressed in both the $\mathcal{C}+$ language [15] and the Event Calculus [16]. The $\mathcal{C}+$ language is associated with a software implementation, the Causal Calculator (CCalc), which supports a wide range of computational task which might be applied to a *specification*, for example prediction, planning and postdiction queries. For richer modeling of normative and institutional aspects, the language $n\mathcal{C}+$ [17] has been developed. An Event Calculus (EC) specification can also be used for prediction queries and animation, but a Prolog EC specification can be its own implementation, so the artificial system could be engineered using the calculus directly. Practical experience of operating such systems exposes the limitations of using the EC directly, leading to use of cached EC, and so on. In other words, the search for conceptual clarity or observation of performance lead to a modification of the 'middle layer' calculi and associated support tools, not the underlying theory. Furthermore, Sergot and colleagues have been working on a formal framework for connecting and comparing different formalisms for reasoning about action and change. This framework therefore defines a whole range of formalisms for this purpose, from the EC, to $\mathcal{C}+$, to $n\mathcal{C}+$, each formalism with is own tools, advantages and applications, and semantics-preserving transformations between different formalisms. However, it may be that in the operationalisation of a calculus, expressive power is lost. Knowing what is lost, and why, is the difference between principled operationalisation and just hacking.

In the following section, we look at some of the undesirable consequences of not defining an adequate 'middle layer' formal characterisation between the theory[1] and the artificial system.

[1] Note that the counterpart in Section 2.2 to what we are now referring to as the theory, is a natural language formulation of, for example, a set of norms, rules or policies

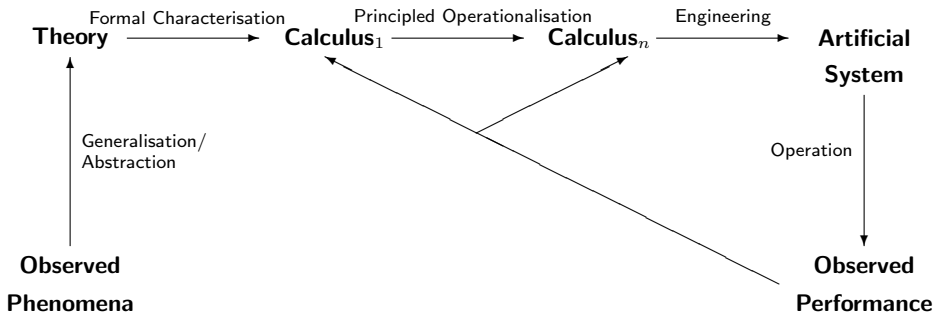


Fig. 1. Extended Adapted Synthetic Method

4 Two Illustrative Examples

We look here at two examples that serve to expose further the methodological points developed above. The first concerns a 'model' of *power* that exhibits no awareness of the complexity of the concept. The second concerns an aspect of e-government, specifically electronic voting, for which sensitivity to the ambiguity of 'institutionalised power' is undeniably essential.

4.1 "Normative Power"

Despite its title, 'A Model of Normative Power', [18] fails to provide an explicit analysis of the concept of normative power – a concept the authors associate with the power to create and/or modify norms. They characterise normative power in terms of a tuple, itself expressed in terms of first-order logic, the key element of which is called 'Mandators'. "Mandators is a set of predicates identifying agents" (op. cit., section 3.2), and furthermore "A mandator of the form $professor(x)$ means that any agent in the professor role is able to exercise the power", for instance the power to place a student under an obligation to write a conference paper – see, op. cit., section 4.1. So then the interpretation of what it means for an agent to be able to exercise a normative power is not explicated. It remains implicit in the natural-language reading the authors assign to this 'mandator' predicate.

Referring to the three-fold Hohfeldian distinction we mentioned above in illustrating the ambiguity of *can*, Oren et al. state that the notion of normative power they adopt is a combination of legal (i.e., institutionalised) power and physical power. So they do not attempt to articulate that distinction. But why not? Given the approach that they take, they could surely just introduce another 'mandator' predicate, call it 'professor1', understood in the following way: a mandator of the form $professor1(x)$ means that any agent in the 'professor1' role is able to exercise the power, at least in the sense of having the practical possibility to act as a 'mandator', but does not necessarily also have legal, institutionalised power. But, of course, introducing another predicate in this way

would throw no light on the nature of the distinction between legal and physical power, just as the original *professor* predicate leaves us none the wiser about the nature of normative power itself.

Thus the proposed model eschews any analysis of institutionalised power, whose distinctive elements were identified by Hohfeld in [2], given a formal characterisation by Jones and Sergot [8], and given a computational formulation by Artikis et al [14]. However, by conflating concepts, the resulting formalism reintroduces the ambiguities and creates confusion, not clarity or simplification, and yields a vacuous representation.

4.2 The Formal Characterisation of Voting

The need to distinguish between different types of power was clearly exemplified by the formal characterisation of enfranchisement in electronic voting systems given by [19]. Here, the notion of enfranchisement was characterised as being composed of two elements: firstly, having the right to vote; and secondly having an entitlement associated with that right.

Furthermore, having the right to vote was decomposed into three aspects: having the power (being empowered) to vote; denying anyone else the power to object to 'appropriate exercise of this power; and subjecting inappropriate removal of this power to sanction. The associated entitlement was also decomposed into three aspects: having the physical power to access the 'voting machinery' (in one sense, an obligation on some party to provide a mechanism with which to exercise the power, the absence of which effectively renders the power otiose); there being an obligation on someone to count (or provide a mechanism to count) the vote in accordance with the way it was cast; and there being an obligation on someone, occupying a designated role, to report a 'fair outcome (i.e. the result should be declared according to the way the votes were cast with respect to the standing rules of the election).

Clearly there are different aspects of 'power' in operation here and without being able to distinguish conceptually and clearly between them it is not possible to verify the properties of an electronic voting system against, for example, the requirements given by the ACM Statement on E-Voting. However, the EC specification of [19], with its separation of concerns in terms of (institutionalised) powers, permissions, and obligations, makes it straightforward to check which aspects of the right and entitlement encapsulated in 'being enfranchised' are captured by which part of the specification, and clearly distinguish between institutionalised and physical powers.

5 Summary and Conclusions

One of the key factors that has led us to formulate the methodology outlined in this paper is our observation that computer scientists, not least in the MAS domain, tend to allow considerations of computational tractability and implementation to influence their model construction from the outset, with the consequence, all too often, that a seriously impoverished conceptual model steers the

process of system design. We have tried to indicate that the failure to provide a rich, conceptual model – constructed independently of computational considerations – is particularly unacceptable in the normative domain, the language of which is notoriously subtle and complex, as evidenced by hundreds of years of published work in legal, social and moral theory.

If adopted, these methodological proposals should carry with them recognition of the need to incorporate, in the training of future researchers in the field of norm-governed MAS, appropriate courses on the analysis of normative concepts its history, its theories, and its controversies - derived from sociological, legal and philosophical sources, with accompanying training in the formal methods that have been proposed in those disciplines as tools for conceptual analysis.

We do not, of course, wish to give the impression that we think that all existing MAS work lacks a middle-layer formal model of the type we advocate. For instance, in the field of agent communication there exist a number of formal, analytical models – so-called Agent Communication Languages (ACLs) – one primary function of which is to characterise the structure of particular types of communicative acts. The FIPA ACL language[2], the commitment-based languages of Colombetti and Singh [20, 21], and the convention-based language of [11] are representative examples.

This raises, in turn, the following question: where there are alternative, competing middle-layer analyses, which criteria should be employed to evaluate their respective merits ? Obviously, considerations of expressive capacity are crucial here, not only with regard to the ability to furnish adequate characterisations of the basic communicative act-types, but also with respect to providing a means of describing how communicative interaction emerges, evolves, in an agent population, and of how the possibility of deceit arises. (Consider here, for instance the work reported in [22].) Inspired, in part, by [23], an ongoing investigation into the emergence of communication, conventions, and roles promises to provide both an interesting test-bed for the assessment of ACLs, and a further domain of application for the methodology we have proposed in this paper.

References

1. Neville, B., Pitt, J.: A computational framework for social agents in agent mediated e-commerce. In Omicini, A., Petta, P., Pitt, J., eds.: Engineering Societies in the Agents World (ESAW) IV. Volume 3071 of LNCS., Springer (2004) 376–391
2. Hohfeld, W.: Some fundamental legal conceptions as applied in judicial reasoning. Yale Law Journal **23**(16) (1913)
3. Kanger, S.: New foundations for ethical theory. Almqvist & Wiksell (1957)
4. Kanger, S., Kanger, H.: Rights and parliamentarism. Theoria **38**(1-2) (1966) 1–44
5. Lindahl, L.: Position and Change: A Study in Law and Logic. Dordrecht: Reidel (1977)
6. Jones, A., Sergot, M.: Deontic logic in the representation of the law: Towards a methodology. Artificial Intelligence and Law **1**(45–64) (1992)
7. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. CUP (1969)

[2] Standard specification available from http://www.fipa.org

8. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Journal of the IGPL **4**(3) (1996) 429–445
9. Grossi, D.: Desigining Invisible Handcuffs. Formal Investigations in Institutions and Organizations for Multi-Agent Systems. PhD thesis, Utrecht University (2007)
10. Grossi, D., Jones, A.: Constitutive norms and counts-as conditionals. Handbook on Normative Systems ((submitted))
11. Jones, A., Parent, X.: A convention-based approach to agent communication languages. Group Decision and Negotiation **16**(101–141) (2007)
12. Jones, A., Parent, X.: Normative-informational positions – a modal-logical approach. Artificial Intelligence and Law **16**(7–23) (2008)
13. Steels, L., Brooks, R.: The artificial life route to artificial intelligence: Building Situated Embodied Agents. New Haven: Lawrence Erlbaum Ass (1994)
14. Artikis, A., Sergot, M., Pitt, J.: Specifying norm-governed computational societies. ACM Transactions on Computational Logic **10**(1) (2009)
15. Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., Turner, H.: Nonmonotonic causal theories. Artificial Intelligence **153**(1-2) (2004) 49–104
16. Kowalski, R., Sergot, M.: A logic-based calculus of events in. New Generation Computing **4**(1) (1986) 67–95
17. Sergot, M.: Action and agency in norm-governed multi-agent systems. In Artikis, A., O'Hare, G., , Stathis, K., Vouros, G., eds.: Engineering Societies in the Agents World (ESAW) VIII. Volume 4995 of LNCS., Springer (2008) 1–54
18. Oren, N., Luck, M., Miles, S.: A model of normative power. In: Proc. 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2010) 815–822
19. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Voting in multi-agent systems. Computer Journal **49**(2) (206) 156–170
20. Singh, M.: An ontology for commitments in multiagent systems. Artificial Intelligence and Law **7**(1) (1999) 97–113
21. Fornara, N., Colombetti, M.: Representation and monitoring of commitments and norms using OWL. AI Communications **23**(4) (2010) 341–356
22. Skyrms, B.: Signals: Evolution, Learning, and Information. OUP (2010)
23. Steels, L.: The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. Connection Science **17**(3-4) (2005) 213–230

Organizations with Improvised Coordination: OJAzzIC

Kathleen Keogh[1,2] *, Liz Sonenberg[2], and Wally Smith[2]

[1] Graduate School of Information Technology and Mathematical Sciences, The University of Ballarat, VIC 3353 Australia
[2] Department of Information Systems, The University of Melbourne, Australia
{k.keogh@ballarat.edu.au, l.sonenberg@unimelb.edu.au,wsmith@unimelb.edu.au}

Abstract. We consider requirements for adaptive agent organizations in a dynamic and complex situation. We propose a new 'jazzy' organizational approach to enable flexible real-time coordinated agent behaviour: plan-based performance that is loosely scripted and modified in real time with improvisation between parts of an organization.

Keywords: Multi-agent Systems, Adaptation, Organizations

1 Introduction

In this paper, we explore requirements for sophisticated agents acting in situations that demand flexible and adaptive coordination and describe an organizational model that can enable such behaviours. We are particularly concerned with settings where multiple agents, working together across multiple organizations, need to achieve goals that can change over time as the environment and the organizations themselves change.

Our model builds on prior work addressing adaptivity for agent organizations. In section 2 we examine the design of agent organizations and reflect on a case study of robotic agent teams working in a virtual search for weapons of mass destruction. In part, motivated by coordination requirements evident in human emergency management scenarios, we modify this simple case study to highlight more complex requirements for adaptivity.

The model we present starts with the approach used in OMACS [4], one that provides agent organizations with an ability to reorganise by selecting alternative goals and role-switch by reallocating agents based on available capabilities. Our extended case study and prior analysis [14] illustrates limitations of this approach, deficiencies that we seek to address by drawing on concepts from the literature. For example: the idea of using a Role Model as a role graph defining a hierarchy of roles in an organization has been used previously in the context of defining a functional decomposition of a goal and associating roles with goals e.g. [6, 12], and dynamic Role models are used in KB-ORG [22]; using contracts

* PhD student at The University of Melbourne, Australia

to define social behaviour was introduced in OperA [6]; Goal Trees that decompose goals into tasks and allow for synchronisation relationships can be seen in some models like STEAM [23] and TAEMS [13]; and Shared Plans [10] introduced explicit coordinating interactions based on obligations to ensure consistent goals and plans are dynamically shared in a group of agents sharing a goal.

Our approach is also informed by studies of adaptivity in human organizations, observed in complex dynamic, time critical settings [14]. For example, Bigley and Roberts [2] investigating the Incident Control System employed by a fire agency in USA identified four basic processes for improving reliability and flexibility in organizational change: structure elaborating – structuring the organization to suit the situation demands; role switching – re-allocating roles and role relationships; authority migrating – semi-autonomous adoption of roles according to the expertise and capabilities of the individuals available; and system resetting – changing organizational structure.

In Section 3 we describe our model: OJAzzIC and discuss how it supports adaptive organizations by enabling dynamic coordination and appropriate information sharing. We illustrate how this model can be used with a simple scenario and go on to propose that it meets the needs of more complex scenarios. The model combines the adaptive nature of a design enabling both reallocation of agents to tasks and dynamic goal decomposition with a dynamic social contract defining a dynamic organization structure (roles hierarchy, definitions and relationships) and knowledge sharing obligations within an organization.

We conclude in Section 4 with brief reflections on future work required to evaluate the described approach.

2 Agents in Organizations

2.1 Sensor Case Study

A robotic search for weapons of mass destruction scenario previously used in adaptive agent system design will be outlined and then we will modify this scenario to highlight the requirements we are addressing. This scenario was used to describe a simple adaptable organization based on OMACS (Organization Model for Adaptive Computational Systems)[4].

The scenario is based on a number of sensor robots used to search an area and identify suspicious objects. Each suspicious object needs to be checked to see whether it is a biological, chemical or radioactive weapon. Not all sensor robots have the capabilities to perform each object identification. When a suspect object is found, checks need to be performed until one matches. Each weapon may only be of one type, and the checks may be performed in any order. There are six agent types: Base Robot, Sophisticated Robot, Chemical Robot, Biological Robot, Nuclear Robot and Remover Robot. Each agent type is distinguished by a different set of default capabilities. Both the Chemical Robot and the Sophisticated Robot can identify chemical weapons, but the Sophisticated Robot's chemical detector is not as good as the Chemical Robot's chemical detector. All

robots can search and find suspect objects. When a weapon has been successfully identified, the Remover Robot removes it.

We describe modified scenarios to highlight complex requirements: mutual plan adjustment and sharing of limited resources.

Modification 1: Goals requiring multiple agents Suppose the chemical weapon can only be detected successfully by two robots together - Chemical Robot and Sophisticated Robot. In this scenario, the two robots need to be at the same object at the same time in order to perform the detection task. Both agents would have the goal to move to the object and wait for the other, then simultaneously perform the detection.

Modification 2: Goals achievable with multiple agents Suppose two agent types, Base Robot and Chemical Robot, can combine their capabilities in order to achieve the removal capabilities of a Removal Robot agent. In this scenario, if a Removal Robot fails or leaves the scene and another Removal Robot is not available, these two agents could together combine to achieve the tasks that were previously allocated to agent Removal Robot. This would require that the tasks previously allocated to Removal Robot could be shared between Base Robot and Chemical Robot with a detailed plan that coordinates their activities.

Modification 3: Resource contention Suppose the Removal Robots require an additional resource: a trolley, to help remove the detected weapons, but there are fewer trolleys than Removal Robots. Sharing of this resource requires that the agents coordinate their use and movement of the trolley, and hence they must share knowledge relevant to that sharing.

Reorganisation can involve reallocation of agents to existing roles, but may also be structural (e.g. roles, role definitions, changes in agent availability). Importantly, when agents engage in joint activity, they need predictability of behaviour so that each player can predict the behaviour of others in order to coordinate their own plans [15]. One approach is to include strict standardised processes and outputs to allow for predictability between tasks [16]. In this paper we focus on dynamic coordination and knowledge sharing within and between groups of agents, and leave for later important considerations of the management of interdependent resources. We use our previous analysis of SharedPlans [10] in the context of human coordination in the emergency management domain, which highlighted that agents need to ensure that they cultivate knowledge about their organizational structure as well as domain knowledge [14].

In summary, we arrive at the following requirements for coordination in an adaptive organization: awareness; appropriate knowledge sharing; and flexible adjustment of behaviour.

Awareness Players are obliged to work with awareness of others in the organization and to appropriately share information to relevant others. This includes creating agreed plans to work together, sharing resources and not creating intentions that could impede the achievement of other members' goals.

Knowledge sharing When there is a need for players to coordinate, then the organization members are obliged to communicate with each other as appropriate to establish coordination with relevant others. Information may flow out of one organization into other intersecting organizations as members identify relevance to other organizations that they belong to.

Flexible adjustment of behaviour In a dynamic situation involving uncertainty, there is not a fixed plan or recipe. Players mutually adjust their own behaviour to fit in with others. The plan emerges over time based on the situation and adaptation to address changes as they are realised. Goals may also change.

2.2 Some existing systems addressing organizational adaptability

We have looked at the following multi-agent system models: OMACS [4], KB-ORG [22], ORA4MAS [12], SharedPlans [10] and Opera [6], as each provides features that address part of our requirements. We use the term *organization* [24] to distinguish groups with the capability to reason about their organizational structure and intentional attitudes from groups of individuals who do not share mental attitudes. Coordination by proxies or intermediate layers within an agent architecture has been suggested to enable open systems with heterogenous agents to work together. For example, [21] assign a proxy agent responsible for coordination to each team player. This enables domain specialised agents to work as part of a larger team, without the need for knowledge about the team itself. However, with this approach, the agent players are not able to directly reason about team issues or coordination and this limits its applicability in our context.

OMACS [4] has been developed as a model to support reorganization (changing the roles required to achieve a task) and reallocation (changing the agents who are allocated to particular roles). In this model, Deloach also uses the *Capabilities* abstraction to enable flexible and dynamic reallocation of agents to roles. OMACS is well suited to adaptive goal selection and reassignment, however is limited to a 1:1 relationship between agents-roles and also roles-goals. A Role Model is used to define a list of tasks or responsibilities to be fulfilled. Each Role definition within the Role Model includes a required Capabilities list as well as a function that enables capabilities to be prioritised as to their relative importance. This function enables the measurement of agents' utility to play each role. It is possible to define multiple alternative roles that are capable of achieving a particular goal, however only one role can be allocated to one goal at any one time.

