COIN@AAMAS2015

The XIX International Workshop on Coordination, Organizations, Institutions and Norms in Multiagent Systems

PRE-PROCEEDINGS

edited by
Pablo Noriega and Murat Sensoy

COIN@AAMAS2015

Coordination, Organizations, Institutions and Norms in Multiagent Systems

***

Istanbul. May 4th, 2015

***

Pablo Noriega & Murat Sensoy (editors)

Preface

COIN, the International Workshops in Coordination, Organizations Institutions and Norms in Multiagent Systems, is about open multiagent systems and how one can bring governance to them. In COIN the focus is on social rather than individualistic aspects of agency. In particular, COIN workshops are concerned with multiagent systems that show the following features: (i) Agents engage in collective activities where other agents need to be involved in order to achieve individual or group goals. (ii) The systems are open in the sense that agents may enter and leave the system at will, and neither the system nor participating agents may know which agents will be active at some point. (iii) Agents are heterogenous and respond to different principals. (iv) Agents are autonomous, they act on their own interest and may be malevolent or incompetent. (v) Agents are "opaque" to the system because the system has neither control of their decision-making nor access to their internal state. (vi) In some cases, the systems are hybrid: containing humans as well as software agents. The main challenge, in this context, is how to endow the multiagent systems with mechanisms that facilitate the pursuit of those collective activities. The COIN workshops series is a meeting point for a community that recognises the role played by coordination, organizations, institutions and norms in facing that challenge of regulating open MAS.

COIN@AAMAS2015 is the nineteenth edition of the series and the fourteen papers included in these proceedings demonstrate the vitality of the community and will provide the grounds for a solid workshop program and what we expect will be a most enjoyable and enriching debate.

We would like to thank the program committee for the fantastic effort they put in the reviewing process, and the authors for submitting their papers.

May 4, 2015
Istanbul

Pablo Noriega
Murat Sensoy

Program Committee

Mohsen Afsharchi	University of Zanjan
Huib Aldewereld	Delft University of Technology
Estefania Argente	Universidad Politecnica de Valencia
Alexander Artikis	NCSR "Demokritos"
Guido Boella	University of Torino
Olivier Boissier	ENS Mines Saint-Etienne
Dídac Busquets	Imperial College London
Patrice Caire	university of Luxembourg, computer science dpt.
Cristiano Castelfranchi	Institute of Cognitive Sciences and Technologies
Amit Chopra	Lancaster University
Rob Christiaanse	EFCO BV
Luciano Coutinho	Universidade Federal do Maranhão (UFMA)
Stephen Cranefield	University of Otago
Natalia Criado	University of Bolton
Mehdi Dastani	Utrecht University
Geeth De Mel	IBM TJ Watson Research Center
Marina De Vos	University of Bath
Frank Dignum	Utrecht University
Virginia Dignum	TU Delft
Nicoletta Fornara	Universita della Svizzera Italiana, Lugano
Amineh Ghorbani	Delft University of Technology
Aditya Ghose	University of Wollongong
Davide Grossi	Department of Computer Science, University of Liverpool
Ozgur Kafali	Royal Holloway, University of London
Anup Kalia	North Carolina State University
Martin Kollingbaum	University of Aberdeen
Christian Lemaitre	Universidad Autonoma Metropolitana, UAM
Victor Lesser	University of Massachusetts Amherst
Henrique Lopes Cardoso	FEUP/LIACC
Maite Lopez-Sanchez	University of Barcelona
Emiliano Lorini	IRIT
Samhar Mahmoud	King's College London
Eric Matson	Purdue University
Felipe Meneguzzi	Pontifical Catholic University of Rio Grande do Sul
John-Jules Meyer	Utrecht University
Simon Miles	King's College London
Daniel Moldt	University of Hamburg
Pablo Noriega	Artificial Intelligence Research Institute (IIIA-CSIC)
Eugénio Oliveira	Faculdade de Engenharia Universidade do Porto - LIACC
Andrea Omicini	Alma Mater Studiorum–Università di Bologna
Nir Oren	University of Aberdeen
Sascha Ossowski	University Rey Juan Carlos
Julian Padget	University of Bath
Simon Parsons	University of Liverpool
Jeremy Pitt	Imperial College London
Alessandro Ricci	University of Bologna

Antonio Carlos Rocha Costa	Universidade Federal do Rio Grande - FURG
Juan Antonio Rodriguez Aguilar	IIIA-CSIC
Antonino Rotolo	CIRSFID, University of Bologna
Bastin Tony Roy Savarimuthu	University of Otago
Murat Sensoy	Ozyegin University
Christophe Sibertin-Blanc	University of Toulouse / IRIT
Jaime Sichman	University of Sao Paulo
Liz Sonenberg	Melbourne University, Department of Information Systems
Charalampos Tampitsikas	Università della Svizzera italiana
Pankaj Telang	North Carolina State University
John Thangarajah	RMIT University
Luca Tummolini	ISTC-CNR
Leon van der Torre	University of Luxembourg
M. Birna van Riemsdijk	TU Delft
Harko Verhagen	Dept. of Computer and Systems Sciences, Stockholm University/KTH
Dani Villatoro	IIIA-CSIC
George Vouros	University of Piraeus

Table of Contents

A Cognitive Framing for Norm Change

Cristiano Castelfranchi

ISTC-CNR GOAL Lab Italy
cristiano.castelfranchi@istc.cnr.it

Abstract. Norms are within minds and out of minds; they work thanks to their mental implementations but also thanks to their externalized supports, processing, diffusion, and behavioral messages. This is the normal and normative working of Ns. Ns is not simply a behavioral and collective fact, 'normality' or an institution; but they necessarily are mental artifacts. Ns *change* follows the same circuit. In principle there are two (interconnected) *loci* of change with their forces: mental transformations vs. external, interactive ones. Ns change is a circular process based on a loop between 'emergence' and 'immergence'; that is, changes in behaviors presuppose some change in minds, while behaviors causal efficacy is due to their aggregated macro-result: acts that organize in stable choreographies and regularities build (new) Ns in the minds of the actors. More precisely the problem is: which are the crucial mental representations supporting an N conform (or deviating) behavior? And *which kinds of 'mutations' in those mental representations produce a change in behavior?* I will focus my analysis on Social Norms, in a broad sense.

Keywords: Norm change; Normative mind; Normative Agents.

1. Premise: Situated normative cognition

I will discuss the internalized/externalized nature and working of Norms (Ns) and its impact on N change. What I have in mind is a *hybrid society* (humans and AI-Agents interacting together) with "norm sensible Agents". On the one side the Agent mediating and supporting human interaction, exchange, organization should be able to understand human conduct in terms of Ns and to monitor and support that; on the other side Agents should be themselves regulated by true Ns (not just pre-implemented binds, executive procedures, but real deontic representations with the mission to regulate their decisions and conducts) and able to violated them in the right situation.

The analysis and typology that I will propose (that will not be complete and fully systematized, but just *in fieri*) is focused on Social Norms (SocNs), in a broad sense, covering various kinds of.[1] Of course here I will put aside legal Ns (where there are institutional and legal ways for N change) although I think that several of the mechanism that I try to enlighten for SocNs also hold for legal ones.

Norms are in minds and out minds; they work thanks to their mental implementation but also thanks to their externalized supports, processing, circulation, and dynamics. This is the normal and normative working of Ns. Also because usually a N is a strange relation between a practical, effective, externalized object (the conduct of X; however mentally/internally regulated) and a cognitive artifact: a written "table of law", a symbolic representation, a (verbal or non-verbal) message that has to pass into minds. This double face of N (cognitive and behavioral, both internal and external) is intrinsic. Ns are not simply a behavioral and collective fact, a "normality" or an institution; but they necessarily are mental artifacts [22], [13]. A N impinges on us and works *thanks to* its mental representation, (partial) understanding, and specific motivations. However, as we just said,

[1] From politeness to customs, from moral norms to Ns and rules in organizations, associations, communities of practice with their "rules". For a systematic analysis of social norms and discussion about the general theory see [6], [5], [35], [12], [31].

they are not just a mental fact: this serves to determine and control the actors' conducts and to build shared practices, scripts, messages and collective effects.
Our claim is that also Ns *change* follows the same circuit. In principle *there are two (interconnected) "loci" of change with their forces: mental transformations vs. external, interactive ones*. Of course, they are interrelated since the mental changes determine behavioral changes, which determine collective new dynamics. Vice versa, behavioral changes that we observe will change our mind and our norm conception or repertoire. In other terms it is both a process of 'emergence' [42] and 'self-organization' and a process of 'immergence' [14], [21] and mentalization: a feedback from behavior and collective structure/phenomenon back to the individual minds layer. Not just a bottom-up and top-down, and an inside-outside and outside-inside process, but a real 'loop': virtuous or vicious *circles* of Ns change or confirmation or instauration. We need the same dynamics in normative Agents, able to learn and evolve SocNs, and to read the behaviors of others in these terms for monitoring it or adjusting to it.
It would also be relevant to consider that there is no just one and unique normative role for actors with its specific mental attitudes (beliefs, goals, expectation, ..). We are not only 'subjects' to the N (prescribing us certain behaviors and mental states), we also have to play the role of 'watchman' and 'punishers' of the others [11], [30]; a fundamental role in N script and for the maintenance of the social order. We have to play the role of 'issuers' too: (either explicitly or implicitly) proclaiming Ns, prescriptively informing about them, explaining and reminding us them (for example parents towards children). I will put aside here these different normative minds and roles[2], although I believe that the role of a normative 'watchman' will be very relevant for Agents.
What we will try to do in this work is to examine: (a) some of the *main mutation 'events'* in particular *internal* to the *subject's* normative minds; but also (b) as individual conducts become signs (cues) and/or messages (signaling), and change the others and the collective emergent conducts, so becoming public phenomena and institutions. Also the other way around; I will give some hints about that: (c) how acts that organize in stable collective conducts build Ns in the minds of the actors [6] but not just as regularity to conform to, but us expectations and "prescriptions" from the others [23], [19].

2. Roots of Ns into minds

Ns as "norms" are based on the possibility to be violated, not obeyed. Otherwise they are not "norms" but physical barriers or ties and chains. They are devices for the control of "autonomous" agents that decide what to do on the basis of their beliefs, reasoning, and goals. Ns not only presuppose (accept) but also postulate a freedom in the addressees.
Our main claims are the following ones:

> A N is not just aimed at regulating our conduct, at inducing us to do or not to do a given action; it is aimed at inducing us to do that action *for specific motives*, with a given mental attitude (belief, goal, expectation). The ideal-typical *Adhesion* (see 3.2) to a N is for an intrinsic motivation, for a "sense of duty", recognition of the authority, because it is right/correct to respect Ns, etc.; and only sub-ideally one should respect for avoiding external or internal sanctions (see below). Also normative education goes in this direction [18].