DeLoach and colleagues have proposed that adaptability in planning can be addressed by having alternative paths available in a goal decomposition. Their approach relies on being able to specify these in such a way as to match explicitly one agent to one role, and one role to one goal. When agents are no longer available and goals cannot be met according to the original goal-role-agent assignment, the OMACS system automatically reorganises and newly revised roles or goals are selected based on the currently available agents' capabilities [4]. However, if there is no agent available with an exact match on a required role, the

OMACS system does not support this. It does not provide for dynamic flexibility at the level of coordination of multiple agents together performing one role to achieve a goal – coordination between agents is implicit in the goals.

We have looked for more flexibility in the organizational knowledge so that a goal might be broken into synchronized/ordered tasks that could be assigned to agents. This approach is similar to that used in ORA4MAS [12] in which goals are decomposed into missions, then allocated to a set of responsible agents. A related approach has been promoted in the KB-ORG system, designed for automatic allocation of tasks to agents in a dynamic organization [22]. In KB-ORG, roles contain an assignable list of responsibilities and if necessary, roles can be split between a set of agents. KB-ORG is not an agentified system, however it addresses the concerns raised by allowing for roles to be split and when that occurs, to instantiate special explicit coordination roles. These roles help manage the coordination between an organization of multiple roles performing what might have previously been a single role. This dynamic organization is then governed by obligations to coordinate. In our design for Organizations Joining Adaptively with Improvised Coordination(OJAzzIC), we adopt a similar approach.

We are not unique in articulating agent interactions as requirements and modelling these separately in the design process for an agent system [20]. Others have described interactions as part of an organizational design [1, 9]. Relationships and awareness of relationships between agents in a dynamic organization are important to enable the appropriate coordination and communication. We have, however, attempted to design an organizational model that is flexible, allows multiple agents to coordinate dynamically to achieve a single goal and share knowledge appropriately within and across overlapping organizations.

3 Agent OJAzzIC: Agents in Organizations Joining Adaptively with Improvised Coordination

In this section, we outline our model, OJAzzIC, and how it can be used to instantiate a collection of organizations. This model is for Organizations Joining Adaptively with Improvised Coordination, the adaptive requirements resulting in a model that can capture the necessary static and dynamic knowledge in such as way as members can behave as a Jazz musician might - to improvise and adapt their script on the fly, but not in such a way as to interfere with the script or plan adopted by others. The plans need to be clear, but flexible. Behaviour needs to be coordinated, but not prescribed.

3.1 Adaptability in design - features of OJAzzIC

In order to establish an organizational design with the flexibility to adapt, the following major decisions were made: the model would include agentified organizations as first class entities [24]; Agent-Role-Task mapping using Capabilities would be used to enable flexible automated reallocation [4, 22]; Goals

could be shared by multiple roles using Capabilities and Tasks; and Organization instances created would include social contracts to define coordination and interaction obligations dynamically [26].

Agentifying the organization means we can treat the organization as one agent, with mental attitudes than can then be semantically related to individuals in the organization as desired [24]. This also means that no one individual needs to stay in a particular role (e.g. Leader) for appropriate communication with the organization. The organization is addressable as an agent in its own right [17].

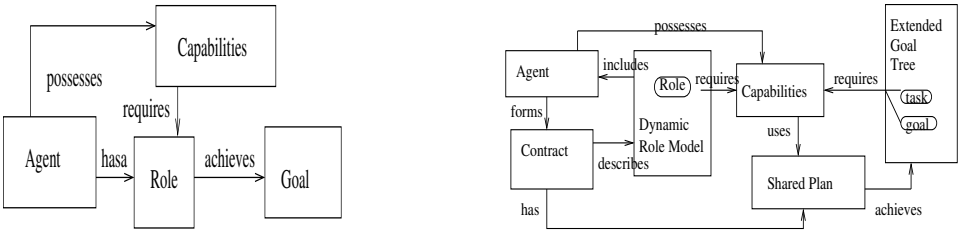
The essence of novelty in our approach is to combine the adaptive nature of a design enabling both reallocation of agents to tasks and dynamic goal decomposition with a dynamic social contract that defines explicit obligations and coordination policies on the fly. The social contract may be defined based on a predefined script that can be well defined or loosely governed by predefined landmarks [26]. Additionally, our proposal for use of multiple *instances* of overlapping organizations in a dynamic way enables adaptable coordination within and across organizations.

Key to the approach of OJAzzIC is the creation, as required, of dynamic instances of organizations. An organization is a group of agents with a structure (role model), shared knowledge and obligations/policies (contract) to ensure consistency of knowledge and plans. An organization instance may be a structure based on predefined roles, but it may also be a short term organization with coordination negotiated dynamically rather than based on predefined scripts. All organization instances are considered to be dynamic, first class, agentified entities.

Within each organization, there are context specific dynamic contracts defining agent allocations, obligations and roles. These ensure that coordination, knowledge sharing and behavioural obligations can be dynamically defined within a particular agent organization. Multiple organizations may be created as part of the original organization. Each organization has at least one goal to be achieved. Each organizational instance has a set of organizational knowledge attributes with values available to all members.

Figure 1 shows an abstract high level view of the relationships in an OJAzzIC organization along side a partial view of similar components in an OMACS organization [4]. Figure 1(b) is expanded upon in Figure 2. In Figure 1(a) Agents are related to Goals using the Role abstraction. As discussed in section 2.2, this assumes that one role will achieve one goal. We introduce an extra level of separation between roles and goals. In OJAzzIC, the Goal Tree is extended to include Tasks as a possible decomposition of Goals. Agents can be allocated based on responsibilities for Roles or based on Capabilities to fulfil Tasks. Extending Goal Trees using Tasks will be discussed further in section 3.3.

Our design is motivated by the need for a role description to change based on context and by the possibility that roles might need to be shared. In OJAzzIC, the dynamic Role Model enables goals to be shared between multiple roles and where necessary coordination roles are created. The structure of the organization - role relationships and role definitions - are dynamically defined. The OJAzzIC



(a) Agent-Goal Abstraction in OMACS

(b) Agent-Goal Abstraction in OJAzzIC

Fig. 1. Comparison of Agent to Goal Relationships in OMACS

Role Model is dynamic, it is created based on context and represents the roles instantiated by agents in the organization. The Contract is an explicit mental attitude adopted by the OJAzzIC organization and also defines obligations regarding knowledge sharing and will be discussed further in section 3.4.

Commitments toward maintaining and proactively sharing information in teamwork has been addressed in similar work with agent/human teams [27]. In order to establish information relevancy, explicit information-needs graphs have been used along with explicit mental models of team structure, team processes, and domain knowledge [8]. Information flow within groups has been described in terms of the relationships that form in a coordination loop [19].

We propose that encouraging information sharing in each network of agents requiring coordination is possible if each is considered a dynamic instance of an organization. Within each organizational instance, obligations exist ensuring appropriate knowledge sharing. Each organizational instance has a shared goal - such as a knowlege seeking goal or a goal to manage a dependency, or a goal to achieve a particular set of tasks. Agents may belong to multiple organizational instances simultaneously. In this way, an agent's knowledge can be shared across organizational boundaries where it is relevant to agents in more than one organization.

3.2 OJAzzIC Organization Model

Figure 2 shows the OJAzzIC organizational model. Each Organizational instance is created following this model, so an instance contains a role model that defines the roles, relationships between roles (including authority), coordination roles; a goal tree that defines goals, tasks and dependancies; a set of agents, possibly with relationships between them (such as distance, trust) with capabilities; a contract; and beliefs. Our organizational model is loosely based on OMACS [4], with extensions to provide for more flexible and dynamic role sharing. Where we adopt OMACS concepts without extension, we do not provide details here, but direct the reader to details elsewhere [5].

Goals are described in terms of tasks and tasks require capabilities. To achieve goals, the organization may require more than one agent to coordinate behaviour for related tasks. A player has capabilities, enacts a role and is also allocated

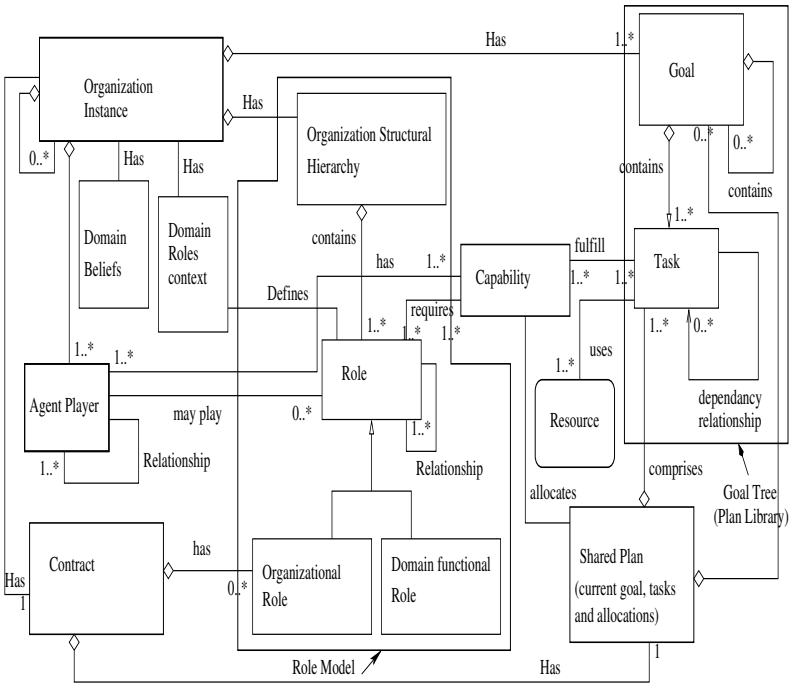


Fig. 2. The OJAzzIC organization model

tasks based on that role. A player may also be allocated tasks based on capabilities directly. Plans are based on instantiating general default plans or created based on the Goal Tree, according to a particular context. These plans to achieve goals are established dynamically using SharedPlans [10].

Agents start by belonging to at least one organization. All members are also part of the organization responsible for the main goal (e.g. the entire system). When two or more players need to coordinate and have not already formed an organization, a new (temporary) organization is formed and within that organization an explicit contract is formed that dictates norms, expectations, agreed goals and agreement to coordinate with others in that organization. Coordination within the organization relies on appropriate communication to share relevant information and share plans. New organizations that form overlap with existing organizations. We propose that in a dynamic organization, multiple temporary organizations are created. Permanent organizations may exist based on domain context (e.g. Fire Brigade). Each organizational instance created would be based on the OJAzzIC organizational model.

In OJAzzIC an organization O is a tuple:

$$<G*, R, A, C, P, \sum, \beta, \text{Re, Contract, } oaf\text{, achieves, requires, possesses}>$$

This extends OMACS with G*, R, Re and Contract. These are explained further in the subsequent sections.

G*: extended Goal Tree, including ordered tasks where possible. This defines the goals and how they could be decomposed into sub goals and ordered tasks;

R: the Role Model including a set of Roles, relationships between Roles and context based Role Definitions. This also includes coordination roles created dynamically as necessary;

A: a set of Agent Players;

C: a Capabilities set;

P: fixed policy constraints to apply to all members and to the allocation of tasks;

$\sum$: the domain model defining static objects in the environment and domain context Role definitions,

Re: a dynamic Resources list defining objects in the environment that can be used to help perform tasks;

Contract: The dynamic Contract contains a social contract and an information contract. The social contract comprises a SharedPlan [10] and a set of coordination Roles agreed for the organization. The Shared Plan outlines the current selection of tasks to achieve the goal and the allocations thus far assigning responsibilities for tasks. The information contract is a set of agreed policy obligations and commitments to intentions to ensure consistency of beliefs within the agent organization. The information contract contains β the current Beliefs set. This includes situated beliefs about the environment as well as beliefs about status of resources.

oaf: organization assignment function measures the utility of a particular SharedPlan assignment of Agents to Roles to Tasks;

achieves: function defining role assignment - how effective the behaviour of roles can be to achieve task T or goal G in a sharedPlan. This is used if a default plan needs revision or if a default plan cannot be found for a context. The Goal-Tree can be used to derive a plan.

requires: defines the capabilities required to play a role R or task T; and

possesses: function defining the quality of an agent's capability for a particular task. To decide how well an agent can play a role, the requires and possesses functions are combined into a function: capable. To decide how well an agent can play a role to achieve a goal, the capable function and achieves function are combined as a function: potential.

3.3 G* Extended Goal Trees and Dynamic Role Model

In order to achieve our aim of flexibility, we choose to keep separate the goals of an organization and the available roles that may be used to define (or allocate) responsibility to achieve these goals. The functional specification/decomposition of tasks to be achieved by similar multi-agent systems has been referred to as a goal tree that can be associated with individual plan recipes that achieve each leaf goal[11],[22].

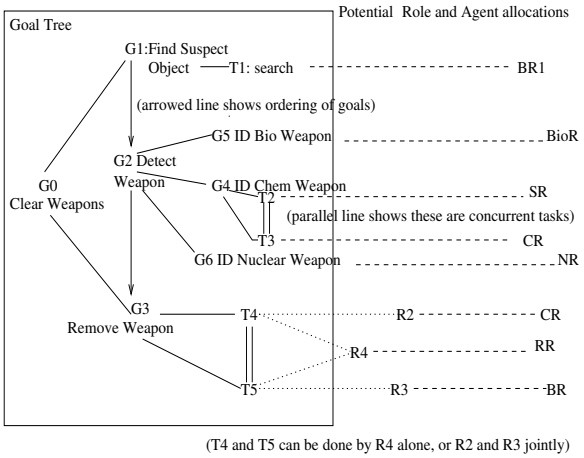


Fig. 3. OJAzzIC Extended Goal Tree G* showing potential allocations with Players

Synchronized Tasks We introduce an abstraction linking roles and goals, so that if a goal can not be achieved by one role, then multiple roles or agents can combine to achieve a goal by working together. We describe goals as composing synchronised tasks and tasks can be performed by agents with the appropriate capabilities. We choose to split goals into separate synchronised tasks rather than split roles, as intuitively, this abstraction fits with our observations from real life examples in the emergency management domain.

In Figure 3 an example OJAzzIC Goal Tree for the simple sensor case from section 2.1 is shown, as defined at design time, with an instance of allocations to roles and players also shown. The Goal Tree encapsulates knowledge about goal decompositions and in this case, we have indicated how the goal tree could be expanded at run time to link to capabilities of particular players and generate potential plans. The Goal Tree can be thought of as a plan Library. Default or preferred plans can be defined by indicating preferred paths at design time, however the flexibility to use the Goal Tree dynamically allows for dynamic planning and revision of plans.

As Figure 3 illustrates, a goal may comprise multiple subgoals. Goals may also be decomposed into tasks. Tasks and goals can be ordered. This abstraction is to enable the splitting of goals to share between multiple players. When a goal is split, performing the tasks requires coordination between the players. In figure 3, goal G0 is decomposed into sub goals G1, G2 and G3. G1 must be performed before G2. G1 is described in terms of search task T1. This figure also shows potential allocations of players (sensor robots) to these goals and tasks at run time. BR1 has the capabilities to perform task T1 and in this example has been allocated that task. Player RR has the capabilities to fulfil role R4 and thus could be allocated in that role to perform both tasks T4 and T5. Alternatively, CR could perform T4 and BR could perform T5. Tasks T4 and T5 must be done simultaneously - indicated by the parallel lines connecting them.

Using the sensor agent case as a very simple example, we can identify that if allocations were made based on Figure 3 with CR and SR working together to achieve the goal Detect Chemical Weapon, then CR and SR would create an instance of an organization. Creating this organization would then obligate each agent (as defined in their social contract) to appropriately share information such as the SharedPlan to facilitate coordination.

Suppose initially an organization forms involving BR1 and perhaps some other agents conducting a search of the area. Then as suspect objects are found, new organizations would spring up including agents able to detect weapons. For example, an organization with BioR, SR, CR, NR could form with BR1 as a leader to achieve the goal, G2. Then when a weapon has been identified, a new organization involving these or other agents would form to remove the weapon. We could imagine a more complex scenario where there were hundreds of agents involved in the weapons search over a large area. Multiple organizations could be created dynamically as agents elect to work together to achieve goals. When one agent is part of multiple organizations, this overlap allows for relevant information (e.g. location, detection status of object) to be dispersed across the network from organization to organization.

Resources are explicit entities in the OJAzzIC organization's knowledge base. This is because of the potential for resource contention amongst agents in the organization and thus the need to coordinate interdependencies. Having this knowledge explicit will enable direct reasoning within the Agent Organization, based on priorities, negotiated protocols or the use of coordination artefacts [3, 18]. We do not address such reasoning further in this paper.

The sharing of relevant knowledge within and across organizations is important as knowledge may be distributed. Obligations to ensure appropriate knowledge-sharing about SharedPlans [10] has been well defined and we adopt this intention based approach to planning. We have adopted the use of contracts to manage obligations to ensure agents will share domain-knowledge as well as structural and coordination knowledge within each organization. Collective obligations can be implemented as policies to govern joint activity and teamwork [25]. We leave implementation details aside and in the next section, describe at a conceptual level the contents of the contracts that need to be created.

3.4 Contracts

At the time a new organization is instantiated in OJAzzIC, players explicitly form an organizational contract. As agents join an organization, then agents commit to a social contract that defines interaction within the organization. Beliefs, values, objectives, protocols and policies may be defined in the context of the social relationships that exist within the society. Each organization exists for the duration of time in which there is a need for that group of agents to be coordinated. The network of organizations within organizations is dynamic.

Within each organization, the *social contract* is formed to define role descriptions and agreed role allocations. Default Role descriptions may be abstractly

defined at organizational design time and adapted dynamically. Roles are defined with default capabilities, authority levels and obligations. Having an explicit social contract provides the ability to predict others' behaviour and flexibly adapt individual goals in anticipation of others' needs and behaviour. The social contract also defines an agreed model for command, control and coordination. Coordination Roles such as Leader, Resource Manager, Knowledge Manager and Contract Manager could be identified and allocated if needed.

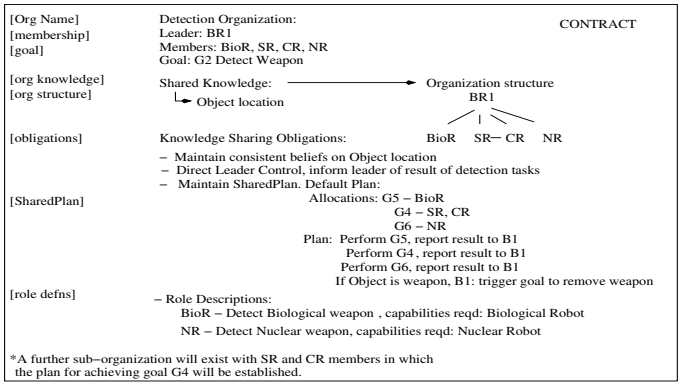


Fig. 4. Partial contract for the detection organization in the sensor robot case scenario

To address the requirement of sharing mutual knowledge within relevant organizations, we propose to use *information contracts* within each organization. Based on the need to cultivate knowledge sharing [14], information contracts include policies that obligate players in an organization to all adopt joint intentions to cultivate mutual knowledge within a relevant sphere of influence (the organization). These joint intentions must be shared by relevant groups as they emerge as networks requiring coordination within a larger organization. To ensure that information flow is relevant and not saturating the network, the awareness of relationships and multiple coordination networks within the organization is important. Obligations to share information are limited to the players within each circle of coordination ie. within each individual organization that forms. As a player can belong to multiple organizations, the overlap enables relevant information to be dispersed across a wider network as necessary. Figure 4 shows parts of an example contract based on the sensor robot case study.