> We agree with Bicchieri's theory that an "empirical expectation" and the perception of the existence of a "normal" diffused behavior is not enough for creating a real N in

[2] I will also do not examine the other crucial phenomenon in Ns evolution: the introduction of a completely new N, and its issuing or negotiation. I will mainly focus on adherence or violation (and their reasons) in N changing, adaptation, or extinction.

"normative" sense (to use Kahaneman' terminology [37]). A merely "descriptive" N is not "injunctive" [40]; a N implies for us a prescriptive character: it is for inducing us to (not) do something. There is a social pressure: expectation and *prescription*.

> As we said, our object is "norms" in the "normative" (prescriptive) meaning/sense, not in the "normality" (descriptive or statistic or standard sense). However there is an important and bidirectional goal-relation between N in *normative* sense and N in *normality* sense:

a) Normality-N creates and becomes a Goal for the actors and even a normative-N (a prescription, something "due"), in order to conform, to be like the others. This conformity is either a need of the individual or a need (and request/pressure) of the group, or both.

b) Normative-N creates a statistical normality-N, a normal conduct in the community, if it is respected: N conformity is "normal". Moreover:
Normative-N has the *goal* and the *function* to be respected and thus to create a normality-N, a normal behavior (at the individual, internal level this helps it also o become an automatic response, just an habit);
If normative-N doesn't become/create a normality-N it is weakened and perceived as less credible and less binding [6], [22].

> In order to *perceive* a social practice as a N we have to guess, presume, or understand some "end" in it: the protection of the interest or rights of somebody, of the community; from that a deontic "should", an obligation. Not conforming is an harm, is noxious, not just something irregular, strange. I'm at least frustrating your prescription to maintain regular practices; you count on that and plan to regulate your behavior on that; so I'm upsetting and betraying you, not just amazing you. I'm harming social order, and the natural 'suspension' of uncertainty, the assumption of normality: a fundamental good [32].

> Ns have to be "impersonal" and depersonalized (and perceived as such) on both sides: the issuer's and the addressee's side. It is not a conflict between you and me; it is not "my" personal request (for me, for my desires, etc. for my personal will that you have to adopt); and it is not a request to "you". The message is:
"I do not talk, monitor, sanction, in my name"; "I'm not addressing to you "ad personam", but as an instance of a class, a member, a citizen, ... like any other in the same conditions". Also for that *"You have no reasons for rebelling"*.
This really is a crucial point in the perception of Ns as Ns; thus it is something that must be signaled in some way (for official Ns: uniform, role symbols, specific documents, etc; for Social Ns by collective practice or attitude or explicit messages)) or at least contextually presupposed and assumed in the script. [3]

> As we said, Ns are social devices controlling behaviors through minds [14] but in a specific way; through a partial understanding. They require (for their existence and effectiveness) their *explicit mental representation*, their (partial) understanding and recognition "as Norms"; specific cognitive representations and motivational processes (*"Cognitive Mediators"*: [22], [24]); differently from other social phenomena like *social functions,* that can be played by social actors even without understanding - and even less intending - them [16]. Not necessarily the agent supporting the N in some role has as his/her mental goal ("intends") G1 and G2; these are the goals (and functions) of the N not of the individuals.

[3] The fact that Ns are always relative to a "class" of subjects, not just to one specific person and it holds "for all the values of X" is one reason why the violation has not an individual meaning. X the violator is just "one of all/many", is a representative, an "example"; that's why his (bad) behavior can be a (bad) "example"; and the impact of the behavior is more that "individual": It is not longer true that "for *any* value of X, X has to, will do, and does action A".

> Ns have to build in us an "ought", a "duty", "you have to"; with a rather constrictive feeling, a negative "frame", an avoidance orientation (even when it elicits "you have *to do* this action"). And this "ought" is a non-technical "ought", not instrumental to and planned for a given outcome/goal. This entails a process of Adhering *without sharing* the 'instrumental' nature of the N, and without (necessarily) understanding /adopting its 'function' or end. My 'plan' is different from the authority's 'plan'. Citizens are not real "cooperators" but "subjects". They have to "alienate" their own powers and products [18].

3. N internalization

Anyway, all this requires a specific "translation" of Ns into the minds of the addressees such that they recognize a N as such, and – on the basis of various motives – decide whether to conform or not to it. Let's sketch the basic constituents of Ns internalization in our theory [24], [18]. Ns are based on a *specific* process of *Goal-Adoption* or better *Adhesion*; since they have the nature of an "imperative".

3.1 Goal Adoption and Adhesion

Ns induce new goals through "adoption". *Goal-Adoption* is how an autonomous agent is not an isle but becomes social, or better pro-social [4]; that its s/he does something *for* the others; *puts her/his autonomous goal-pursuing (intentional action), her/his cognitive machinery for that, and her/his powers and resources, into the service of the others and of their interests.* What is needed is the architecture of a social Agent able to import goals from outside (and to influence other agents by giving them goals and relying on him/her) but remaining 'autonomous'. S/he is able to arrive to set up an intentions not only from her own endogenous 'desires', but also from imported goals.

Goal-Adoption means that:

X believes that Y has the goal that p and comes to have (and possibly pursue) the Goal that p just because he believes this.

"I do something 'for' you" (which doesn't mean 'benevolence'!); I want to realize this since and until you wants/ needs this; because it is your goal.

Of course there are different kinds of Goal-adoption, *motivated* by different reasons: merely selfish and instrumental, like in exchange; altruistic; or strictly cooperative, for a common goal. Ns prescribe a specific motive for accepting the injunction: in Bicchieri view's a "normative expectation", for us also the recognition of the expectation/prescription by the others and their the authority (see below).