4 Conclusion

We have proposed a model for agent organizations that require adaptability and dynamic improvisation. We have drawn on investigations of adaptive human organisations [2, 7, 15, 16, 19, 24],taken features from existing multi-agent systems design approaches [4, 6, 10, 22] and synthesised these to fit our requirements.

We have organizations as first class entities so membership does not need to be fixed and members have access to the mental state of the organization. Organizations are created based on a need to coordinate or a shared goal. Within organizations, players are obliged to share information and maintain consistent plans. We have contract based dynamic organizations so that organizational structure is defined/agreed and modifiable at run time. We have a flexible goal-task decomposition that enables definition of tasks and goals that require multiple agents. We allow goals to be shared, where multiple agents can combine their capabilities to achieve a goal. Each organization manages obligations to ensure relevant knowledge is shared within and between organizations.

Future work is needed to formalise this design, implement and evaluate it. We plan to begin with a simplified scenario based on the modifications to the search for weapons discussed early in this paper [4], with an ambition to extend to use in organizations involving agents and humans, c.f. [27].

Hybrid human-agent systems have potential applicability in simulation and training[3]. Sophisticated agents could potentially be used as surrogate humans in virtual organizations for training exercises. To be useful, such agents need to behave in a predictable way from a human perspective [15] and need to be able to naturally coordinate behaviour with other players. Our work contributes by proposing a conceptual model for dynamic and adaptable agent organizations working together.

Acknowledgements The authors thank Gil Tidhar for his conversations and suggestions regarding OJAzzIC. Also thanks to the anonymous reviewers for their helpful remarks.

References

1. E. Argente, V. Julian, and V. Botti. Mas modeling based on organizations. In *Proceedings of Agent-Oriented Software Engineering workshop IX, 2008, LNCS*, volume 5386, pages 16–30. Springer-Verlag, 2009.
2. G. A. Bigley and K. H. Roberts. The incident command system: High reliability organizing for complex and volatile task environments. *Academy of Managment Journal*, 44(6):1281–1299, 2001.
3. D. Daghan L. Acay, G. Tidhar, and L. Sonenberg. Extending agent capabilities: Tools vs. agents. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 259–265, 2008.
4. S. DeLoach, W. H. Oyenan, and E. Matson. A capabilities-based model for adaptive organizations. *Autonomous Agents and Multi-Agent Systems*, 16:13–56, 2008.
5. S. A. DeLoach and G.-O. J. Carlos. O-MaSE: A customizable approach to designing and building complex, adaptive multiagent systems. *Int. Journal Agent Oriented Software Engineering*, 4(3), 2010.
6. V. Dignum, J. Vázquez-Salceda, and F. Dignum. Omni: Introducing social structure, norms and ontolgoies into agent organizations. In *PROMAS 2004 Proceedings, LNCS*, volume 3346, pages 181–198. Springer, 2005.

[3] e.g. The HART Workshop – `http://www.lorentzcenter.nl/lc/web/2010/422/program.php3?wsid=422`

7. E. Entin, F. Diedrich, D. Kleinman, W. Kemple, S. Hocevar, B. Rubineau, and D. Serfaty. When do organizations need to change (Part II)? Incongruence in action. In *Proceedings of the Command and Control Research and Technology Symposium.*, Washington, DC, 2003.
8. X. Fan, S. Sun, B. Sun, G. Airy, M. McNeese, and J. Yen. Collaborative RPD-enabled agents assisting the three-block challenge in command and control in complex and urban terrain. In *Proceedings of 2005 BRIMS Conference Behavior Representation in Modeling and Simulation*, pages 113 – 123, Universal City. CA, 2005.
9. B. Gâteau, D. Khadraoui, E. Dubois, and O. Boissier. Moiseinst: An organizational model for specifying rights and duties of autonomous agents. In *Proceedings of Third European Workshop on Multi-Agent Systems (EUMAS 2005)*, pages 484–485. Elsevier Science B.V., 2005.
10. B. Grosz and L. Hunsberger. The dynamics of intention in collaborative activity. *Cognitive Systems Research*, 7(2-3):259–272, 2006.
11. B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2005.
12. J. Hübner, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents: 'Giving the organisational power back to the agents'. *Autonomous Agents and Multi-Agent Systems*, 20:369–400, 2010.
13. S. Kamboj and K. Decker. Organizational self-design in semi-dynamic environments. In *Proceedings of the sixth International Joint Conference on Autonomous Agents and multi Agent Systems*, Hakodate, Japan, May 2006.
14. K. Keogh, E. Sonenberg, and W. Smith. Coordination in adaptive organisations: Extending shared plans with knowledge cultivation. In *Post conference proceedings-Organised Adapatation in Multi-Agent Systems (OAMAS08), LNAI*, volume 5368, pages 90–107. Springer, 2009.
15. G. Klein, P. Feltovich, J. Bradshaw, and D. D. Woods. Common ground and coordination in joint activity. In W. Rouse and K. Boff, editors, *Organizational simulation*, pages 139–184, New York, 2004. Wiley.
16. H. Mintzberg. *Structure in Fives Designing Effective Organizations*. Prentice-Hall, New Jersey, USA, 1983.
17. J. Odell, M. Nodine, and R. Levy. A metamodel for agents, roles and groups. In *Proceedings of AOSE 2004, LNCS*, volume 3382, pages 78–92, 2005.
18. A. Omicini, A. Ricci, and M. Viroli. Coordination artifacts as first-class abstractions for mas engineering: State of the research. *SELMAS 2005, LNCS*, 3914:7190, 2006.
19. B. Prue, M. Voshell, D. Woods, J. Peffer, J. Tittle, and W. Elm. Synchronised coordination loops: A model for assessing distributed teamwork. In *Proceedings of NATO Research and Technology Organisation (RTO) Human Factors and Medicine Panel (HFM) symposium. Adaptability in Coalition Teamwork*, pages 17–1–17–12, April 2008.
20. I. Rahwan, T. Juan, and L. Sterling. Integrating social modelling and agent interaction through goal-oriented analysis. *Computer Systems Science and Engineering*, 3:87–98, 2006.
21. P. Scerri, D. Pynadath, N. Schurr, A. Farinelli, S. Gandhe, and M. Tambe. Team oriented programming and proxy agents: The next generation. In M. Dastani, J. Dix, and E. Falah-Seghrouchni, editors, *LNAI 3067 PROMAS 2003*, pages 131–148. springer-verlag, 2004.
22. M. Sims, D. Corkill, and V. Lesser. Automated organization design for multi-agent systems. *Autonomous Agents and Mult-Agent Systems*, 16:151–185, 2008.

23. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
24. G. Tidhar. *Organization-Oriented Systems: Theory and Practice*. PhD Thesis, University of Melbourne, 1999.
25. J. van Diggelen, J. Bradshaw, M. Johnson, A. Uszok, and P. Feltovich. Implementing collective obligations in human-agent teams using kaos policies. In *Proceedings of Workshop on Coordination, Organization, Institutions and Norms (COIN), AAMAS*, Budapest, Hungary, 2009.
26. H. Weigand, V. Dignum, J.-J. Meyer, and F. Dignum. Specification by refinement and agreement: Designing agent interaction using landmarks and contracts. In *Engineering Societies in the Agents World III, LNCS*, volume 2577, pages 67–79, 2003.
27. J. Yen, X. Fan, S. Sun, T. Hanratty, and J. Dumer. Agents with shared mental models for enhancing team decision-makings. *Journal of Decision Support Systems: Special issue on Intelligence and Security Informatics*, 41(2), 2005.

Towards Justifying Norm Compliance

Ioan Alfred Letia[1] and Anca Goron[1]

Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-400391 Cluj-Napoca, Romania

Abstract. Norms should help societies of agents to achieve higher performance without restricting their autonomy too much. As systems evolve the need to accommodate different sets of norms requires a more refined understanding of their effects on the overall behavior of agents. We introduce justifications in normative systems with the aim to better understand the compliance to norms, including situations of conflict between norms that might arise.

1 Introduction

Jones and Sergot [13] have shown the way on using a deontic and action logic to analyze a number of notions crucial to the understanding of organized interaction in institutions, starting with the following introduction.

> It is a commonplace feature of legal systems, and other norm-governed organizations, that particular agents are empowered to create certain types of states, by means of the performance of specified types of acts. Typically, the states created will have a normative character according to which obligations and rights are established for some agents vis-a-vis others, as for instance when a contract is made, or a marriage is effected, or ownership of an item is transferred. The performances by means of which these states are established will often be of a clearly prescribed, perhaps ritualized nature, involving the utterance of a particular form of words (e.g., the utterance of a specific type of performative sentence), or the production of a formal document, or the issuing of a pass, perhaps in a particular context (e.g., in the presence of witnesses).

We are concentrating here to develop a method that can help such systems visualize, in the sense of offering proofs, whether the norms have been complied with or broken.

Advantages and some of the problems when using norms in multi-agent systems have been shown lately to be of significant importance [17]. Controlling multi-party interactions with norms [7], but also programming norm change [16], making norms concrete [1] have contributed to a deeper understanding of ways to achieve flexible systems and also to monitor them [2].

Provision of explanation generating functionality has been found to be very helpful for both developing and providing proofs when using ontologies [12] by way of explanations via justifications. We are pursuing

a similar path for the usage of norms in multi-agent systems, looking to ways of exploiting a justification logic version [4, 10, 9] of a hybrid logic [3]. As remote positive/negative justification checkers we employ the value based argumentation [6, 5].

2 Running example

Example 1. We consider a Traffic Management System (TMS), responsible for controlling the traffic flow through an intersection. The system functions based on a set of predefined regulations or norms (table 1) that specify the permissions and obligations of the driver agents passing through the intersection. The role of each norm is to ensure the safety of the passing drivers, passengers and pedestrians, general traffic security and a continuous and regular flow.

Table 1. Set of norms for the Traffic Management System

Norm	Specification
N1	A driver must stop at the Red color of the traffic lights
N2	A driver may pass at the Green color of the traffic lights
N3	A driver must signal its intention of changing the lane or the driving direction
N4	A driver must not exceed the speed limit of 50 km/h when passing through the intersection

Any violation of the norms N1, ..., N4 is sanctioned by a fee and the application of 10 penalty points for the driver agent. When summing 50 penalty points, the driving license of the driver is automatically suspended.
Having specified the set of norms together with the corresponding sanctions to be applied in case of the infringement of one or more norms, we might assume that the proper functioning of the TMS is ensured. However, taking into consideration the examples offered by the real world, which is governed by unpredictable events, we can easily state that the above assumption is false. Nevertheless, we are obliged to analyze if the set of abstract norms can cover all the possible concrete events from inside the intersection.
In this direction, we consider the following scenario for our analysis (figure 1). The driver from vehicle A stops on the right lane at the Red color of the traffic lights. Other three drivers stop on the middle lane and four drivers proceed likewise on the left lane, waiting for the Green light of the traffic lights. In the meantime, an ambulance carrying a patient to the hospital represented by vehicle B approaches the intersection, signaling its presence for the other drivers. Checking each of the three lanes, the driver from vehicle B decides to use the right lane to enter the intersection as there is only one vehicle stopped. The driver from vehicle

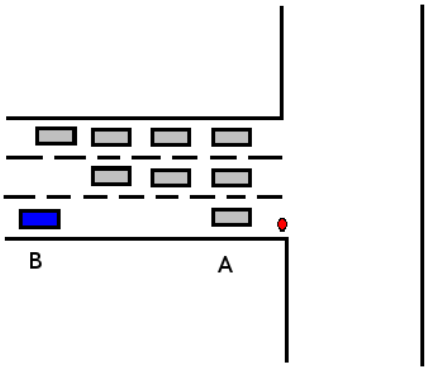


Fig. 1. Example scenario in traffic intersection

A notices the approaching ambulance, checks its right and left side and decides to pass on the Red color of the traffic lights and drive to the right, clearing the right lane for the ambulance to pass. We might now raise the questions: Did the driver from vehicle A make the right choice? Did vehicle A violate the TMS norms? Should the driver from vehicle A be sanctioned for his actions?
In order to provide a proper answer for each of the above questions, we will try applying two reasoning methods and provide a justification for the course of actions chosen by the driver agent from vehicle A.

3 Fragment of the Hybrid Justification Logic

Justification logic [4] is based on the classical propositional logic augmented with justification assertions $t : F$, meaning that t is a justification for F. Hybrid logics are extensions of standard modal logics, involving symbols that name individual states in models [3]. The language of proofs we are using is a fragment of the justification logic of hybrid logics, that internalize both semantics and proofs [10, 9].
In justification logic we can use $justification\ variables\ x_1, x_2, \ldots$ (capable of making relative justification conclusions) and $justification\ constants$ $c_1, c_2, \ldots$. The operation $\cdot$ is an application operation, used for showing that if t is the justification of $X \supset Y$ and u is the justification of X then $t \cdot u$ is a justification of Y. Facts that make up a state of a possible world i should be certified by the facts f_i, the indexed family of constant symbols.
In the hybrid language $\mathcal{H}(@)$ the prefix operator $@_i$ is used for the formula $@_i\phi$, for saying that ϕ is true at the world named by i.

Definition 1. *Some of the operators of the basic hybrid justification logic [10] that we need in our presentation are the following:*

– **modus ponens**

$$\frac{X \;\; X \supset Y}{Y} \tag{1}$$

– $\cdot$ **axiom**

$$t : (X \supset Y) \supset (u : X \supset (t \cdot u) : Y) \tag{2}$$

– **remote fact checker***: For each nominal i and propositional letter P we have the facts*

$$@_i P \supset f_i : @_i P \tag{3}$$

$$@_j \neg P \supset f_j : @_j \neg P \tag{4}$$

where f_i (respectively f_j) certifies the fact P (respectively $\neg P$) in the possible world i (respectively j).

– **remote positive justification checker** *For any formula X justified by the term t and any nominal i the operator $!_i$ is used to show that t is checkable.*

$$@_i t : X \supset (!_i t) : @_i t : X \tag{5}$$

– **remote negative justification checker** *For any formula X justified by the term t and any nominal i the operator $?_i$ is used to show that t is not checkable.*

$$@_i \neg t : X \supset (?_i t) : @_i \neg t : X \tag{6}$$

Example 2 (continued). If we represent the norm N1 by

$$redColor \supset stop \tag{7}$$

then we have the following derivation:

1. $@_1 redColor \supset f_1 : @_1 redColor$,
   (remote fact checker)
2. $x : (redColor \supset stop) \supset (f_1 : @_1 redColor \supset (x \cdot f_1) : @_1 stop)$,
   ($\cdot$ axiom and (7))

The justification for the action of the driver (stop) in the situation $@_1$, according to the norm N1 is given by the result of the above derivation.
$f_1 : @_1 redColor \rightarrow$
$x : (redColor \supset stop) \supset (f_1 : @_1 redColor \supset (x \cdot f_1) : @_1 stop)$

4 Grounding the Norms

The Traffic Management System from our example represents a normative system, whose proper functioning is ensured by a preestablished set of norms (table 1). We can observe that each norm is defined in such a way to abstract from the concrete events and situations that it is supposed to cover [1], allowing this way to be applied for a wide range of situations and possible events. However, the downside of using a high level of abstraction for norm specifications is represented by the difficulty in relating abstract norms to concrete events that may occur inside the system. In order to overcome this issue, [1] proposes a solution based on the use of counts-as statements, which provide a link between abstract and concrete, real-world facts. Moreover, based on their ability to relate

system events to agent actions, highlighting their effects, counts-as relations can provide a powerful reasoning means in what it concerns the meaning of an agent action and what it brings about in a certain context, focusing in the same time on the normative impact [1] of performing those actions in order to achieve a certain goal.
On what it concerns our scenario, counts-as statements can offer the possibility to represent the connection between concrete events and abstract concepts used for specifying the norms of the Traffic Management System under the form of static links, which highlight the connection between them, capturing the normative consequences of an agent actions (figure 2).

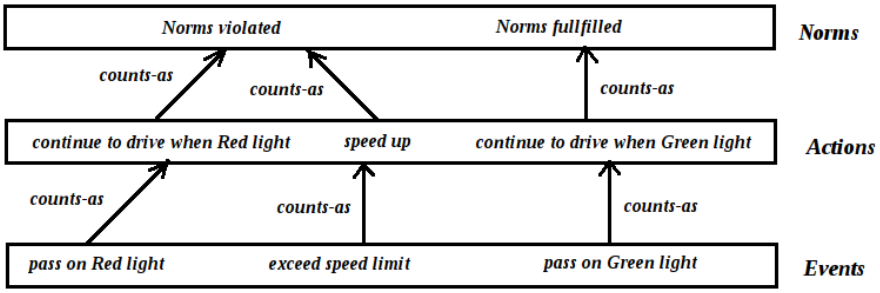


Fig. 2. Fulfillment and violation of TMS norms

Figure 2 captures the representation of the relation between concrete or explicit actions and events happening inside the intersection and the abstract norms governing the traffic flow. We return now to the possible scenario highlighted in figure 1 and try representing it based on counts-as statements, which prove to be really useful when trying to emphasize what happens when the context changes (figure 3).
Figure 3 highlights the violation of norm N1 of the Traffic Management System by the driver from vehicle A when passing on the Red color of the traffic lights and turning to the right in order to allow the ambulance to pass. If we were to base our reasoning only on the TMS norms applied to this context, then we could easily conclude that the agent made a wrong decision and he must be sanctioned for his actions.
However, if we extend the set of norms on which our reasoning is based and include also the general applying driving norms, more specifically the norm stating the obligation of each driver to allow emergency vehicles (police cars, ambulances etc.) to pass, moving out of their way, then we can no longer easily arrive to the conclusion that the agent made the right or the wrong choice. The new representation is captured in figure 4, while the extended set of norms is emphasized in table 2.
By analyzing figure 4, we can observe that a conflict is generated between norms N1 and N5 based on the fact that the same action of passing on

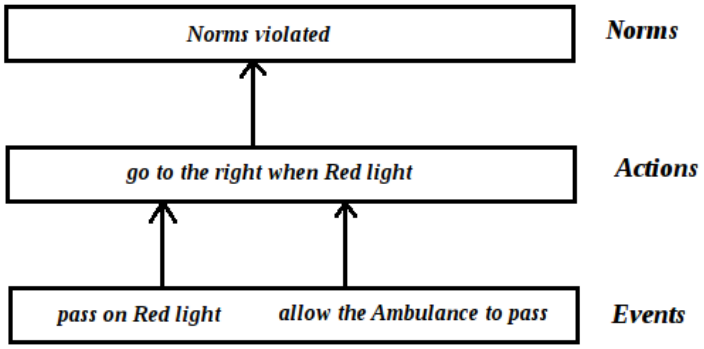


Fig. 3. Violation of TMS norms in a possible scenario

Table 2. Extended set of norms for the Traffic Management System

Norm	Specification
N1	A driver must stop at the Red color of the traffic lights
N2	A driver may pass at the Green color of the traffic lights
N3	A driver must signalize its intention of changing the lane or the driving direction
N4	All drivers must not exceed the speed limit of 50 km/h when passing through the intersection
N5	A driver must allow emergency vehicles to pass by moving out of their way

the Red color of the traffic lights leads to both the violation and the fulfillment of the TMS norms.
Vasconcelos et al. [17] define conflicts as situations that arise when an action is simultaneously prohibited and permitted/obliged, and its variables have overlapping values. The variables of a norm specify its scope of influence, that is, which agent/role the norm concerns, and which values of the action it addresses. Moreover, they propose a normative conflicts resolution mechanism based on the addition of constraints that will prevent variables from having overlapping values. However, the proposed solution restricts the norms representation selection to the use of atomic formulas and might prove difficult or too complex to be applied in different situations as, for example, when dealing with deontic conflicts, that is when an action may be simultaneously prohibited and obliged. As an alternative solution to the normative conflict resolution problem, we further propose a method based on the reasoning power of value based argumentation.

Example 3 (continued). If we represent the norm N5 by

$$emergency \supset \neg stop \tag{8}$$

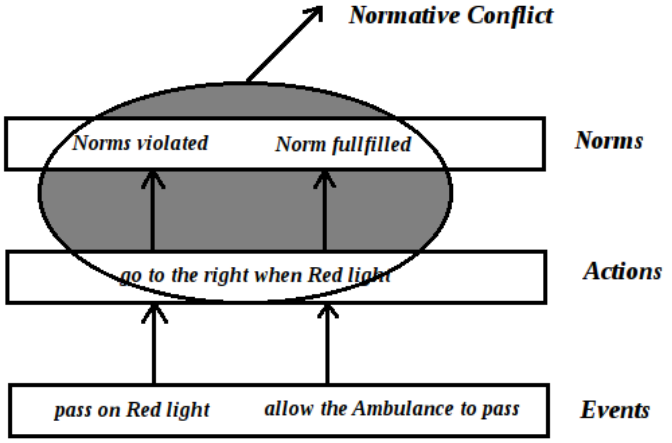


Fig. 4. Conflict between TMS and general applying norms

then we have

1. $@_2 emergency \supset f_2 : @_2 emergency$,
   (remote fact checker)
2. $@_2 \neg y : stop \supset (?_2 y) : @_2 \neg y : stop$),
   (remote negative justification checker and (8))

showing that the driver did not fulfill ¬ stop, thereby allowing the ambulance to pass, or formally:
$f_2 : @_2 emergency \rightarrow$
$@_2 \neg y : stop \supset (?_2 y) : @_2 \neg y : stop$

5 Value-based Argumentation for Compliance

Definition 2. *[5] A value based argumentation framework (VAF) is a 5-tuple: VAF $\{AR, attacks, V, val, P\}$, where AR represents a finite set of arguments, attacks is an irreflexive relation on the set AR, V is a finite nonempty set of values, val is a function mapping the elements of AR to the elements of V and P is a set of possible audiences. Moreover, an argument A relates to value v if accepting A promotes or defends v, where $v = val(A)$, $val(A) \in V$, for every $A \in AR$.*

[14] extends VAF by allowing ordering on values to be applied, such that an argument A_1 successfully attacks (defeats) an argument A_2 only if the value promoted by A_2 is not ranked higher than the one promoted by A_1, according to some partial ordering $>$ of values. Furthermore, the notion of an argument promoting $(P = +)$ or demoting $(P = -)$ values to a given degree X (denoting a real number) is introduced $(val = (A, X, V, P))$, meaning that if A_1 and A_2 would promote the same value v, A_1 would defeat A_2 if it promotes the value v to a higher degree. In practical

reasoning, arguments are used to provide a presumptive justification for a certain action, which instantiates an argument scheme, representing an extension of the Walton's sufficient condition scheme [6]:

In the current circumstances R
We should perform action A
Which will result in new circumstances S
Which will realize goal G
Which will promote value V.

If we consider again the Traffic Management System example, we can observe that, similarly to the real world, norms are defined in order to ensure values such as traffic security, pedestrians safety, efficiency and flow control through their fulfillment. The violation of one or more norms brings about insecurity, traffic disturbances as the mentioned values do no longer apply. A special case is represented by norm N5 stating the obligation of each driver to allow emergency vehicles to pass, a norm whose fulfillment brings about values such as life for our example in which the driver agent must allow an ambulance carrying a patient in critical condition to pass. If norm N5 is violated, the life of the patient is in danger as it may delay the arrival to the hospital and his condition might aggravate.

Each of the previously mentioned values will be further considered in the reasoning process based on VAF [8] about what would be the right solution to apply in such a context and, based on the outcome, to provide a positive or negative justification for the choice of the driving agent. If we were to take into consideration only those actions that fulfill the initial TMS set of norms as the right choices, thus actions that promote values such as traffic security, flow control and pedestrians safety, then the agent should choose to stop at the Red color of the traffic lights. If we consider that the life of the patient is of higher importance, then the choice will be totally different. We can observe that the selection of the right action depends merely on the importance of promoting a certain value over another. Thus, as a first step we must decide over a hierarchy of values ranked according to their importance. In this direction, we apply an ordering relation α over the set of values $V = \{$pedestrians safety, traffic security, flow control, life$\}$, obtaining the following hierarchy of values: flow control $<$ traffic security $<$ pedestrians safety $<$ life, where the value of flow control has the lowest priority in the reasoning process and the value of life the highest priority. Moreover, the entire reasoning process will be translated from the normative level to the agent actions level, actions that bring about the promotion of one or more values. The arguments used for justifying each of the possible options are presented in table 3.

If we analyze the arguments from table 3, we can observe that each of the arguments $\{A_2, A_4, A_6\}$ defeat (represented by $\rightarrow$) argument A_1 along with the subset of arguments $\{A_3, A_5\}$, as based on the option they sustain, the value of life is promoted, which has the highest ranking in the α ordering relation. A_2 is defeated at its turn by A_4 and A_6 (represented by $\leftarrow$) as, although they all promote the value of life, the last two arguments mentioned promote also the value of pedestrians safety, the second in line in the α ordering relation. Thus, at a first glance, it

Table 3. Arguments for justifying the driving agent actions

Arg.	Argument Specification	Promoted/Demoted Values
A_1	In the current circumstances, the agent should wait for the Green light before driving to the right, which will not affect the traffic from inside the intersection, promoting the values of traffic security, pedestrians safety and flow control	+traffic security +flow control +pedestrians safety -life
A_2	In the current circumstances, the agent should climb on the right sidewalk allowing for the ambulance to pass, promoting the values of life, flow control and traffic security	-pedestrians safety +flow control +traffic security +life
A_3	In the current circumstances, the agent should not try climbing on the right sidewalk, but wait for the Green light as it may not create enough space for the ambulance to pass and it may injure the pedestrians, promoting this way the values of pedestrians safety, traffic security and flow control	+pedestrians safety +flow control +traffic security -life
A_4	In the current circumstances, the agent should try moving on the middle lane in front of the first car waiting at the traffic lights, allowing the ambulance to pass, promoting this way the value of life and pedestrians safety	-traffic security -flow control +life +pedestrians safety
A_5	In the current circumstances, the agent should not move on the middle lane, but wait for the Green light as there it might not be enough space for stopping in front of the first car, affecting the traffic and not creating enough space for the ambulance to pass, thus demoting the value of life	+traffic security +flow control +pedestrians safety -life
A_6	In the current circumstances, the agent should drive to the right and pass on the Red light allowing the ambulance to pass, not before checking that no car is approaching, signaling his intention and slowing down, promoting the value of life	-traffic security -flow control +life +pedestrians safety

would seem that the available options for ensuring that the life of the patient from the ambulance is not in danger is for vehicle A to either turn to the right without waiting for the Green light or to move on the middle lane. However, in order to select the best option to consider, we have to take into consideration also the aspect referring to the degree to which the value of life is promoted by each of the two possible choices. In this direction, considering the risks implied by each action and also the justification provided by argument A_5, emphasizing the possibility of vehicle A moving on the middle lane not to provide enough space for the ambulance to pass without additional maneuvers or the possibility of damaging the other cars stopped at the traffic lights, we can state that A_6 promotes the value of life at a higher extent than A_4:

$$A_6 \rightarrow A_4 \rightarrow A_2 \rightarrow A_1$$
$$A_6 \rightarrow A_4 \rightarrow A_2 \rightarrow A_3$$
$$A_6 \rightarrow A_4 \rightarrow A_2 \rightarrow A_5$$

Based on the presented argument scheme, we can conclude that the agent made the right choice when deciding to drive to the right, passing on the Red color of the traffic lights, justifying its actions by the fact that it allowed this way the ambulance to pass, not jeopardizing thus the life of the patient carried to the hospital. However, we also have to consider the negative aspect of his choice, that is the risk of jeopardizing the traffic flow through the intersection and the safety and security of the other agents.
Moreover, we can observe that using a VAF based reasoning approach in such a situation provides us with the possibility to decide upon the course of actions to follow, and, additionally, it offers us a viable justification [8] for the infringement of one or more norms in order to achieve a goal and promoting a higher ranking value.

Example 4 (continued). Here the assumption is that we do not have norm N5, but we can consider an argument like A6 that promotes the value life:

$$\neg stop \supset promoteLife \tag{9}$$

Then we have

1. $@_2 emergency \supset f_2 : @_2 emergency$,
   (remote fact checker)
2. $z_1 : @_2 \neg stop \supset (!_2 z_1) : @_2 z_1 : \neg stop)$,
   (remote positive justification checker and (8))
3. $z_2 : (\neg stop \supset promotesLife) \supset (z_1 : @_2 \neg stop)) \supset (z_1 \cdot z_2) : @_2 promoteLife$,
   (remote positive justification checker and (9))

showing that the driver did not stop on Red light, promoting life (A6). That is, given that
$f_2 : @_2 emergency$
we have the justification:
$z_2 : (\neg stop \supset promotesLife) \supset (z_1 : @_2 \neg stop)) \supset (z_1 \cdot z_2) : @_2 promoteLife$

6 VAF Approach for Remote Justification

The entire reasoning process presented in the previous section was captured in the CaSAPI [11] argumentation environment [1], based on the MARGO platform [2], which provides mechanisms to evaluate and suggest decisions, explaining at the same time the choice made.

```
goalrule(r01, react(X), [waitForGreen(X)]).
goalrule(r02, react(X), [passOnRed(X)]).
goalrule(r03, react(X), [moveAside(X)]).
decisionrule(r10(X),waitForGreen(X),[d(X),
             doNotAllowAmbulanceToPass(X),
             t_security(X),p_safety(X),flowcontrol(X),
                    sn(life(X))]).
decisionrule(r20(X),passOnRed(X),[d(X),
                    allowAmbulanceToPass(X),
                    sn(t_security(X)),p_safety(X),
                    sn(flowcontrol(X)),life(X)]).
decisionrule(r30(X),moveAside(X),[d(X),
                    ambulanceCannotPass(X),sn(t_security(X)),
                    sn(p_safety(X)),flowcontrol(X),
                    sn(life(X))]).
decisionpriority(r20(X),r30(X)).
epistemicrule(f00,do_Not_Allow_Ambulance_To_Pass(traffic_participants),[]).
epistemicrule(f03,allow_Ambulance_To_Pass(patient),[]).
epistemicrule(f10,security(traffic_participants),[]).
epistemicrule(f13,sn(security(patient)),[]).
epistemicrule(f20,safety(traffic_participants),[]).
epistemicrule(f23,sn(safety(patient)),[]).
epistemicrule(f30,flowcontrol(traffic_participants),[]).
epistemicrule(f33,sn(flowcontrol(patient)),[]).
epistemicrule(f41,life(patient),[]).
epistemicrule(f42,sn(life(traffic_participants)),[]).

...
```

Fig. 5. CaSAPI description of reasoning process

In order to implement the decision making process, three basic types of rules were considered. First, the set of goal rules {r01: $passOnRed(X)$,r02: $waitForGreen(X)$,r03: $moveAside(X)$} was specified, each of which being a possible solution for the $react(X)$ predicate. This step is followed by a detailed evaluation of each of the possible decisions (decision rules),

[1] Available at http://www.doc.ic.ac.uk/~dg00/casapi.html
[2] Available at http://margo.sourceforge.net/

establishing the implied actions and the promoted (and/or demoted) values. For each decision rule, a unique identifier is set and further used to define an optional decision priority. We can observe that rule $r10$, representing the option of waiting for the Green light, corresponds to argument A_1 from the argumentation scheme (table 3), which promotes the values of traffic security ($t\_security(X)$), pedestrians safety ($p\_safety(X)$), flow control ($flowcontrol(X)$) and demotes the value of life ($sn(life(X))$. Similarly, rule $r20$ corresponds to argument A_6, promoting the value of life ($life(X)$) and pedestrians safety, while $r30$ corresponds to either of the arguments A_2 and A_4. Next, the last set of rules (epistemic rules) is specified, establishing for each of the decision criteria X ($X \in$ {patient, traffic_participants}) the corresponding actions and associated values, which will be further used in the justification of the selected action. The implementation of the reasoning process is presented in figure 5.

```
admissibleArgument(react(patient),P,S).
SENT= [d(patient), wn(del(f03)), wn(del(f13)),
      wn(del(f23)), wn(del(f33)), wn(del(f41)),
      wn(del(r02)), wn(del(r20(patient)))]
P = [passOnRed(patient)],
S = [d(patient)]
```

Fig. 6. Solution considering the best interest of the patient

```
admissibleArgument(react(traffic_participants),P,S).
SENT= [d(traffic_participants), wn(del(f00)), wn(del(f10)),
      wn(del(f20)), wn(del(f30)), wn(del(f42)), wn(del(r01)),
      wn(del(r10(traffic_participants)))]
P = [waitForGreen(traffic_participants)],
S = [d(traffic_participants)]
```

Fig. 7. Solution considering the best interests of the traffic participants

Finally, different solutions are suggested taking into consideration the best interest of the patient from the ambulance on one side (figure 6) and that of the traffic participants on the other side (figure 7).
We can observe that the solution proposed when considering the best interest of the patient as the main criterion ($X = patient$) is that of passing on the Red color of the traffic lights to allow this way the ambulance to enter the intersection (figure 6), while choosing to wait for the Green color of the traffic lights and not allowing the ambulance to pass is considered to be the best option when selecting as main criterion

the best interest of the traffic participants ($X = traffic\_participants$) (figure 7).

7 Discussion and Conclusions

In the conclusions of their paper [15], the authors present the following situation regarding the requirements for the deployment of normative systems.

> Providing explanations of norm violations is important if managers are to appropriately assign responsibility and apply sanctions. Explanations can also help to evolve the normative specification of a system in order to prevent future violations. Thus far, monitors provide limited explanation of violation and fulfillment, in terms of the labels of the arcs transitioned. Future work will investigate generation of more comprehensive explanations. This may require reference to representations of work-flow separate from that implicitly specified by norms. This information may in turn be gleaned from observers.

Our aim in this paper has been to provide a method that can show in a concise and precise manner whether some norms have been complied with or broken.

While some work has been concerned with mechanisms for the detection and resolution of normative conflicts [17], we have been focused here on proofs/justifications of agent behavior, by allowing them to make the proper decision considering the current environmental situation.

While work on adequately dealing with unpredictable and dynamic environments where normative frameworks are deployed, by providing mechanisms for modifying the norms at runtime [16] is very important, we concentrated on understanding the workings of norms. This is a significant aspect since otherwise the developers cannot know exactly how the norms should be changed.

Making norms concrete [1] relate the abstract concepts used in the specification to concrete ones used in practice. Our work is based on this approach extending it to proofs on how norms are being complied to.

Although we have used a very simple illustrative example, we are aiming at a general behavior of collaborating/competing agents, including interaction in normative organizations [7].

Considering the formalism for the monitoring of both regulative and substantive norms using monitoring [2] we feel that our approach can bring some light in this direction.

We are reasonably optimistic regarding the future work on better understanding how the approach presented in this paper could be exploited by offering more visualization of the behavior of agents in a normative setting.

Acknowledgments

We are grateful to the anonymous reviewers for the very useful comments. Part of this work was supported by the grant ID_170/672 from the National Research Council of the Romanian Ministry for Education and Research.

References

1. Aldewereld, H., Alvarez-Napagao, S., Dignum, F., Vazquez-Salceda, J.: Making norms concrete. In: 8th Int. Conf. on Autonomous Agents and Multiagent Systems. pp. 807–814 (2009)
2. Alvarez-Napagao, S., Aldewereld, H., Vazquez-Salceda, J., Dignum, F.: Normative monitoring: Semantics and implementation. In: 11th Int. Workshop on Coordination, Organization, Institutions and Norms in Agent Systems. pp. 1–16 (2010)
3. Areces, C., ten Cate, B.: Hybrid logics. In: Blackburn, P., Van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic. 821–868, Elsevier. Amsterdam (2007)
4. Artemov, S.: The logic of justification. The Review of Symbolic Logic 1, 477–513 (2008)
5. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13, 429–448 (2003)
6. Bench-Capon, T., Atkinson, K., McBurney, P.: Altruism and agents: an argumentation based approach to designing agent decision mechanisms. In: 8th International Conference on Autonomous Agents and Multiagent Systems. pp. 1073–1080 (2009)
7. Boissier, O., Balbo, F., Badeig, F.: Controlling multi-party interaction within normative multi-agent organizations. In: 11th Int. Workshop on Coordination, Organization, Institutions and Norms in Agent Systems. pp. 17–32 (2010)
8. Burgemeestre, B., Hulstijn, J., Tan, Y.H.: Value-based argumentation for justifying compliance. In: 10th International Conference on Deontic Logic in Computer Science. pp. 214–228. Springer-Verlag (2010)
9. Fitting, M.: The logic of proofs, semantically. Annals of Pure and Applied Logic 132, 277–322 (2005)
10. Fitting, M.: Justification logics and hybrid logics. Journal of Applied Logic 8, 356–370 (2010)
11. Gaertner, D., Rodriguez, J.A., Toni, F.: Agreeing on institutional goals for multi-agent societies. In: 5th Int. Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (2008)
12. Horridge, M., Parsia, B., Sattler, U.: Justification oriented proofs in OWL. In: 9th International Semantic Web Conference. pp. 354–369. Springer-Verlag (2010)
13. Jones, A.J.I., Sergot, M.: A formal characterisation of institutionalised power. J. of the IGPL 4, 427–443 (1996)
14. Modgil, S.: Value based argumentation in hierarchical argumentation frameworks. In: Conference on Computational Models of Argument. pp. 297–308. IOS Press (2006)
15. Modgil, S., Faci, N., Meneguzzi, F., Oren, N., Miles, S., Luck, M.: A framework for monitoring agent-based normative systems. In: 8th Int. Conf. on Autonomous Agents and Multiagent Systems. pp. 153–160 (2009)
16. Tinnemeier, N., Dastani, M., Meyer, J.J.: Programming norm change. In: 9th Int. Conf. on Autonomous Agents and Multiagent Systems. pp. 957–964 (2010)

17. Vasconcelos, W.W., Kollingbaum, M.J., Norman, T.J.: Normative conflict resolution in multi-agent systems. Autonomous Agents and Multi-Agent Systems 19, 124–152 (2009)

Multi-agent Coordination Through Mutualistic Interactions

Miguel Lurgi and David Robertson

School of Informatics, University of Edinburgh
10 Crichton Street. EH8 9AB, Edinburgh, UK.
miguel.lurgi@ed.ac.uk
dr@inf.ed.ac.uk

Abstract. In this paper we present an ecologically-inspired approach to agent coordination. Mutualistic networks of interacting species in nature possess characteristics that provide the systems they represent with features of stability, minimised competition, and increased biodiversity. We take inspiration from some of the ecological mechanisms that operate at the interaction level in mutualistic interactions, and which are believed to be responsible for the emergence of these system level patterns, in order to promote this structural organisation in networks of interacting agents, enhancing in this way their cooperative abilities. We demonstrate that given plausible starting conditions, we can expect mutualistic features to appear in self-organising agent systems, and we compare them with natural ones to show how the characteristics displayed by ecologically inspired networks of agents are similar to those found in natural communities. We argue that the presence of these patterns in agent interaction networks confer these systems with properties similar to those found in mutualistic communities found in the real world.