A stronger form of G-Adoption is ***Adhesion***: *when I adhere to your (implicit or explicit) 'request'* (of any kind: prey, favor, order, law, etc.). In other words, *you (Y) have the goal that I adopt your goal p, that I do something (action a of X) realizing that goal, and I adopt your goal p or of doing a, (also) because I know that you expects and wants so.*

In Adhesion one of the *reasons* for Adopting the goal of the other is that the other wants so:

-She also has the (meta-)goal that we adopt her goal;

-We adopt her goal by adopting the meta-goal.

In a sense, there is a double level of adoption (a meta-adoption): *I know and adopt your goal that I adopt.* Moreover, in case of Adhesion there is a (presupposed) agreement between X and Y about X's adoption, X doing something as desired by Y. Other forms of adoption (like help) can be unilateral, spontaneous, and even against Y's desire. Ns require from us not just adoption but adhesion.

[4] Not to be used as synonym of "altruistic", "benevolence", etc.

3.2 Normative Adhesion

Adhesion obviously presupposes specific beliefs into the mind of the agents (and this is the first aim of the N: to be conceived/perceived as such). In particular the recognition of the N as a N, in force on me, and valid in that context.
It is implied a 'generalized' G-Adoption where:

-X believes that there is a goal impinging not directly on a single individual but on a class or group of agents:
-if X believes to belong to that class,
-she believes to be concerned by the norm, and
-she instantiates a Goal impinging on her; adopts it.

Having adopted the 'generalized' goal X doesn't limits her mind and her behavior to this (self-regulation); she will also worry about the others' behavior:

- X is also able to have Goals about the others' behavior: she adopts the Goal not to do but that for any z (DOES z A).
-Given such an Adoption she has *expectations* (predictions + prescriptions) about the others behavior, and is not only surprised, but also 'disappointed' by their non-conformity.

Also because she is paying some cost for respecting the norm and the authority, for maintaining the prescribed social "order", which is supposed to be a "common". She wants the other be fair, reciprocates, contributes.

3.3 Equity and spreading

Conte and Castelfranchi [23] claim that the decision to conform to what is perceived to be an obligation plays a relevant role in its spreading over a population of cognitive agents. While the conventionalist view derives social norms from the spreading of conformity, in our view *conformity is derived*, so to speak, *from the spreading of obligation-recognition and -adoption*.
"The very act of accepting an obligation implies and turns into enforcing it. The agent respecting the obligation turns into a supporter. Conforming leads to prescribing. The agent undergoing an obligation becomes a legislator. The more an obligatory behavior is believed to be prescribed, the more it will be complied with, and the more, in turn, its prescription will be enforced. Rather than acting only through a behavioral contagion or a passive social impact, the spreading of norms is affected by *cognition* in a variety of ways and attitudes:

(i) It leads to implementing effective conformity. When an autonomous agent recognizes a norm as a norm and decides to conform to it, the number of conformers will be increased, and the norm is more effective.
(ii) *Effective conformity contributes to the spreading of normative beliefs*. The larger the number of conforming agents and the more likely the observers will form normative beliefs and the strength/certainty of the belief will increase.
(iii) *The spread of normative beliefs contributes to the spreading of normative actions.*
(iv) *The spread of normative actions contributes to the spreading of normative influence.* The larger the number of agents conforming to one given norm, and the more distributed will be the want that other agents will conform to the same norm. "This is due to:

- An *equity* rule. People do not want others in the same conditions as their own to sustain lower costs - benefits being equal (this is, indeed, one the most probable explanations of the Heckathorn's [34] group sanction control: the more agents respect the norms, and the more likely they will be to urge others to do the same).
- "*Norm-sharing*". Agents are likely to "share" the respected norms, that is, to believe that those norms are sensible, useful, necessary, etc. This is also a powerful self-defensive mechanism (agents share the norms they happened to respect). Agents will defend the norms they share, implementing the number of agents who want those

norms to be respected." [17].

(v) *The spread of normative influence contributes to the spreading of normative beliefs*, and the whole process is started again in a circular way.

The same cognitive mediation olds for an *observed* violation, deviance, and their crucial *interpretations* and meanings by the observer (see also Bicchieri & Mercier [7]).

Also for Agents this might be relevant: do we want/need just agent doing as expected/ordered or agent able to violate but also able to conform to the norm as a decision and for specific deontic motives/reasons (N-Adhesion). Don't want we to "share" norms (social, moral, legal) with our Agents? To really have a hybrid society regulated by values and norms?

4. *Internal Locus*: kinds of N mutation within Subjects' mind

Let's identify the various though and 'reasons' of the 'subject' (S) for *abandoning* or *violating* a given N. We will distinguish between:
(i) 'Unintentional' effects; where changing or weakening that N (or Ns) is not the end or an end of S, and
(ii) 'Intentional' act; where S understands, expects, and intends to jerk the N.

4.1 Unaware violations

S does not realize that her behavior is an N violation. Mental conditions for such a conduct:

- Ignorance of the N (*beliefs*); or
- A mistaken interpretation or instantiation (*beliefs*): S does not realize to be a member of the set of the addressees of that N or that it does apply in those circumstances and context; or
- No memory retrieval of the N in those circumstances, lack of attention, absent-mindedness ((*beliefs*).