Keywords: agents coordination, ecologically-inspired interactions, complex systems, mutualism, emergent behaviour, cooperation

1 Introduction

Complex systems of interacting entities are ubiquitous in nature and the artificial world, ranging from networks of interacting computers in cyberspace, different types of transportation networks such as airports and the links connecting them, power grids that provide cities with electricity, human interactions of different kinds such as researchers and their collaborations; to the highly organised webs of interactions that we find in biological systems such as protein, gene, cell, and neural networks, and the relationships found in real communities in a given ecosystem, where species form complex but organised collections of interacting entities.

In societies of artificial agents we are sometimes interested in promoting long-lasting and complex relationships of the kinds described above in order to accomplish difficult tasks that are achieved more easily in a cooperative way.

In nature, the process of evolution by natural selection has produced different types of interactions between organisms composing communities, which together are the mechanisms responsible for the formation and maintenance of the entangled web of life found in ecosystems.

From the range of interactions found in natural communities, we are mainly interested in mutualisms: interactions in which both of the species involved benefit from participating on them. This kind of relationship allows for the creation of relations amongst species that facilitate each others survival. It has been argued that this kind of relation can enhance the stability of the community the interacting species belong to [2] and promote biodiversity within it [3].

In this paper, we explore the extent to which relationships of this kind amongst agents in artificial systems can promote the stability and persistence displayed by communities of interacting species within natural ecosystems.

Ecologically inspired approaches have been applied to the design and development of multi-agent systems in order to promote features such as self-organisation and adaptation of agents in open, digital environments. Some of them have even taken inspiration from the way interactions occurr among species in nature; as in [16], where the authors propose a "food-web" based approximation. In this work we go beyond this kind of approaches by deepening in the level of detail for the definition of interactions among agents, based on ecological mechanisms believed to account for important system-level properties.

We describe a methodology for the specification of protocols of interactions for enabling relationships amongst artificial agents living in a multi-agent system based on ecological concepts, and show how the organisation of the interactions among these agents is similar in many ways to the patterns encountered when analysing ecological networks in general and mutualistic webs in particular.

We argue that these specifications at the interaction level will enable our intelligent systems with the features encountered in their natural counterparts, and which undermine the adaptability, persistence, robustness, and stability found in those complex natural networks of individuals.

2 Mutualistic Networks

A focus of research in ecology is based on the characterisation of different kinds of interactions that are observed amongst individuals within ecological communities. The study of the patterns and structure of these interactions and their implications for community organisation and persistence has been tackled through the application of concepts borrowed from the mathematical fields of graph and network theory, giving rise to the ecological discipline of ecological networks.

Studies of this kind performed on natural communities have shown that these systems are generally characterised by small world patterns [9], contributing in this way to fast propagation of the information across the networks; truncated power law degree distributions [8], a feature characteristic of scale-free networks, with a small number of nodes possessing degrees greatly exceeding the average (usually referenced as *"hubs"*) that are believed to be the strength and, at the

same time, the weakness of this kind of networks; and the importance of weak interactions for the maintenance of the overall community [4].

A particular type of ecological network, in which the interactions depicted represent mutualistic relationships among the species in the community, are mutualistic networks. Research on this type of ecological webs has demonstrated that they share some of the patterns and features described above with other kind of ecological networks [2], as for example their heterogeneity or scale-free character; but at the same time are different in many respects, exhibiting properties that are specific to these kind of networks: with generalist species interacting with proper subsets of the species with which more generalist species interact [2], and being pervasively asymmetric in the links between species [14].

The features displayed by mutualistic networks of interactions, and their architecture, are believed to facilitate biodiversity maintenance and to minimise competition in this kind of systems [3]; features that are desirable in agent systems in which we would like to promote organisational abilities.

It has been demonstrated in real ecosystems, and particularly in mutualistic communities, that biological and ecological processes describing the interactions between species within them, are ultimately responsible for the structure and patterns that we observe in these mutualistic networks of relationships [15].

We are particularly interested in those mechanisms because in this work we have taken inspiration from some of the processes characteristic of mutualistic interactions, such as for example: *trait matching*, *habitat* occupation (i.e. spatial distribution), or the formation of *meta-communities*; to define the protocols of interactions among agents in agent systems, with the objective of promoting the emergence of collaborative links among them and system level properties that allow for its stability and persistence. We want our societies of artificial agents to display some of the features we encounter in ecological networks of self-organised, autonomous individuals.

Thus, the way mutualistic species interact in nature gives us useful insights about the way artificial intelligent entities may interact in a cooperative digital environment in order to form organised and structured societies.

3 Methods

Since the focus of interest in this research is the design of interactions among agents in a multi-agent system, we adopted an interaction centred approach for the implementation of the system on which to develop the ideas presented in this paper.

For the implementation of a multi-agent system based on protocols for the formalisation and description of the interactions among the entities sharing our digital environment, we use the OpenKnowledge system [5]; a peer-to-peer (P2P) based system that makes use of the Lightweight Coordination Calculus (LCC) [11] for the specification of the interaction models among autonomous agents which communicate through a peer-to-peer (P2P) network.

For describing interactions among agents in multi-agent systems we focus on mechanisms believed to account for system-level patterns in natural communities [15]. Based on key concepts in ecology such as habitat, meta-communities, traits, and resources, we focus in the mechanism of *"trait matching"* and take inspiration from how species in nature are bound together via complementary phenotypic traits to propose a method for describing interactions between artificial agents within a multi-agent system.

These features, commonly employed to describe interactions in natural communities, will help us not only to define interactions between our agents, but also to design the internal structure of the agents themselves for coping with their interaction partners.

Protocols for agent interaction coordination in LCC are based on the definition of roles, which are in turn composed of clauses specifying the actions to be performed by the agents taking part in a particular interaction, when they embark upon any of its instances, by assuming one of the roles available in it.

We defined the roles of "visitor" and "host" inside our protocol of interaction in resemblance to the actors taking part in plant-animal mutualistic interactions in nature. Appendix A shows the definition of the visitor role in our ecologically inspired interaction protocol, which is written in LCC.

Agents implementing the roles mentioned interact through the exchange of a series of messages (generated via local constraint solving) in which they exchange and process information about the habitat they occupy within the environment, the size of the trait through which they are interacting in order to obtain a complementarity degree which will partially determine the actual occurrence of the interaction, their ability of forming meta-communities (interacting with agents in different habitats than their own), and the amount of resource they are going to exchange if the interaction is successfully completed. This can be seen in the extract of our interaction protocol shown in Appendix A.

During the execution of the interaction protocol, and through the exchange of messages and information as explained above, agents are able to gather information from other agents with which they interact and are allowed to quit the interaction at certain points when they consider it appropriate, either because the ecological constraints posed by the protocol are not met or simply because agents decide not to continue on that particular interaction.

Our model has a number of parameters and settings that can be configured in order to produce different outcomes given different circumstances for the agents to interact. Our default general settings however, define certain values that are common to all of the simulation results presented. The main settings are:

1. number of agents: in resemblance to the proportion of plants versus pollinators in plant-animal interaction networks, where there are generally more plant species than animal ones, we set the number of host agents in our agent systems to ten and that of the visitors to fifteen.
2. number of habitats: the habitat a given agent occupies will partially determine whether it will be able to interact with the agent selected for interaction; they will be more likely to interact if in the same habitat than

otherwise. In our experiments the default number of habitats employed is two, and agents are evenly distributed among both habitats.

3. minimum and maximum resources: any given interaction is based around the exchange of resources that will ultimately allow agents to survive; at the beginning of each run, agents are given a certain amount of resources that is selected randomly from a normal distribution bounded by a minimum and maximum amount of resources. By default the range for this amount is between five (min resource) and twenty (max resource). Although the initial amount of resource may determine the fate of an agent, these numbers were selected in order to ensure stability in resource distribution.
4. meta-community formation: agents possess the ability of interacting with individuals belonging to different habitats than their own, when this happens they are said to form a meta-community. The probability of any given agent forming a meta-community association is 0.1.
5. maximum trait size: in our model we have a parameter specifying the maximum size of the interaction trait, and all the agents will have sizes ranging from zero to this maximum size. The default value for the max size of traits in our system is five. The size of a trait will determine the extent to which partners in an interaction will be complementary to one another, thus, this parameter is a measure of overlap of features (the closer the size of the traits on complementary partners the more likely the will be able to interact)

Sensitivity analyses have been performed on the initial values of the parameters described above in order to ensure that different configurations of these parameters do not produce a noticeable variation on the behaviour of the system (data not shown [7]).

The analyses which follow are based on multiple series of simulations, configured as described above, with initially random encounters between agents. These allow us to generate networks of interactions for analysis.

Our study of the networks obtained in this way takes advantage of a set of metrics borrowed from the field of ecological networks and which are commonly used to analyse this type of complex networks of interactions:

1. connectance ($C = L/S^2$): the connectance of the network gives a quantitative value of the proportion of links that are actually realised (L is the number of links in the network) relative to the total number of possible interaction links (S is the number of nodes) in the network.
2. frequency distribution of the number of interactions: represents the distribution of the frequencies with which we encounter a node with a certain number of interactions.
3. degree distribution: the degree of a node is the number of links it has. The distribution of degrees is a good indicative of the extent to which a given network possesses scale-free features.
4. interactions matrix and nestedness index: the interactions matrix lets us see the relations happening between pairs of agents. Based on these interactions we calculate the isocline of perfect nestedness, as introduced in [12], which

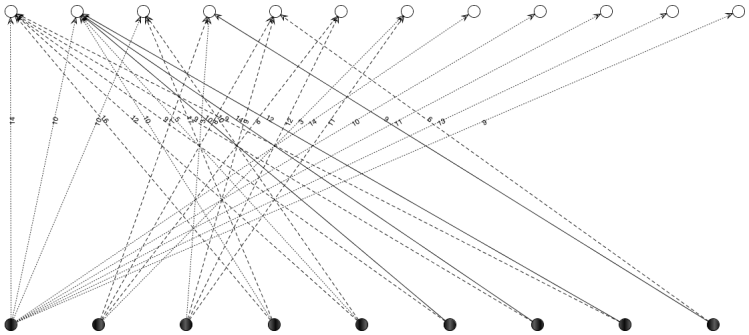


Fig. 1. Example of the output of a simulation run performed using the model described in this paper. A network of interactions between agents in a multi-agent system.

by taking into account the number of interactions occurring in the network obtains an estimate of a perfectly nested matrix and gives a curve that allows us to visualise the pattern that we should expect from it; this curve, plotted on top of the actual interactions matrix gives an idea about how nested our matrix is. Additionally, we calculate the Nestedness metric based on Overlap and Decreasing Fill (NODF), as presented in [1], and which is a metric commonly used in the analysis of nestedness in ecological networks [6] for determining the extent to which a given interaction network presents a nested pattern.

4 Results

Our experiments consisted of the execution of a series of independent simulation runs following the specifications outlined above for parameter initialisation and run configuration. During these runs, relationships among pairs of agents arose with different strengths (the number of times an agent interacts with any other relative to the number of times it has interacted during the entire simulation) and with different configurations.

4.1 Ecologically Inspired Mutualistic Agents Networks

Figure 1 shows an example, taken from one of the runs in our experiments, where the relationships among agents in our multi-agent system are represented in a fashion similar to networks of interacting species described in the field of ecology.

In this graph, the agents and their relationships with other agents are represented by nodes and arcs respectively. A link (arc) is generated between two agents whenever an interaction is successfully completed amongst them.

By representing the relationships between the agents in agent systems in this way we are able to extract features, obtain descriptors, and perform analyses over the resulting network based on methods borrowed from network theory.

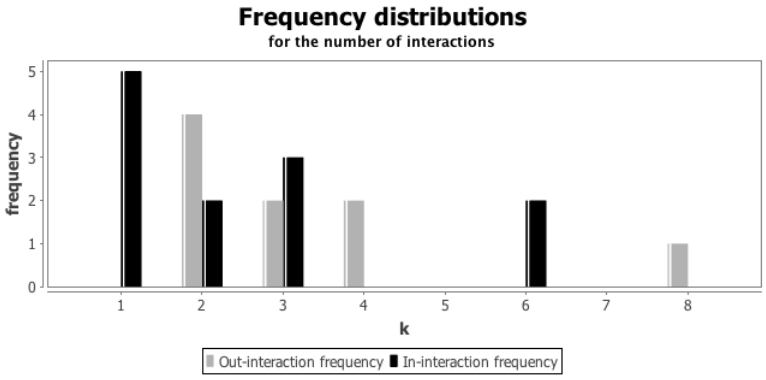


Fig. 2. Frequency distribution of the number of interactions for the network in Fig. 1.

We obtained networks that display a scale-free structure: the plot displayed in figure 2 shows data from the network in figure 1, where it can be observed that the majority of nodes in this network have small degree (≤ 2), while a low fraction of them are highly connected, showing a distribution of the frequency of interactions biased towards low values (1 and 2 interactions). Additionally, small-world properties are found in our networks: with short paths between any two nodes, a property commonly found in ecological networks.

Properties of the kind mentioned above, which are encountered in the networks of interactions among our agents, are common patterns also found in different kinds of complex networks in nature and the artificial world [13], and which differ significantly from the structure that we would expect from a randomly assembled network.

Another property seen in our networks, which is related to their scale-free character, is the preferential attachment displayed by visitor agents with low degrees (e.g. the four white nodes on the top right corner of figure 1) to host agents that are highly connected. This is a common feature encountered in mutualistic networks, where specialist species are more likely to interact with generalists [2]. Patterns of this kind are important in practice because, as has been argued, they can give us information about functional properties of the system such as: information propagation speed and resistance to node failures [13], which in turn provide us with a better understanding of the relationship between the complexity and stability of our agent systems.

Particularly, in the realm of ecological networks, asymmetric specialisation (i.e. a specialist interacting with a generalist) has been found to be a pervasive feature of plant-pollinator interaction networks, and it is believed to be beneficial for the majority of species in these communities because it facilitates the avoidance of extinction risks when species are highly reciprocally specialised [14].

The network architecture displayed by the interactions amongst agents is an emergent property of our system since the only mechanisms involved in agents'

Sim	C (L/S^2)	NODF	Emp	C (L/S^2)	NODF
1	0.053	25.203	SAPF	0.031	18.36
2	0.062	37.063	CACO	0.034	29.693
3	0.064	32.456	CAFR	0.039	34.166
4	0.066	28.667	SCHM	0.041	56.66
5	0.067	36.132	MOMA	0.045	32.067
6	0.068	33.333	GEN1	0.061	34.243
7	0.07	42.913	OFLO	0.062	35.961
8	0.077	61.06	BAIR	0.064	50.98
9	0.079	44.409	ESKI	0.071	54.586
10	0.079	51.404	BEEH	0.074	67.66
11	0.083	54.82	WYTH	0.075	45.411
12	0.083	70.102	KANT	0.084	67.344
13	0.086	68.528	LOPE	0.103	57.423
14	0.087	59.211	HRAT	0.111	78.756
15	0.087	63.739	FROS	0.163	74.571

Table 1. Connectance and NODF values of the fifteen simulated and fifteen empirical networks analysed in this paper.

interactions are those specified by the protocol presented in section 3. The creation of such a complex and intricate pattern of relationships is not a hardwired property of the artificial communities, but rather the product of many different agents interacting together for achieving their respective goals (gathering resources to survive).

4.2 Comparing with Empirical Data

In order to test the extent to which the networks of interactions found in our artificial systems are similar to mutualistic networks of interactions found in the real world (apart from the qualitative similarities introduced above), we have compared the architecture of the networks obtained from our simulations to empirical data of networks collected from real communities and that have been compiled, analysed, and provided as supplementary material by Rezende et al in [10]. Although in that paper they used the networks for different kinds of analyses, the datasets provided are useful for getting an idea of the common features encountered in mutualistic networks.

Because some of the properties of interest for analysing ecological networks are scale dependent, we have selected fifteen networks from this dataset, based on the number of species composing it and that were more close to the number of agents in our simulations, for comparison against our fifteen simulated networks.

As mentioned in section 3, connectance ($C = L/S^2$), the fraction of all possible links that are realised in a network, is an important property that is commonly employed in the analysis of ecological networks and which provides with information about the degree of connectivity between the nodes of the network. We use the connectance and the NODF index of nestedness, introduced in section

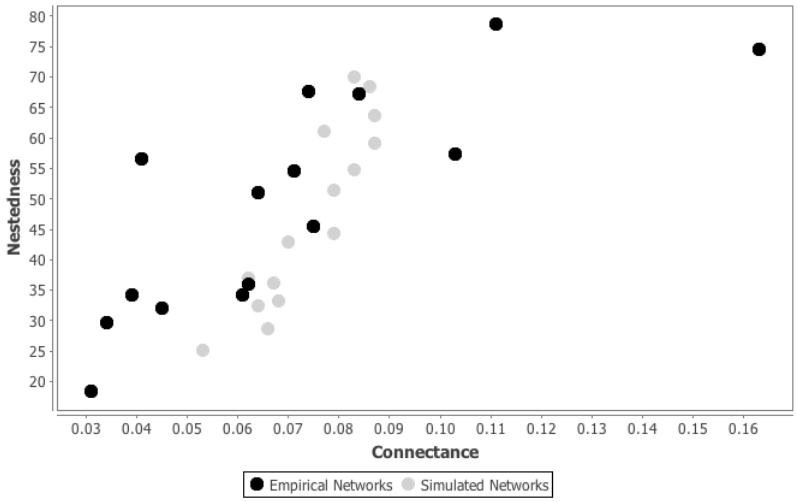


Fig. 3. Connectance versus NODF nestedness values in the natural (black dots) and simulated (grey dots) communities presented in Table 1.

3, for comparing our simulated networks with fifteen empirically obtained plant-animal mutualistic networks. Additionally we perform qualitative comparisons between the structures obtained in one of our networks and one of the natural communities considered for this analysis.

In table 1 we can see the connectance and NODF values derived from our simulated networks and the selected empirical networks obtained from natural communities; the names of the empirical networks are as presented in [10], and the numbers used to identify the simulated networks are only for id purposes and are not related to any feature of the network itself. Figure 3 displays the plot of the NODF against the connectance values with the double purpose of analysing the behaviour of nestedness in relation to changes in the connectance of the network, and to compare these relationships in our simulated communities against their natural counterparts.

As we can see, in both types of networks there is a positive relationship between the connectance of the network and the nestedness value. This is not surprising, since the more connections a network has the more they can contribute to the nested diets observed in those networks. It is perfectly possible however, that more links could mean less nested communities (if they go to the wrong side of the perfect nestedness isocline), so, it is important to bear in mind that in well organised communities, like the ones we consider in this work, the more connected networks are, the more nested they become.

We can also appreciate from this plot that the data points corresponding to the naturally occurring communities are more distributed through the graph, while the points representing our simulated networks are more concentrated

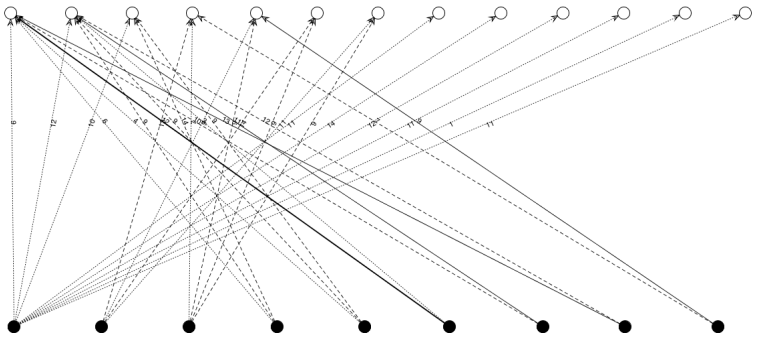


Fig. 4. Network representation of community number 3 presented in Table 1.

around connectances between 0.05 and 0.09, and nestedness indexes of 25 and 70. Apart from telling us about the variability encountered in real communities in this respect, this reinforces the fact that our simulated communities, although differing in some aspects from run to run, share a similar structure and enjoy features seen in natural communities.

These data (table 1 and figure 3) also show us that the values of connectance and nestedness obtained from our simulated communities agree with the values of these measures commonly found in natural communities, where the connectance is normally between 0.03 and 0.1, and the nestedness values are usually found between 20 and 80. In our communities, the connectance values were greater than 0.05 and lower than 0.1; similarly, the NODF values for our communities were distributed along the 25-80 range of values. This provides more evidence for the self-organised character of our digital societies, a distinctive feature inherited from their biological peers.

4.3 Structural Similarities between Ecological and Agents Networks

We want to further explore the similarities between our agent systems and natural communities; for this we have selected, from the set presented above, one of our simulated and one of the empirically observed systems to perform a one to one comparison.

Two networks were thus selected based on their similitudes not only in connectance and nestedness values but also taking into account other features, as we shall see in the following paragraphs. The networks selected were: the one represented by the number 3 in table 1, with values 0.064 and 32.456 of connectance and NODF index for nestedness respectively, for representing our simulated communities; and the OFLO natural community, with 0.062 connectance and 35.961 NODF index, as a representative of the natural communities considered.

Figures 4 and 5 show the network of relationships between entities in the simulated community number 3 and the OFLO natural community, respectively.

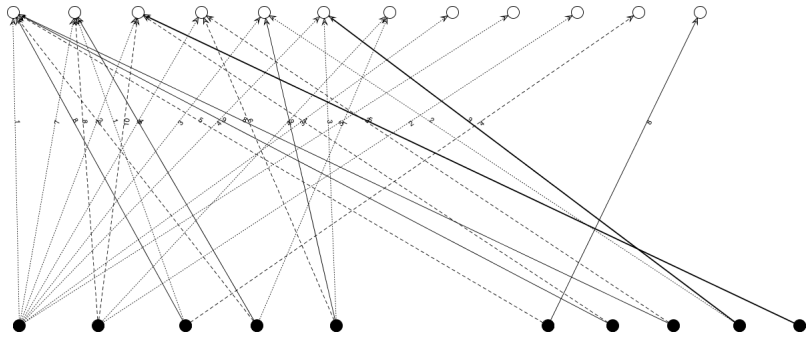


Fig. 5. Network representation of the OFLO natural community presented in Table 1.

When closely inspecting these two networks we can easily realise a number of broad similarities such as the high proportion of visitor nodes (nodes in white) possessing only one interaction and also an important fraction of host nodes (nodes in black) with two interactions or less. Also notable is the presence of one highly connected host node in both of the networks (black node in the bottom left corner in each of the networks), and also on the side of the visitor species/agents (white node on the top left). These nodes act as generalists/hubs, bringing cohesion and reachability to the whole network.

Another important feature captured by looking at the network of interactions and that is shared by our natural and artificial communities is the low fraction of strong dependences among nodes (solid dark edges in the graph) and the abundance of weak dependences, which in natural communities is believed to account for the stability and resilience of these systems, since the loss of a link can be easily adjusted for by recurring to other connections in the network. In agent systems, this property can be translated into the ability of agents to quickly adapt to missing links and not to depend strongly in any other agent in the network, facilitating in this way the persistence of each individual agent and the system as a whole.

Apart from the graphical representation of the interactions network, we also analyse certain properties derived from its structure and that can serve for deepen our comparison among the selected networks.

Natural communities, as we have seen above, are somewhat less predictable than our simulated ones. This can be confirmed when comparing the interactions matrices of our simulated community number 3 and the OFLO natural community (figures 6 and 7 respectively): community number three presents a much more organised structure, with few interactions below the isocline of perfect nestedness, while the OFLO community presents not only more interactions below the isocline but also a few of them are actually far removed from it. Also the degree distributions in both communities, although agreeing in the over-

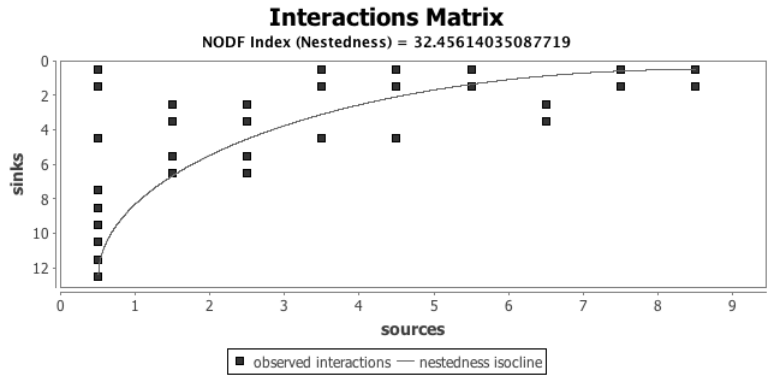


Fig. 6. Matrix of the interactions found in community number 3 in our simulations (including NODF index and nestedness isocline).

all scale-free pattern, differ subtly by the fact that the values in our simulated community are closer to the fitted power law (data not shown).

In spite of these differences, the communities present very similar distributions of the frequencies of the number of interactions (data not shown), with practically all the nodes possessing less than five interactions and the majority of them with two or one (this can be appreciated in the graphs representation of the networks). Additionally, only three nodes in the case of community number 3 and two in the case of the OFLO community possess more than five interactions. Reaffirming in this way the scale-free character present not only in the natural communities employed here as reference, which is expected from this type of networks, but also in our simulated communities as an emergent feature of the self-organisation in our agent systems.

The similarities found between natural communities and our simulated ones, not only in terms of the network structure, but also in terms of their features and characteristics, let us see how the mechanisms implemented at the individual interaction level between our artificial agents has allowed for the development of mutualism through agent interaction; obtaining system level properties that are commonly encountered in natural communities of mutualistic species, and which in turn, as has been argued throughout this paper, can provide our self-organising agent societies with stability and robustness.

By comparing the structure of the networks of relationships resulting from our simulations, with the networks of interactions amongst mutualistic species in nature, we can translate the results of the research performed until now in the field of ecology in general, and ecological networks in particular into the domain of multi-agent systems. The characteristics described above, which are shared between our networks of artificial agents and those observed between species in the natural world, let us conclude then that the features conferred to these natural systems by the characteristics displayed by their networks of interactions

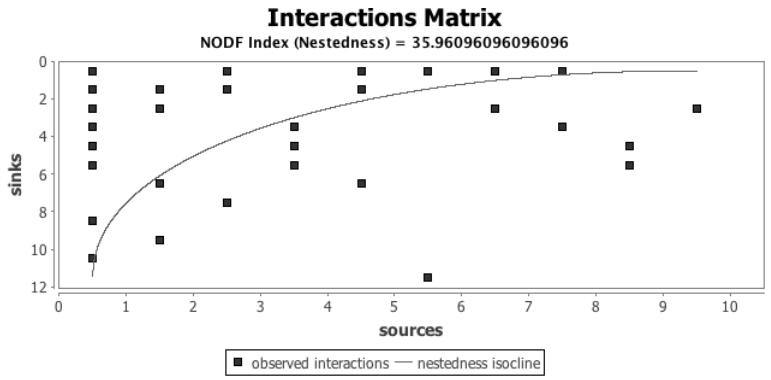


Fig. 7. Matrix of the interactions found in the OFLO natural community (including NODF index and nestedness isocline).

are also given to our agent systems thanks to the emergence of a similar network structure to that found in these ecological networks.

5 Conclusion

The ecologically inspired techniques for agent coordination and communication presented in this work allow societies of autonomous agents to form self-organised complex networks of interactions that benefit from many of the features encountered in ecological networks of interactions among mutualistic partners in nature.

Since the systems from which we have taken inspiration are composed of mutulistic relationships arising in nature, and by equating the patterns encountered in the network of interactions between agents in our simulated communities to those seen in their natural counterparts, our systems also benefit from the properties of increased biodiversity and minimised competition characteristic of these type of natural systems [3], which promotes organisation and collaboration between the artificial entities in our intelligent systems.

To our knowledge no attempts have been made so far to develop agent interactions inspired directly by ecological theory, which we think is a fruitful research area. Furthermore, in the field of multi-agent systems, little effort has been devoted to the analysis of this kind of systems from the point of view of complex systems. The tools provided by the field of complex networks in general and ecological networks in particular, and that we have used as part of our study, constitute then, as far as we know, one of the first attempts to analyse complex networks of agents interactions in multi-agent systems from a complex systems perspective.

6 Acknowledgments

This work is part of the project "EcoBusiness: A Multi-Agent Approach for Digital Business Ecosystems", funded by the European Commission through the Seventh Framework programme Marie Curie Actions - Industry-Academia Partnerships and Pathways (IAPP). Grant agreement no.: 230618.

References

1. M. Almeida-Neto, P. Guimarães, P. R. Guimarães Jr., R. D. Loyola, and W. Ulrich. A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos*, 117(8):1227–1239, August 2008.
2. J. Bascompte and P. Jordano. Plant-animal mutualistic networks: the architecture of biodiversity. *Annu. Rev. Ecol. Evol. Syst.*, 38:567–593, December 2007.
3. U. Bastolla, M. A. Fortuna, A. Pascual-Garcia, A. Ferrera, B. Luque, and J. Bascompte. The architecture of mutualistic networks minimizes competition and increases biodiversity. *Nature*, 458(7241):1018–1020, April 2009.
4. E. L. Berlow. Strong effects of weak interactions in ecological communities. *Nature*, 398(6725):330–334, March 1999.
5. A. P. de Pinnick Bas, D. Dupplaw, S. Kotoulas, and R. Siebes. The openknowledge kernel. *International Journal of Applied Mathematics and Computer Sciences*, 4(3):162–167, 2007.
6. L. N. Joppa, J. M. Montoya, R. V. Solé, J. Sanderson, and S. L. Pimm. On nestedness in ecological networks. *Evolutionary Ecology Research*, 12(1):35–46, January 2010.
7. M. Lurgi. Ecologically inspired agent networks. Master's thesis, University of Edinburgh. College of Science and Engineering. School of Informatics., 2010.
8. J. M. Montoya, S. L. Pimm, and R. V. Solé. Ecological networks and their fragility. *Nature*, 442(7100):259–264, July 2006.
9. J. M. Montoya and R. V. Solé. Small world patterns in food webs. *Journal of Theoretical Biology*, 214(3):405–412, February 2002.
10. E. L. Rezende, J. E. Lavabre, P. R. Guimarães Jr, P. Jordano, and J. Bascompte. Non-random coextinctions in phylogenetically structured mutualistic networks. *Nature*, 448(7156):925–928, August 2007.
11. D. Robertson. A lightweight coordination calculus for agent systems. *Lecture Notes in Computer Science*, 3476:183–197, July 2005.
12. M. A. Rodríguez-Gironés and L. Santamaría. A new algorithm to calculate the nestedness temperature of presence–absence matrices. *Journal of Biogeography*, 33(5):924–935, May 2006.
13. S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.
14. D. P. Vázquez and M. A. Aizen. Asymmetric specialization: a pervasive feature of plant-pollinator interactions. *Ecology*, 85(5):1251–1257, May 2004.
15. D. P. Vázquez, N. Blüthgen, L. Cagnolo, and N. P. Chacoff. Uniting pattern and process in plant-animal mutualistic networks: a review. *Annals of Botany*, 103(9):1445–1457, March 2009.
16. C. Villalba and F. Zambonelli. An nature-inspired approach for large-scale pervasive service ecosystems. In *Proceedings of the 4th AAMAS Workshop on Massive Multiagent Systems*. IFAAMAS, May 2009.

Appendix

A. Definition of the "visitor" role in our ecologically inspired interaction protocol (written in LCC)

```
01 a(visitor, X)::
02     null <- chooseHost(Y, ListOfHosts)
03     then
04     whereabout => a(host, Y)
05     then
06     ( in(HabitatHost) <= a(host, Y)
07       then
08       ( ( ( ( null <- sameHabitat(HabitatHost)
09               then
10               whichtrait => a(host,Y))
11             or
12             ( null <- metaCommunity()
13               then
14               whichtrait => a(host, Y)))
15           then
16           availabletrait(Trait) <= a(host, Y)
17           then
18           null <- haveTrait(Trait)
19           then
20           whichsize => a(host, Y)
21           then
22           size(TraitSize) <= a(host, Y)
23           then
24           ( null <- complementary(TraitSize)
25             then
26             null <- need(Amount, Reward)
27             then
28             exchange(Amount, Reward) => a(host, Y)
29             then
30             offer(Offered) <= a(host, Y)
31             then
32             null <- consume(Offered, Reward)))
33         or
34         ( quit => a(host, Y))))
35     then
36     a(visitor, X)
```

Where *null* denotes an event which does not involve message passing; the operator *::* is used to declare the definition of a role within the protocol; and the operators $<-$, *then* and *or* are connectives for logical implication, sequence and choice respectively. $M => A$ denotes that a message, M, is sent out to agent A. $M <= A$ denotes that a message, M, from agent A is received.

MANET: A Model for First-Class Electronic Institutions

Charalampos Tampitsikas[1,2], Stefano Bromuri[2], and Michael Ignaz Schumacher[2]

[1] Università della Svizzera italiana,
Communication Sciences Department,
via G. Buffi 13, 6900 Lugano, Switzerland,
charalampos.tampitsikas@usi.ch
[2] University of Applied Sciences Western Switzerland,
Institute of Business Information Systems, 3960 Sierre, Switzerland
{stefano.bromuri, michael.schumacher}@hevs.ch

Abstract. In this paper we present a new approach to model electronic institutions that are situated in agent environments where heterogeneous agents reside. An electronic institution is seen here as an entity that is deployed within an environment infrastructure that directly mediates the agents interaction. The environment allows the rules of the institution, in terms of powers, permissions and obligations to be perceivable as first class entities by the agents belonging to the institution. We express institutions as first class abstractions that can be inspected, manipulated and modified, created and destroyed by the agents populating the agent environment where the institution resides. To represent the institution we utilize the Object Event Calculus (OEC) formalism that deals with the evolution of complex structures in time and we extend it to deal with the mediation of the events and with the perception of complex structures and events within institutions. We use an e-Health marketplace based on dutch auctions to illustrate the properties of our model.

Keywords: normative systems, multi-agent systems, electronic institutions, logic programming, agent environments

1 Introduction

In the Web 3.0 [14], human beings and software applications freely interact to carry out complex activities, inclusive of (but not limited to) e-business and e-government applications. People and organizations delegate many of their tasks to software applications, called agents. An agent is considered an autonomous entity which observes and acts upon an environment and directs its activity towards achieving its goals [25]. These agents behave as representatives acting both reactively and proactively in their principal's interest while they are also empowered to carry out tasks that have legal effects, like signing contracts and performing business transactions.

Many Web 3.0 applications can be defined as complex *open multi-agent systems* (MASs) [13]. A MAS can be considered open [18] when it satisfies the following properties: i) agents are free to join and leave at any time and ii) agents are designed by and represent different stakeholders with different objectives.

Due to the properties of open MASs, a set of issues must be considered [4]: an open MAS is by definition dynamic as the agents may join or leave at any time; it is insecure as an agent may be programmed to be malicious; it is not deterministic as no agent can have a global knowledge of the system; and finally it has not a central authority. Normative systems, or as Ägotnes et al. specify in [1], systems were social rules apply, try to tackle these issues by defining rules to coordinate heterogeneous agents.

Electronic institutions (EI) [12] are an approach to normative systems, containing a constitutive and regulative part [5]. We can regard an EI as a means for imposing a well-defined structure to the social reality within which agents interact [20], based on a set of rules that mediate the interaction taking place between the agents.

However, EIs suffer from a number of drawbacks. First of all, despite the fact that normative systems provide a level of abstraction in terms of social rules amongst agent societies, it is not clear how these rules mediate the interaction in a MAS in terms of concrete mechanisms. Moreover, the governor agent approach suggested by some enforcement based normative systems [10] has the disadvantage of mixing the concept of infrastructure with the concept of agent, implying that everything in the system is represented as a communicating agent, even when encapsulating low level reactive resources, resulting in quite computationally expensive applications. An example of a model where everything inside the MAS is considered as a communicating agent is the framework proposed by Campos et al. [7] as an extension to electronic institutions.

Secondly, EIs lack of mechanisms that allow the perception of institutional entities and events. The perception of an EI could allow the agents to decide whether participating in the institution would benefit the accomplishment of their goals. Thirdly, current research on normative systems is mainly focused on the communication events inside the system. While communication events are of great importance, they are not sufficient to describe all the possible interactions of the institutional entities and they cannot fully describe the evolution of the system.

To avoid the three drawbacks described above, we present a meta-model which considers the notion of EI as the social constitutive element of an agent environment. In particular, our contribution is to provide a model that proposes the following solutions to the current drawbacks of EIs: i) we introduce the concept of institutional space as the mediator of the social interaction between agents in the agent environment; ii) we propose a perception model to observe institutional spaces and their norms; iii) we present an event system to handle the evolution of institutional spaces and of MASs related to these institutional spaces. We illustrate the perception properties of these concepts by means of an e-Health marketplace example. Although the perception of norms implies that agents can interpret them, for the purposes of this paper we focus only on the social interactions of the agents.