The violation is unintended since it is fully unaware, but - given the observable behavior ("bad example") - it equally injures the N.
There are also extra-mental conditions facilitating or inducing such a "mistake". For example, the N and its local pertinence should have been appropriately and explicitly signaled, not given for obvious: "Please, do not park more than one car in our courtyard; this is our polite convention". [5]

4.2 Aware violations

A) *Without the goal of injuring/weakening the N*

As we do not intend the supportive 'function' of our conforming to the N, equally we do not necessarily intend the destructive 'function' of out violating it.
There are several *reasons for dropping* a N-goal, do not adhere to it and formulate a conform intention:

a) Goal-conflict: the N-goal contrasts with another goal of the agent;
Apart from the *belief* that the N is in conflict, what matters are the following parameters:
- value of the goal based on the value of the meta-goal of respecting Ns;
- value of the contender goal;

[5] An interpersonal example may be: X: "You can not go around in underwear!" Y: "But you had to say me that there were guests in our house!"

- value of the negative expected consequences of violation, including feelings associated to N-violation; and in particular the perceived threat: estimated probability and weight of 'punishment' and blame (*beliefs*). [6]

A sub-case of (a) is a N-conflict: N contrasts with other Ns accepted by the agent (see below).
The decision to violate if I can an N that is not convenient for me now and here (not necessarily "in general") can just be for my private interests. However, not necessarily the goal in contrast with the N is a private/personal one; it might be a goal formulate for efficiently perform S's role or mission [17]: violating for functional reasons, for an intelligent problem-solving in our work.

b) N Application & Instantiation disagreement: S is aware of N but he contests to be a member of the set of the addressees or that it *does apply* to that circumstances and context.

c) Material impossibility: S forms a N-goal but cannot comply with it (*beliefs*); the intention would be impossible (*beliefs*).
As we said, a remarkable case of (a) – but in a sense close to (c) (in terms of not "material" but of "deontic" impossibility) – is:

d) Norm conflict: the N I should apply and respect is in contrast (*beliefs*) with another N:

- Either another social N (social Ns are not so coherent and non contradictory, especially in their application). For ex. the social N about our male group meeting for drinking beer implies the possibility or prescription to burp in public (just for funny and be deviant), while I would desire – due to my "education" – do not burp;
- Or a conflict with legal or organizational N.

In all these cases S will not conform to the N but she is *not* motivated by the aim of weakening it. For sure that violation (given the message to myself and to the stakeholders) weakens the N, however the agent's intention is not necessarily this.

(e) Expectation of not sanctions
Either because some reason in the others of not sanctioning; or just because I expect to not be detected, to hidden: "I will get away with it; they will not see me; nobody will know that"; or "They do not catch any violator, they never punish"[7]. Of course, these beliefs are relevant in particular for agent motivated to respect Ns just by the fear of sanction.

(f) Indifference to sanction
There are cases and individuals where the fact that other people respect N and that there will be a negative judgment by the others (sometime even publically expressed), is not a sufficient reason for not violating: an important sub-kind of conflict. Consider for example a young guy sited in a waiting room where there are quite old waiting people standing up, and not giving up his seat to them, although he knows that he "should do" that, and that he is disapproved. Either there is in this guy (and context) indifference to the judgment and sanction from the others (*goals*), since "I do not care of these guys", "who knows them?" "I will never meet them again.." (*beliefs*). Or there might even be a provocation attitude (*goals*): "Yes! I'm not like you, I do not care of you", "I'm underbred, so what!?". Or the attitude is "motivated" by an opposition specifically to the N, as a meaningless N: a value opposition (like in people violating the rule of giving priority to women).

[6] This expectation should be part of what Bicchieri calls "empirical expectation" ("what we expect the other do"? However, we should distinguish between "to expect that the other conform" and "to expect that the others monitor and sanction". Two different predictions based on different experiences that might also don't be fully correlated.
[7] This is a change in our "empirical expectations" in Bicchieri & Xiao terminology [8].

All these are (more or less sincere and not self-deceptive) beliefs and motives of the violator.
Sometime we (unconsciously) find a new interpretation of framing of our action and circumstance, and of the N, in order to facilitate our violation. Consider the very famous and beautiful case of people "interpreting" the monetary sanction for the violation of the N as a fair, a price, and thus deciding to systematically violating it, and just pay what they have to pay [33]. Let's rewrite in our mind as a tax what in fact would be a fine! But this morally facilitates our decision to violate.

(g) Violation as epistemic act

I know and intend (in case) to violate, but my motive is to "see": to see if that N is there or if I correctly understood it; or to see if the violation will be noticed/punished; to see your reaction. Even to see if you know that N, not in order that I know the N, but in order to know if you know it.[8]

Of course, there are other kinds of assumptions and reasoning that induce or facilitate (intentional) N violation; in particular interpretations of observed deviant behaviors, changing our mind. We will see some of them below: the effect of external changes (observed deviant behaviors) on our mind and conduct.

B) *Aware violations with the goal of harming, breaking down the N*
Violation is not just intentional but motivating: I violate in order to violate (Ns or that N)

(h) Violating for changing

Intentional and public violation of N for rebellion and opposition to that N, for rejecting and breaking it; to send a *message* to the others, to the "authority". Like Gandhi that rips in a central place of Johannesburg in front of the police the special document obligatory of Indian people. The message (and belief) is "This N is discriminatory, unacceptable, unfair; it has to be abolished: rebel to it!"[9] Notice that I can violate an N as unacceptable, not fair even if it does not directly damage me.

(i) Violation against stigma, for changing values, building our identity

I violate for provocation and rebellion towards stakeholders' values and attitudes. There are two different cases.