The remainder of this paper is structured as follows: Section 2 is a description of the main properties of agent environment that we have to take into consideration to define our EI meta-model; Section 3 presents our meta-model of first-class Electronic Institutions; Section 4 shows how we apply our model within an e-Health market place based on Dutch auctions; Section 5 shows sketches of our implementation in Prolog; Section 6 puts our work in comparison with existing EI frameworks; finally Section 7 concludes this paper and shows some possible future work directions.

2 Normative Systems and First-Class Agent Environments

Although there is not a clear definition of the agent environment in the traditional normative system models, Esteva in [9] was the first to mention that EIs can be considered as an effort to shape the environment where the agents are situated, offering the agents the conditions to exist and interact. Usually, in the literature, the environment is considered as a domain-specific infrastructure for agents while its main responsibility is the objective coordination of the agents [24].

The agent environment can be used as a first-class abstraction that mediates the interaction between agents taking part in a distributed MAS. First-class abstraction means that the environment is an independent component inside the MAS structure that has its own responsibilities irrelevant to the goals of the agents. According to Weyns [24], the agent environment as a first-class entity can offer four different levels of support:

Basic level: at this level the environment enables agents to access to the *deployment context*. By deployment context, it is meant the external resources with which the MAS interacts (e.g. printers, databases and Web services).

Abstraction level: at this level the environment shields low-level details of external resources defining a standard interface that the agents can access from the environment.

Interaction-mediation level: the interaction mediation level offers support to regulate the access to resources and to mediate the interaction between agents.

Reflection level: The environment supports the modification of its composition and function during runtime. The agents can perceive the properties of the environment and interact in order to modify its state.

A distributed implementation of the model of agent environment proposed by Weyns is represented by the GOLEM agent platform [6]. One of the drawbacks of GOLEM is that it does not model the social interaction amongst the agents, handling it in an ad-hoc manner according to the application, limiting the reusability of the agents and of the infrastructure. Based on the basic level of support proposed by GOLEM, we want to use the abstraction and mediation level of support provided by the agent environment to offer a solution to the main drawbacks of electronic institutions as presented in the previous section and to provide a reusable social interaction model for agent environments. For the purposes of this paper, we study only the environment perception provided by the reflection level and we do not consider run-time modification of its laws.

In order to embed these levels of support into electronic institutions, we first need to specify the appropriate type of normative systems we will use for the mediation of agent interactions. There are two approaches to define normative systems [17]: a) *regimentation based normative systems*, in which a set of rules and protocols are defined to coordinate the behavior of the agent; b) *enforcement based normative systems*, in which some of the agents in the open MAS have the role of regulator agents enforcing the rules when they discover they have been violated.

Regimentation based normative systems are less flexible as it is necessary to specify the rules at design time and the agents are not free to perform actions outside the rules defined by the normative system. The enforcement based approach allows agents to take actions outside the rules of the normative system, but it has the drawback that

sometimes the agents can behave maliciously and not being caught by the enforcer of the law. While proper coordination of agents is crucial, at the same time, it is important to ensure the autonomy of agents. Thus, we envisage electronic institutions as enforcement based normative systems where the agents are free to perform actions outside the rules of the system but for every forbidden action, the system is tracking the violation and applies a corresponding sanction to punish the agent and to preserve its stability. We use enforcement based normative systems that can be created at runtime, but where their rules are first class citizens [21], meaning that the agents can observe the rules.

Moreover, EIs include two basic types of norms: i) constitutive norms and ii) regulative norms. Constitutive norms are based on the notion that "X count-as Y in context C" and are used to support regulative norms by introducing institutional facts in the representation of legal reality [5]. Regulative norms are the main mediation drivers in EIs and are realized by using three main concepts (adapted from [3]) for mediation: power, obligation and permission. Power specifies that an agent can perform a designated action in a context, which creates or changes an institutional fact. Obligation expresses the idea that at a given time the agent should produce an action as specified by the rules of the normative system. Obligation implies also the concept of prohibition or negative obligation as an action that is forbidden by the rules of the system at a certain time. The concept of permission is both related to the state of the EI and to the concept of power. An agent could either exercise its power, if and only if the institutional conditions permit it (conditional power [11]) or exercise its power even if it does not have the permission to do it. On the second case the agent will be sanctioned by the system. Which of the two previous approaches will be followed depends only on the choice of the EI designer.

3 Modeling First-Class Electronic Institutions

3.1 The MANET meta-model

The MANET (Multi-agent Normative EnvironmenTs) meta-model is based on the assumption that the agent environment is composed by two fundamental building blocks; the physical environment, concerned with agent interaction with physical resources and with the MAS infrastructure and the social environment, concerned with the social interactions of the agents and coinciding with the notion of electronic institutions.

In the MANET meta-model we assume that EIs can be composed of three structural components inspired by Stratulat et al. [22]: agents, objects and spaces.

The notion of agent describes the proactive entities within the normative system. For the agents, we assume a separation between a cognitive mind and a physical body with sensors and effectors as described in [6]. The cognitive mind analyzes and reasons about the data received by the sensors as well as reasoning about the agent strategy. The agent uses its effectors to act inside the environment.

The notion of object describes first-class entities that represent virtual entities, virtualizations of external resources or web services, offering an abstraction that hide the low level details from the agents. From the standpoint of EIs, these virtual entities can depict either physical objects either institutional objects.

On one hand physical objects are considered the physical entities of the application (web services, databases, external files etc.) that are present inside an EI. On the other hand, institutional objects, are objects existing only in common agreement amongst the agents of an EI. Institutional objects can be further categorized as: a) objects that can exist within the communication amongst the agents, such as the goods to trade in a market; b) objects that represent agreements between one or more parties; c) objects that represent sanctions for the incorrect behavior of the agent in the EI; d) objects that represent norms of an EI e) objects that represent institutional spaces, f) and finally objects that represent roles of agents within an EI. Moreover, physical objects can be considered as institutional ones when they obtain institutional attributes during the evolution of the agent environment.

Finally the third structural component of our model are spaces. By default in our model, there always exists a root space which contains all the physical laws (derived from the infrastructure of the MAS) of the system (as first-class objects) and where all the other spaces of the system are been created. But in direct analogy to the human reality where we can distinguish between the physical world and the social world, in MASs we can consider institutional spaces [22] describing the EIs in a normative system. All the institutional spaces of a MASs are situated inside the root physical space.

Institutional spaces constitute a first-class representation of the boundaries and the structure of legal entities like EIs. These spaces include the objects and the agents participating in an EI, and contain information about institutions' topology and configuration. The term boundary here implies that spaces specify the limits of the effects of the events performed by the agents. In our model we suppose that the effects of an event produced inside one space hold only for that space. Since spaces are the boundaries and containers of events, they manage norm violations and fulfillments. The content of each event and the combination of role/power of the agents that produced the event are always checked by the space against the corresponding norms. In case of a norm violation, a space will retrieve the information of the appropriate sanction objects and will apply them to the agent that did not comply with the rules of the system. In other words, we see institutional spaces as structures whose state exist in the physical environment, that is perceivable and modifiable through production of events.

It is important to stress that in our model, norms, agreements and sanctions are expressed as complex structures, meaning that they can be deployed as objects in an institutional space. Institutional spaces can be folded inside other spaces or can be distributed across more than one space creating complex topologies. In this paper we show how a space can be created inside another but we do not elaborate the details of possible dependencies between different institutional spaces. We assume that norms of a father space are not propagated to a child space inside it. Each institutional space is discrete and distinct.

In general, in normative systems, agents' interactions can create new institutional realities (e.g. new EIs). In our model, each time a new EI is to be born, a new institutional space is being created, which includes all the norms, the objects and the agents of the institution, which combined together constitute a first-class representation of an EI.

3.2 Modeling First-Class Electronic Institutions with the Object Event Calculus and C-logic

In order to describe the dynamics of our meta-model we use the Object Event Calculus (OEC) formalization. The Object Event Calculus is a dialect of the Event Calculus (EC) [16] that is suitable to represent the evolution in time of complex structures by means of events. The main advantage of OEC is that it determines the state of an object by assigning values to its attribute. Based on this property, it deals with the evolution of an object over time, parameterizing its attributes with times at which these attributes hold various values.

The Object Event Calculus predicates we use for the purposes of this paper are shown below:

(C1) holds_at(Id, Class, Attr, Val, T)← happens(E, Ti), Ti ≤ T, initiates(E, Id, Class, Attr, Val),
not broken(Id, Class, Attr, Val, Ti, T).
(C2) broken(Id, Class, Attr, Val, Ti, Tn)← happens(E, Tj), Ti < Tj ≤Tn, terminates(E, Id, Class, Attr, Val).
(C3) holds_at(Id, Class, Attr, Val, T)← method(Class, Id, Attr, Val, Body), solve_at(Body, T).
(C4) attribute_of(Class, X, Type)← attribute(Class, X, Type).
(C5) attribute_of(Sub, X, Type)← is_a(Sub, Class), attribute_of(Class, X, Type).
(C6) instance_of(Id, Class, T)← happens(E, Ti), Ti ≤ T, assigns(E, Id, Class), not removed(Id, Class, Ti, T).
(C7) removed(Id, Class, Ti, Tn)← happens(E, Tj), Ti < Tj ≤ Tn, destroys(E, Id).
(C8) assigns(E, Id, Class)← is_a(Sub, Class), assigns(E, Id, Sub).
(C9) terminates(E, Id, Class, Attr, _)← attribute_of(Class, Attr, single), initiates(E, Id, Class, Attr, _).
(C10) terminates(E, Id, _, Attr, _)← destroys(E, Id).
(C11) terminates(E, Id, _, Attr, IdVal)← destroys(E, IdVal).

Clauses C1-C2 provide the basic formulation of OEC deriving how the value of an attribute for a complex term holds at a specific time. Clause C3 describes how to represent derived attributes of objects treated as method calls computed by means of a solve_at/2 meta-interpreter as specified in [15]. C4-C5 support a monotonic inheritance of attributes names for a class limited to the subset relation. As C1-C2 describe what holds at a specific time, C6-C7 determine how to derive the instance of a class at a specific time. The effects of an event on a class is given by assignment assertions; the clause C8 states how any new instance of a class becomes a new instance of the super-classes. Finally, deletion of objects is catered for by clauses C9-C11. C9 deletes single valued attributes that have been updated,while C10-C11 delete objects and dangling references.

All the structural entities of our meta-model are considered OEC objects and the relationships between them are depicted in Fig. 1

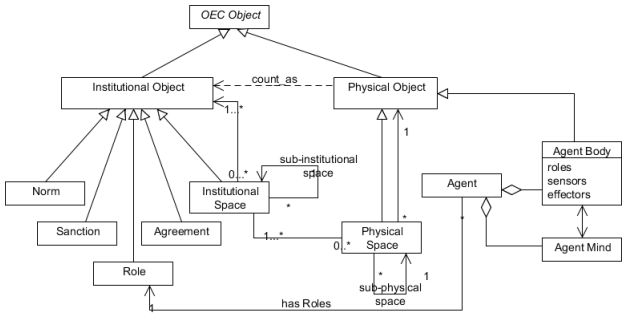


Fig. 1. Entity Categories

To represent the state of the entities at a given time, we will use the C-logic formalism [8] as it is a convenient formalism to represent complex structures and it has a direct translation to the OEC. Complex object descriptions are considered as collections of atomic properties. An object with several attribute labels can be considered as a collection of several atomic formulas. According to the definition of spaces we have introduced, to describe the state of an institutional space at a given time we utilize the following C-logic structure:

```
institutional_space:is1[
    agents ⇒ { agent:a1[ roles ⇒ {role:r1, role:r2} ], agent:a2[ roles ⇒ {role:r1, role:r3} ]},
    institutional_objects ⇒ {norm_object:11, inst_object:o2, inst_object:o3},
    institutional_spaces ⇒ { institutional_space:s2, institutional_space:s3,institutional_space:s4}]
```

that means that is1 is an institutional_space, which has a set of agents a1, a2, a set of institutional objects that the agents can manipulate in the EI and a set of sub institutional spaces. We can translate the C-logic term above to the following first order logic clauses that we can query utilizing the predicates of the OEC:

```
is_a(is1, institutional_space). attribute(institutional_space, agents, multi).
attribute(institutional_space, institutional_objects, multi). attribute(institutional_space, institutional_spaces, multi).
time(e1,1). instance(is1,institutional_space, start(e1)). object(is1,agents, a1,start(e1)).
objects(is1,agents,a2,start(e1)). object(is1,institutional_objects, o1,start(e1)).
objects(is1,institutional_objects,o2,start(e1)). object(is1,institutional_spaces, s3,start(e1)).
objects(is1,institutional_objects,s4,start(e1)).
```

Similarly, the following C-logic structures:

```
power:p1[ mediates ⇒ open_auction:Ev[actor ⇒ IDActor]@T, check_role ⇒ {IDActor, employee}]
sanction:s1[agent ⇒ ag1, credits ⇒ 200]
```

describe respectively a power rule p1, that mediates events of class open_auction, by checking the power of an agent IDActor that enters the EI as an employee to open an auction at time T, and a sanction s1 of 200 credits, applied to agent ag1. We will show later in this paper how such norms and sanctions are applied when the agents execute an action.

3.3 Evolution of Institutional Spaces

To represent our EIs as first-class abstractions, we will need to define how to represent the state of an EI, how to perceive its state and the state of the agents taking part in the interaction, and how to represent the events. In this Section we will illustrate our approach in defining EIs by means of the OEC.

The event schema that we take into consideration in our system is shown in Fig. 2. We distinguish between three kinds of events that are speech events, physical events and sensing events. This distinction is not new and it was already presented in [6], in this paper we further extend the hierarchy of events introducing *institutional events*. Institutional events are considered physical events. This does not go against Searle's definition of social events [20], as, despite the fact that the institutional events modify institutional entitites, they actually change the state of the agent environment acting as regular physical events. Institutional events include the creation/deletion of institutional spaces and objects and are necessary for the construction of every new first-class electronic institution that happens during the evolution of the MAS. Event descriptions are specified as complex terms and are perceivable by any agent inside the institutional space. This property of the events allows the agents to understand every action that happens inside their institutional context. For example, the event description below:

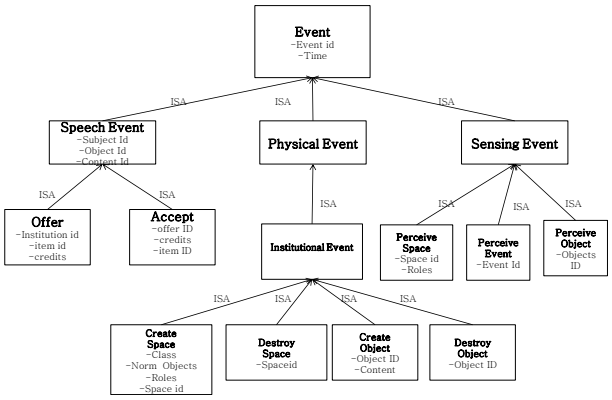


Fig. 2. Events Hierarchy

open_auction:e14[actor ⇒ ag1, auction ⇒ auction:au1[item⇒ medical_item:item1]].

represents an institutional action of agent ag1 who attempts to open an auction about an item item1 of class medical_item. We will see later, how such an action is executed by the agent that causes the event to happen. For the time being, we will assume that the event has happened and we will show how the entities state in the agent environment will evolve as a result of the happening of this event. To do this we need to define domain specific initiates, terminates, assigns and destroys clauses, as shown below:

assigns(E,Obj, auction) ← open_auction:E, auction_of(E,Obj).
initiates(E, Au, auction, item, I) ← open_auction:E [item ⇒ I].

in this way the assigns/3 domain dependent clause above deals with the creation of an auction while the initiates/5 clause assigns an attribute item to the newly created institution. The specification would need also the definition of destroys/3 and terminates/5 clauses to deal with the destruction of an object and termination of an attribute; this is handled in the OEC by the clause C9 shown in Section 3.

3.4 Acting and Perceiving inside Institutions

The representation in terms of C-logic structures of the EIs allows us to have multiple institutions recursively embedded within each others. In order to act within an institutional space, the agents have to be aware of the space where they want to perform an action. For the purposes of this paper, we do not consider dependencies between institutional spaces.

Moreover, the agents' actions are going to be mediated by the regulative rules of the institution as we have already mentioned. As a consequence we say that in order to be performed within an EI, an action has to be attempted in that EI first. We specify how the EI evolves in time by means of assertion of events, where we keep the events description separated from the attempt.

attempt(e14, 120).
do:e14[actor ⇒ ag1, act ⇒ open_auction:m1[institutional_space⇒ IS1]].

In particular, through the rule H1a below we say that in order to happen within the EI the event has to be attempted, the agent producing the event has to have the power to produce the event and the event must be permitted.

H1a)happens(Event, T)← attempt(Event[institutional_space ⇒ IS], T), power(Event,T), permitted(Event, T).
H1b)happens(Event, T)← happens(Event*,T), counts_as(Event*[institutional_space ⇒ IS],Event,T).
H1c)happens(sanction:Event, T)← obligation(Event*@T*,T), T* =T.

As a consequence of defining the happens/2 relation in this way, agents must be aware of the normative systems where they produce events. The rule H1b handles those cases when an event produced outside the normative system, like a physical event in the agent environment, has an effect on a normative system. To achieve this we make use of the counts_as/3 predicate, which states that if an event Event* happens at time T, then also another event Event in relation to an institution identified by IS happens too. The rule H1c specifies that if an obligation has not been satisfied until T, where @T means "at time T", a sanction event happens. We specify further the predicates to enforce the norms of the institution as follows:

H2)obligation(Ev[institutional_space ⇒ IS],T)← instance_of(IS,institutional_space,T), holds_at(IS,norm_object,Oid,T),
instance_of(Oid,obligation,T), apply_norm(Oid,Ev,T).
H3)permission(Ev[institutional_space ⇒ IS],T)← instance_of(IS, space,T), holds_at(IS,institutional_space,norm_object,Oid,T),
instance_of(Oid,permission,T), apply_norm(Oid,Ev,T).
H4)power(Ev[institutional_space ⇒ IS],T)← instance_of(Sid, institutional_space,T),
holds_at(Sid,institutional_space, norm_object, Oid,T), instance_of(Oid,power,T), apply_norm(Oid,Ev,T).

The clauses H2), H3), H4) specify the concepts of power, permission and obligation, that define three distinct kind of norms. The predicate apply_norm/3 is a meta-interpreter that takes the norms in form of objects and check them against the events produced. To express how perception takes place in the EIs, we define the H5) and H6) clauses.

H5)notify(Class:E, Sensor, T) ← happens(E, T), E[institutional_space ⇒ IS], holds_at(IS,agent,Ag,T),
holds_at(IS,owns,Sensor,T), holds_at(Sensor,senses,Class,T).
H6)perceiveE, S, T) ← happens(E, T), perceive_institutional_space(E), E[sensor_of⇒ S, focus⇒ Focus,
institutional_space ⇒ IS].

H5) specifies that whenever an event happens within an institutional space, such event is notified to the agents that are part of such space if they have a sensor that is capable to perceive such events. H6) specifies how an agent can focus to a particular institutional space and perceive its properties, where the solve_at/3 predicate returns a variable substitution of the variables in Focus, if any. The implications of rule H6) is that the agents deployed in the agent environment and taking place in an institutional space can perceive the institutional entities, such as agreements, sanctions and norms, in the institutional space.

4 Applying MANET to an e-Health Marketplace

In this section we illustrate our model with an application from the e-Health domain. We consider an e-Health Marketplace where different hospitals can trade about medicaments, medical equipment and blood. This marketplace can work as a secondary market to the contracts between hospitals and medical providers. The whole marketplace is based on multiple Dutch auctions which can run simultaneously.

To model the e-Health Market place we make use of the general purpose rules presented in Section 3 and we add a set of domain dependent axioms to deal with the evolution of an EI representing a Dutch auction in order to sell medicaments in a e-Health marketplace. In particular, we adapted the Dutch auction as presented in [11] to our formalism based on the OEC and we introduce norms expressed in terms of objects

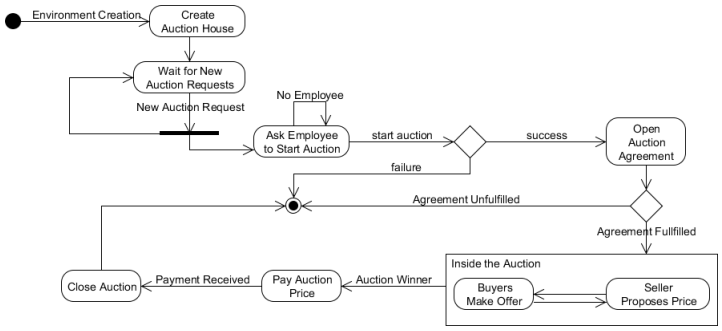


Fig. 3. Auction House Environment State Chart

of the Object Event Calculus formalism. Fig. 3 shows the life-cycle of a Dutch auction within the auction house agent environment.

The Dutch auction electronic institution defines a set of roles for the performance of institutional actions. The roles define the powers that the agents have in the institution. These roles, for the purposes of the e-Health marketplace example, are: a) *Employee*: it is an agent representing the auction house agent environment and that is entitled to open auctions. The agent having this role can also run an auction assuming the role of auctioneer for that auction; b) *Participant*: it is an agent that can express interest for an auction, becoming buyers within the auction; c) *Buyers*: it is an agent that is participating to an auction in the auction house agent environment trying to buy an item of interest; d) *Auctioneer*: it is an agent that coordinates an auction on behalf of a seller agent; e) *Seller*: it is an agent that delegates an auctioneer to sell an item in the Dutch auction. In Fig. 4, once the auction house is created, the environment waits for the opening of an auction. Once a seller contacts an employee agent to open an auction, the employee agent creates an agreement institutional object in the agent environment. This agreement is perceivable by all the agents inside the auction house, which can modify its attributes only with institutional actions. In C-logic terms this agreement object is described as follows:

agreement:c1[
object ⇒ medic_item:b1, debtor:a1[roles ⇒ {role:employee, role:seller}],
creditor:a3[roles ⇒ {role:seller, role:auctioneer}], minimum_price ⇒ 200, deadline ⇒ 2000,
participants ⇒ {agent:aid1, agent:aid2 ... agent:aidn}]

The term above specifies that an agent a1 is going to open an auction before time 2000 for a medical item b1. When an agreement is created, it can be observed from the agents populating the environment, that can express their interest to participate in the auction by modifying the participants attribute of the agreement object with an institutional action. In particular, when the deadline of the agreement expires, we utilize a count_as/3 as follows:

count_as(participate:Ev, assign_role:Ev*, T)← actor_of(Ev,AID),instance_of(C,agreement,T),
instance_of(C,institution,IS,T),holds_at(C,participant, AID,T), role_of(Ev*,buyer), institution_of(Ev*, IS).

to specify that the participate event Ev counts as an assign role event Ev* in the newly created EI for the auction.

The powers of the agents are constrained by the permission norms in the EI: for example an auctioneer is authorized to open an auction only if its starting time has

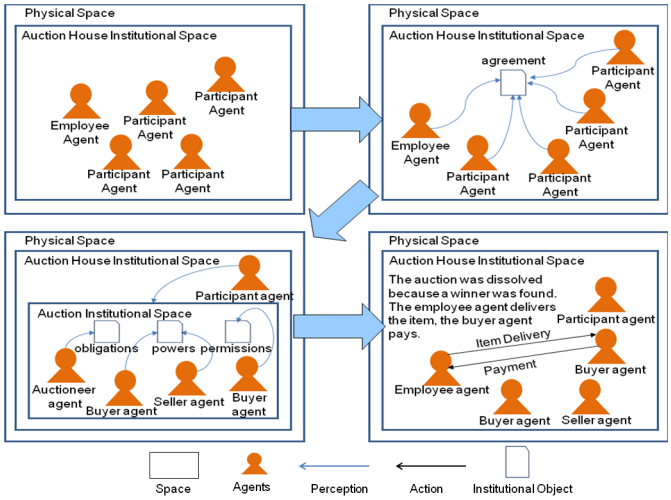


Fig. 4. Interaction in the Auction House Agent Environment

elapsed and if there are at least two agents registered as participants. Fig. 4 represents the interaction taking place in the agent environment represented by the auction house where the auctions are created and dissolved. In particular, as defined in the auction life-cycle in Fig. 3, the e-Health auction is dissolved when an agent wins the auction offering a price that matches the current offer of the seller. To handle the evolution of the auction within the agent environment represented by the auction house, we utilize the following norms:

N1)power:n1[mediates ⇒ start_action:Ev[auctioneer⇒ agent:A, item ⇒ O, starting_price ⇒ P, institutional_space ⇒ IS]@T, check_role ⇒ {A, employee, T}]
N2)power:n2[mediates⇒change_price:Ev[auctioneer⇒agent:A, item ⇒ O, new_price ⇒ Price,institutional_space ⇒ IS]@T, check_role ⇒ {IS, A,auctioneer, T}]
N3)permission:n3[mediates⇒ change_price:Ev[auctioneer⇒agent:A,item ⇒O,new_price ⇒ Price]@T,hasItem⇒{IS,O T} check_role ⇒ {IS,A,auctioneer, T}, hasPrice ⇒ {IS, O, CurrentPrice, T}, lessThan ⇒ {Price, CurrentPrice}]
N4)obligation:n4[mediates ⇒ assign_item:Ev[auctioneer ⇒ Auc, item ⇒ O, buyer ⇒ Buyer, institutional_space ⇒ IS]@T, lastOffer ⇒ {IS,Buyer,O, LastOffer}, currentPrice ⇒ {IS, O,CurrentPrice}, equal ⇒ { LastOffer,CurrentPrice }]
N5)obligation:n5[mediates ⇒ pay:Ev[buyer ⇒ Buyer, amount ⇒ LastOffer, item ⇒ O, institutional_space ⇒ IS]@T, isAssigned ⇒ { IS, Buyer,O,T }, currentPrice ⇒ { IS,O,Price,T }, equal ⇒ LastOffer,Price]

Norm N1 specifies that an agent has the power to start an auction in the auction house space when it is an employee for the auction house, while norm N2 and norm N3 express the power of an agent to change the price of an item within an auction space in which the agent is taking part with the role of auctioneer, and the permission to change the price from the point of view of the auction if the auction has that item and the new price is less than the previous one. Norm N4 expresses the obligation of the auctioneer to assign an item to the winner of the auction, while norm N5 expresses the obligation of a buyer agent to pay for the item assigned by the auctioneer. In the case that the events produced by the agents respect the norms of the institutional space, then the evolution of the Dutch auction institutional space is handled in terms of initiates/4 and terminates/4 clauses that modify the attributes of the Dutch auction whenever an event takes place. For example, the following clauses:

initiates(change_price:Ev, AuID, lastoffer, NewOff)← time(Ev,T), offer(Ev, NewOff), auction(Ev,AuID), value(NewOff, NVal), holds_at(AuID, lastoffer, OldOff, T), value(OldOff, OVal), NVal < OVal.

terminates(makeoffer:Ev, AuID, lastoffer, _)← initiates(makeoffer:Ev, AuID, lastoffer, NewOff).

state that a new offer, is considered the last offer, only if the value of the offer is less than the previous offer. Finally, we introduce a domain dependent count_as/3 to deal with the case of a buyer agent leaving the institutional space of an auction before having paid:

```
count_as(leave_auction:Ev[institutional_space ⇒ IS],sanction:Ev*, T)← actor_of(Ev,AID),
        obligation(pay:Ev**[institutional_space ⇒ IS],T), actor_of(Ev**,AID), institution_of(Ev*, IS), credit_of(Ev*,200).
```

The count_as/3 above specifies that an agent leaving while an obligation of paying holds in the EI, will be sanctioned of 200 credits. A further count_as/3 clause has been defined to handle the exception of an auctioneer not delivering the good after the auction, but we omit it as it is similar to the clause above.

5 Implementation Issues

For the implementation of the normative systems we adopted a logic programming approach due to the formal and declarative semantics of our model and we implemented it as a Prolog theory that we include in the GOLEM framework [6]. In particular, we utilized a version of the OEC described in [15], which is based on caching the periods of time in which an attribute of an object holds. The top-level implementation for the holds_at/4 in OEC is shown below:

```
holds_at(Obj,Attr,Val,T):- object(Obj,Attr,Val,start(E)), time(E,T1), T1 =< T,
    not (object(Obj,Attr,Val,end(E*)), time(E*,T2), T2>T1, T2<T).
```

where the object/4 assertions store when an attribute have been initiated/terminated at a certain time. A similar definition for the instance_of/3 predicate exists, using instance/3 assertions. This representation brings the advantage that indexing can be performed on both the time interval and the object identifier, meaning that the time to retrieve the attribute of an object is O(1), once the identifier and the interval are known as in our specification. An example of the state of a norm of the normative system to create an auction in the auction house agent environment can be expressed as follows:

```
instance(r1:[IDS,ID,auctioneer, T], power, start(e1)). object(r1:[IDS,ID,auctioneer, T], mediates, open_auction, start(e1)).
object(r1:[IDS,ID,auctioneer, T],template, do(ID,open_auction, [property(institution, IDS)]):T,start(e1)).
object(r1:[IDS,ID,auctioneer, T],check_role,[IDS,ID, auctioneer,T],start(e1)). time(e1,1).
```

The state above specifies that there is an instance of a norm that mediates an event of class open_auction, where the template attribute defines the mediated event. The norm also specifies that it calls the check_role/4 predicate to check if the role of the agent performing the event is the one of auctioneer. Notice that we append the variables that will be called in the check_role/4 predicate in the identifier of the rule, so that we can instantiate their value in the apply_norm/2 meta-predicate as specified below:

```
attempt(E,T):- power(E,T), permitted(E,T), add(E,T).
power(E,T):- E = do(Actor,EventClass, Elements), member(property(institution, IDS), Elements),
        instance_of(IDS, institution,T), holds_at(IDS, rules, ID:Vars, T), instance_of(ID:Vars,power,T),
        holds_at(ID:Vars,template, E:T,T), not(not(apply_norm(ID:Vars,T))).
apply_norm(ID:Vars,T):- holds_at(ID:Vars, Attr, Vars, T), append([Attr],Vars, Var2), Pred =.. Var2, Pred.
```

The attempt/2 predicate implemented above checks if the agent has the power and the permission to produce an event in the environment. If this is the case the add/2 predicates add object/4 or instance/3 assertions to the Prolog database, according to the effects of the event. The power/2 predicate, checks if there is a norm that specifies if the agent has the power to produce a certain event with respect to a certain institution.

To do so, the power/2 predicate utilizes the apply_norm/2 meta-predicate to check if the norm specifies any constraint that prevents the agent from performing the action. For example, we implement the check_role/4 constraint as follows:

```
check_role(IDS, ID,Role,T):- instance_of(ID,agent,T), instance_of(IDS,institution,T),
                             holds_at(IDS, roles,RID,T), instance_of(RID,Role,T), holds_at(RID, agent, ID,T).
```

The check_role/4 predicate implemented above checks if an agent identified with the variable ID, has a certain role Role in an EI IDS at a certain time T.

6 Related work

Fornara et al. have developed OCeAN [12], a meta-model for the specification of artificial institutions, and an Agent Communication Language (ACL) to model open interaction systems where heterogeneous software and human agents interact. The OCeAN meta-model consists of the following components: (i) constructs to define the core ontology of an institution, (ii) roles and of events; (iii) the counts-as relation, which is necessary for the concrete performance of institutional actions; (iv) and norms. One difference between MANET and the OCeAN meta-model is that we consider institutions as first-class entities, which allow the perception of their components (e.g. norms, objects and sanctions) that are also described as first-class entities. Another difference is related to the types of events that are possible inside an institution. In the OCeAN meta-model only communication events are considered, whereas in our approach we define a more detailed schema of events in order to describe all the possible situations during the evolution of an open MAS.

Artikis and Sergot in [3] present a model of executable specifications of open MAS where open MAS are considered instances of normative systems. The authors represent the social constraints (laws) of the system in terms of physical capabilities, institutional power, permission and prohibition as well as sanctions and enforcement policies. In our model we adopt a very similar model of institutional rules based on powers which are dependent on permissions, obligations and sanctions. However, in our work we consider institutional rules as first-class entities which can be observed by the agents, allowing them to reason about the normative constraints of the open MAS.

In [23] Urovi and Stathis define the MAGE framework. MAGE uses the OEC formalism to represent games as first-class entities that evolve in time. Such games are interconnected between each others in a hierarchy composed of atomic games and composite games. The state of the composite games is defined by the relationships between the atomic games and their transitions and the agents can perceive the legal actions in a game at a certain time. MANET institutional spaces correspond to MAGE games which also evolve due to the production of events. The main difference between MANET and MAGE is that we include the possibility of defining institutional objects, such as norms and agreements, allowing for norms to have a structure and be perceivable, meaning that the agents can reason about whether or not complying with a norm is to their best interest.

Also related to our work is the work of Piunti et al. to unify at one model the concepts of agents, organizations and environments [19]. This model allows for designing

and programming an environment in terms of a dynamic set of first-class computational entities called artifacts, collected in workspaces. Artifacts represent resources and tools that agents can dynamically instantiate, share and use to support their individual and collective activities. The notions of artifacts and workspaces have similarities with these of objects and spaces. But unlike Piunti et al. spaces in our approach are not just the containers of agents and objects modeling the locality of the application domain but contrariwise they are the main enforcers of the law and regulators of the MAS evolution.

7 Conclusion and Future Works

We presented a meta-model that describes institutions as social agent environments [24] by extending upon the OEC formalism [15], based on the concept of agent environment as presented in [6]. In particular we presented a model that can handle the life-cycle of multiple institutions at runtime, where the institutions are represented as first-class objects that the agents can perceive. Moreover, our model supports the definition of institutional objects, such as agreements, sanctions and norms that the agents can perceive as part of an electronic institution. We presented this model utilizing an e-Health marketplace based on Dutch auctions as a motivating example.

There are several directions that is worth exploring for future works. First of all, we plan to extend our framework to handle dynamic norm change within the institution when an agent society requires it due to an external exception. Secondly, we plan to extend our framework to manage institutions in distributed settings. As recognized by Artikis et al. in [2], learning the rule of an institution is recognized as a problem, as a consequence we plan to investigate machine learning approaches that can make use of our model to create cognitive agents capable to learn how to interact within multiple heterogeneous institutions.

ACKNOWLEDGEMENTS

Our work is partially supported by the SER project Open Interaction Frameworks: Towards a Governing Environment under the Agreement Technologies COST Action IC0801.

References

1. T. Ägotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.
2. A. Artikis, G. Paliouras, F. Portet, and A. Skarlatidis. Logic-based representation, reasoning and machine learning for event recognition. In *DEBS*, pages 282–293, 2010.
3. A. Artikis and M. Sergot. Executable Specification of Open Multi-Agent Systems. *Logic Journal of the IGPL*, 18(1):31–65, 2010.
4. K. S. Barber and J. Kim. Soft security: Isolating unreliable agents from society. In *Trust, Reputation, and Security*, pages 224–233, 2002.

5. G. Boella, G. Pigozzi, and L. van der Torre. Normative systems in computer science - ten guidelines for normative multiagent systems. In G. Boella, P. Noriega, G. Pigozzi, and H. Verhagen, editors, *Normative Multi-Agent Systems*, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009.
6. S. Bromuri and K. Stathis. Distributed Agent Environments in the Ambient Event Calculus. In *DEBS '09: Proceedings of the third international conference on Distributed event-based systems*, New York, NY, USA, 2009. ACM.
7. J. Campos, M. Lpez-Snchez, J. Rodrguez-Aguilar, and M. Esteva. Formalising situatedness and adaptation in electronic institutions. In J. Hbner, E. Matson, O. Boissier, and V. Dignum, editors, *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428 of *Lecture Notes in Computer Science*, pages 126–139. Springer, 2009.
8. W. Chen and D. S. Warren. C-logic of Complex Objects. In *PODS '89: Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 369–378, New York, NY, USA, 1989. ACM Press.
9. M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specifications of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective.*, pages 126–147, London, UK, 2001. Springer-Verlag.
10. M. Esteva, B. Rosell, J. A. Rodríguez-Aguilar, and J. L. Arcos. Ameli: An agent-based middleware for electronic institutions. volume I, pages 236–243. ACM, ACM, 2004.
11. N. Fornara and M. Colombetti. Specifying artificial institutions in the event calculus. In *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, pages 335–366, 2009.
12. N. Fornara, F. Viganò, M. Verdicchio, and M. Colombetti. Artificial institutions: a model of institutional reality for open multiagent systems. *Artif. Intell. Law*, 16(1):89–105, 2008.
13. T. Gruber. Collective knowledge systems: Where the Social Web meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4–13, 2007.
14. J. Hendler and T. Berners-Lee. From the semantic web to social machines: A research challenge for ai on the world wide web. *Artif. Intell.*, 174(2):156–161, 2010.
15. F. N. Kesim and M. Sergot. A Logic Programming Framework for Modeling Temporal Objects. *IEEE Transactions on Knowledge and Data Engineering*, 8(5):724–741, 1996.
16. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95, 1986.
17. S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 153–160.
18. J. Pitt, A. Mamdani, and P. Charlton. The open agent society and its enemies: a position statement and research programme. *Telematics and Informatics*, 18(1):67–87, 2001.
19. M. Piunti, O. Boissier, J. F. Hubner, and A. Ricci. Embodied organizations: a unifying perspective in programming agents, organizations and environments. In *11th Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems*, 2010.
20. J. R. Searle. *The construction of social reality.* Free Press, New York, 1995.
21. C. Strachey. Fundamental concepts in programming languages. *Higher Order Symbol. Comput.*, 13(1-2):11–49, 2000.
22. T. Stratulat, J. Ferber, and J. Tranier. Masq: towards an integral approach to interaction. In *AAMAS (2)*, pages 813–820, 2009.
23. V. Urovi and K. Stathis. Playing with agent coordination patterns in MAGE. In *Coordination, Organization, Institutions and Norms in Agent Systems (COIN@AAMAS09)*, Budapest, Hungary, May 2009.
24. D. Weyns, A. Omicini, and J. Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007.
25. M. Wooldridge. *MultiAgent Systems*. John Wiley and Sons, 2002.