A possible aim is to build our collective identity, to remark that "we" are different, not like you, and we do not want be part of you (like Punk's provocation; or adolescent deviant attitudes). We are not In-group, but Out-group; it is an "exit" or *secession move* from your value and community.

Another possible aim is to change your value, to obtain respect. Like the provocation of the Gay "pride" and exhibition. Our aim is not splitting from you; on the contrary we want to be accepted, integrated, and respected; you have to change your conservative values and thus your social Ns on that.

A crucial construct in human mind is the "sense of justice" and the related sufferance for iniquitous situations (not only harming us personally but even favoring us, or harming others: we can play the role of the victim, of the privileged guy, or of the stakeholder, but always with some discomfort) ("equity theory"), the *need* for equity (a "value" and a "motivation"[10]). We can consider a given N with this perspective, by evaluating its "equity and justice". This changes very much our disposition in obeying to it, or in supporting/defending it as punisher (§ 3.3). I feel "justified" in my violation; not a bad guy

[8] My behavior is like an exam question, where I in fact already know the answer but I want to know if you know it.
[9] This nice example is about a legal N, however similar examples exist also for social ones; like the "provocation" acts of courageous women in Arabic countries.
[10] For a rigorous cognitive notion of "value" and its strict link with evaluations, prescriptions and Ns see [38].

but a good guy; I do not feel guilty but proud of me.[11] If I consider a given N unfair I can have a serious conflict between two internal values, intrinsic motivations: the sense of duty/obedience vs. the sense of justice. The conflict is within my own values.[12] Sometimes this mental justification and motivation in terms of "sense of injustice" is just a convenient alibi (in front of the others, or in front of myself) for allowing my violation for personal advantages and desires (like the "sense of injustice" sometimes used for covering/hiding our envy).

(i) Violation to be noticed, to innovate

Sometime we violate a social Ns or consuetude's just to emerge, to be noticed, and to be original; like women first wearing a bikini or a mini. These provocative guys (actually innovators that may create a new "fashion", but not necessarily with this intention) are aware of and ready to cope with criticism and even insults.

Two examples about previous cases: I violate the norm that on the beach one cannot be nude, and (with other people) I use "topless"; so I create or converge a new use, imposing tolerance to the others (they can no longer blame and reproach me). Or I'm completely nude; but this is too disturbing, intolerable for that group, so this creates a scission of groups and places: you nudist must have tour own beach (and we will not come there!), but you cannot stay in "our" beach and be nude. If you become part of the new group and go to the nudist beach it become not just tolerate to be nude (the old N doesn't constrains you any longer) but there even is a new N of "being nude".

Similar path for vegans: the want not just be permitted to refuse current food without objection, ridiculous, blame, but they are trying to build new Ns - based on new values - ("Do not eat animals!" etc.) on such a basis to criticize, blame the violator (although they are the majority, and make propaganda. Their aim is not just to build a separate culture and community, but also to change the practices and the Ns of the big community.

Notice that this kind of N change requires (and is grounded on and aimed at) a change of "value" which is first of all a specific mental object.

(l) Against the authority as such

It is also possible to violate in order to rebel, but not against a given set of N that we want to reject or change, but against the normative authority A. To impair A, independently from the specific N. What maters is to violate; to show to myself or to my peer or to A that I do not respect A, do not submit: this is the message and motive. Like a "rebel" child that rejects any parents' prescription or restriction to his desires; like some political movement or demonstration where what matters is to broken something, to do something prohibited, not what to broken and why.

The crisis of the authority (see § 6.2) can be due to various assumptions and motives; like the fact that A is no longer credible, trustworthy, correctly and competently playing its role; so I do not want longer depend on and delegate to it. Or a crisis of identity and membership: I do no longer feel one of "you". Or for a crisis of values grounding that A, I do not feel any longer morally "obliged". And so on.

Again; it is not necessarily a matter of sanctions, power, and fear.

5. *External Locus*: The others' observed behaviors

Which and how many observed changes in normative behavior are necessary for changing our conform conduct? Not necessarily we need diffused and spreading practices. Even a

[11] Agents too should have some moral value and should be able at least to interpret our behavior and reasons in these terms, and possibly mediate our interaction caring of moral norms.

[12] This is Antigone tragedy. This also is Socrates' message to us while taking the poison: respecting Ns and authorities (even when their decision is incorrect and harming us) may/should be a prevalent value.

single violation act or meta-violation (for example do not monitoring or punishing) can call into question a given N in my mind (for example, a single resounding act of euthanasia); a single provocation can be enough for discredit authority (see Gandhi's example).
To know that somebody has violated N is an important factor in the crisis of that N. However, this passes through our mind and its changes, and what matters is the *interpretation* we give of that behavior: accidental? Intentional? And why? And which are the consequences?

Let's first see some examples/kinds of assumptions and reasoning that induce or facilitate (intentional) N violation; in particular interpretations of observed deviant behaviors, changing our mind:

(h) *Interpretations of observed deviant behaviors*
> "If he (they) is doing that me too I can do so! It is not *fair* that he does that and I cannot!"
> "If he (they) is doing that it *means* (it is a *sign*) that it is permitted/possible: there is not a N or is no longer in force here"
> "If he (they) is doing that it *means* (it is a *sign*) that this is the right way; what we have to do (he expects that I do so)".[13] Actually this is an intentional action entailing a violation, but not intentional *as* violation.
> "In fact he is right! He is courageous. It is correct to violate this N!" (Thanks to his violation behavior I change my value-attitude towards N; this goes in the direction of N criticism).

5.1 A single bad example

The impact of an external, observable deviating behavior does not depend only from the *number* of violators: the many the violators the more impaired the N.
A single guy's deviant behavior can be sufficient for a large impact. It depends on the network, on the number of stakeholders and – of course – on his/her role and influence
It is also important the fact that (a) *not all violations are equivalent, although behaviorally identical*; and (b) that sometimes a single deviating example (not a multitude) be enough for; but of course it depends on its visibility and significance and interpretation. The single violation of a leader is not the same of the one of a follower; the violation a well-known person is not like the violation of an anonymous person, and so on.
The number of violator is of course a relevant factor because one principle for the strength of our persuasion is *the number of converging sources or examples*. But also the single's reliability - as model or authority – and prestige has a precise impact on the degree of our persuasion.

5.2 The others (deviant) behaviors as messages

Since minds are typically read off behavior "it is impossible not to communicate" about our minds even those prescribed by a specific role. Our behaviors or their traces inevitably "signify" our mental attitudes. And we use our everyday behavior or its traces (practical actions not "expressive" ones or conventionalized gestures) on purpose to send this information to others, for *signaling*. This is a special form of communication crucial for human social coordination, and conventions and institutions establishment via "tacit" negotiation and agreement, not to be mixed up with gestural or other forms of non-verbal communication [41].[14]
Also N maintenance or innovation "circles" (observation-interpretation-change-action-observation- and so on) (§ 6) works thanks to the fact that a cognitive agent "reads" the

[13] This case and the previous one change our "normative expectation" in Bicchieri & Xiao [9] terminology.
[14] On the relevance of Norm-signaling, and also of explicit communication, not just of punishment, see also [2] [3].

others' conducts, and they signify/inform about the existence, respect, or violation of Ns [3]. Thus a violation conduct may acquire either the communicative function or the communicative intention of impairing the N or of explaining my reasons. Demolition or establishment of SocNs is mainly based on such a kind of not explicit communication, negotiation, and tacit agreements.
This factor contributes to the explanation of a crucial issue. As remarked by Christine Cuskley[15] "frequency and stability exhibit an interesting relationship in language: *the more frequent a linguistic construction is, the less it tends to change over time*." In my view this might be generalized to behaviors, and in particular to normatively regulated behaviors. Also linguistic constructions are "norms" and "rules" for people aimed at using that language; just a sub-case (with its specific additional dynamics). "Despite the evident relationship between frequency and stability, it is still unclear what specific social and cognitive factors underlie this relationship." As for social Ns, I would say that part of these factors is rather clear: the more diffused a (normative) behavior, the greater the probability to be observed and imitated/learned (a very strong and repeated "message"!), and thus *not just to spread around but to be "reinforced" in its prescriptive character*. Moreover, the more it is diffused the greater the absolute number of necessary "exceptions" and "violations" for its change or elimination. Thus the more widespread the more stable. And vice versa: the more stable in time and people, the greater the probability to be diffused and repeated (frequency). And so on.

6. Collective destruction/construction: Emergence-Immergence Cycles

On the basis of this analysis of internal mutations and their behavioral consequences, let's focus on the description of the internal-external, mental-behavioral, individual-collective loops, and on the description of the phases of Ns change (vicious) 'circles' (Fig. 1).

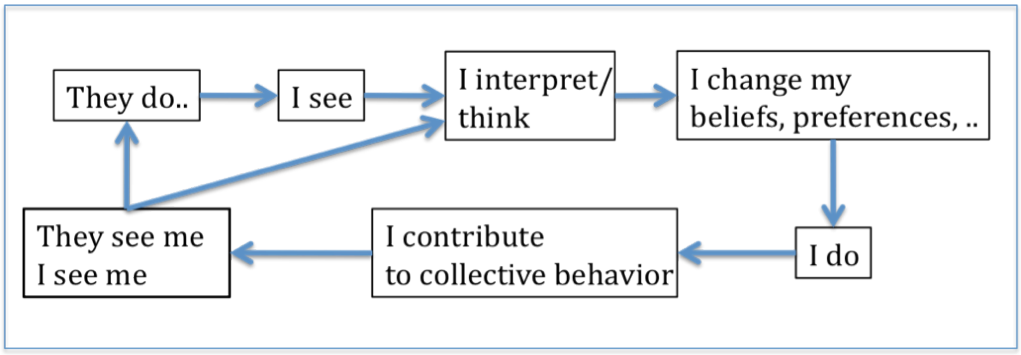


Fig.1 Internal-external cycle

6.1 External ⇔ Internal circles

Obviously – as for the "external" observed events (single or regular) – what matters is the Intentional Stance *interpretation*, the ascribed mind and reasons. I observed an individual violation by S or by W (not blame, no sanction); is it by accident, ignorance, or lack of attention? Or was it intentional? And "why"? Was S just egoist and self-maximizing, or is he violating because disagrees about the N or for invalidating the A? As we saw in § 5 there are various possible interpretations and effects. And about norm 'watchman' role: was he

[15] Christine Cuskley "Frequency and stability in linguistic rule dynamics", Invited seminar at ISTC October 2014.

indulgent because lazy or corrupted or familiar with S? Or was he thinking the N doesn't apply in that circumstance or was bad and unfair?
The effect on my mind and on my view of the N in the various cases is very different. The external event impact depends on our subjective interpretation of it.
That's why also a very clear collective behavioral regularity is not always and automatically interpreted (and complied) as a N. There are "vicious" and "virtuous" circles, from the point of view of normative behavior. Both, the vicious one (that is, violation, behavioral messages, N impairment, and collapse) and the virtuous one (N emergence, implicit negotiation, establishment, and maintenance) are due to the same internal-external cycle (fig. 1).
There is also a very interesting self-referential feedback: the violating or conforming subject is observing his/her own behavior, and interpreting it, and confirming or changing his/her beliefs and preferences and feelings (as we saw in § 4.), and so on. Our behavior signifies a lot to us, and we send (intentional or unintentional) messages to ourselves. Also because, if I act on the basis of some implicit, presupposed, assumptions or choices, and the action is successful (good results), this automatically reinforces the presupposed mental conditions for that act, and increases the probability to take the same path next time.

6.2 The crisis of N authority

A nice example of a multilayer vicious circle between normative behavior and norm-related mental attitudes is the crisis and discredit of the "authority". To work well authority requires not only respect/submission for authoritarian strength, threats, coercive power (credible sanctions), but "prestige" or more precisely "authoritativeness". That is, A's "credibility". An A requires trust for its role; without trust it cannot work. Information authority, source of knowledge must be "credible" in strict sense: it has to be perceived (evaluated and felt) as "competent" in that domain and honest, not cheating for some private interest. Analogously the norm-A must be "credible" and trustworthy, its Ns should be perceived/given as the right one (from a technical and a justice point of view) and not due to private interests. If the A is authoritative, I accept its information or prescription, without need for prices or threats, without conflict, rebellion: I have a *generalized adoption disposition*; in a sense I obey for intrinsic motivations.
However this authoritativeness can collapse, and A have a crisis of credibility, be discredited and no longer "automatically" respected. Which are in Individual Mind changes that might start (or reinforce) this process?

(a) I no longer *believe* that A or its behavior is respectable, that A is authoritative, credible; thus
(b) I do not adopt its prescription/N, I start do not conform to (*decision*);
(c) this feedbacks, and reinforce my belief about violability of N and my right to violate, and - since my deviating behavior can be observed –
(d) it discredits the A in the others' eyes; diffuses the same evaluation about A (and probably also its perceived capacity or right of sanctioning), builds a "collective belief"[16]
(e) it infects, diffuses deviating behaviors; but this spreading of the evaluations and of the deviating behaviors
(f) confirms and reinforce my perception of A, of that N, and my behavior; and so on.

The collapse of A's authoritativeness is a mental and behavioral, and internal and external, and individual and collective, fact.[17]

[16] Not in the sense of a "collective mind" but in the more basic sense of a collective of minds; many minds sharing certain assumptions and infecting each other.
[17] It is clear that such an internal/external dynamics of Ns change might be fully simulated only with cognitive Agents in MAS.

7. Concluding remarks

Three issues.

> As we said, Ns are based on the possibility to be violated, not obeyed. They are devices for the control of "autonomous" agents that decide what to do on the basis of their beliefs, reasoning, and goals. Ns not only presuppose (accept) but also postulate a freedom in the addressees. Is this just a not so good but unavoidable feature? Or violability in this regulating device of social conduct has some advantages? N "violation" usually has a negative connotation, since to "violate" is an evil in itself (as harm at a general and meta-level, of order, authority, trust; as we explained). However – actually – not only it can be morally justified and even noble and courageous, but also it plays a key function. It is one of the mechanisms and pressure for N change, adaptation, and evolution[18] [16].[19]

> I'm no sure that the current theory and definitions of social norms (see for example [35], [6]) fully captures some of the aspects we have discussed [20]. For example, there are social norms (not only legal ones) that are still there even if systematically violated by a large part of people. The norm *is still in force since it is perceived as such by that people*, although they violate it. They actually know/decide to "violate" it, and in a sense that N still "regulates" their conduct. For example, in several part of Italy it is very frequent that people throw papers on the street or do not collect the excrements of his dog; however, they know (and even agree) that this is bad, not "correct" (N violation), but since it is tiring do not do so, and since a lot of people does the same… Is that N "in force" in this group? Yes: everybody knows what one "should" do. In our view a social norm to be there doesn't require to be a behavioral norm, a stable practice. It is *sufficient* that the large part of the group knows it, reminds and considers it, although regularly or frequently violating it. It is perceived *as* a N, *taken into account* in the individual cognitive process and mentally shared in the group, although ineffective on the conduct. It is a strange N state: an still in force but ineffective N. We shouldn't forget that first of all a N is into the (shared) mind of the agents; this is its presupposition.

  Of course it is fully true – coherently with Bicchieri's theory – that:

  (i) On the one side the norm not only is ineffective but is probably in "decadence", close to disappearing also from the mind of people, for example for the learning process of the new guy or for the mental automatization of the bad practice without no longer considering/perceiving that you are violating.

  This is reasonably a possible and rather typical *intermediate step in the path of N extinction*: N respect and sanctioning; bad practices but the N is still considered as such; non longer taken into account as a N, no longer impinging on us.

[18] This obviously shouldn't be an excuse for the selfish violator just for his own private interests (although – as Adam Smith has explained – even this guy plays his social function, beyond his personal motives).

[19] I worry about the rigorous computational (intelligent) coordination and surveillance on human work and organization. At least in "critical states" we need violations, although not foreseen in the program; but just opportunistic and reactive to a given contingency.

[20] For example, the synthetic motto of Bicchieri for synthesizing the spirit and working of social Ns **"**Do the right thing: But only if others do so" could create some misunderstanding. This might be the mental rule, the prescription that the individual gives to himself in front of a N (it can explain his conformity or violating behavior) but is not the prescription of the N: the N says, prescribes, just "Do the right thing!" Ns want to be obeyed and respected in any case; this is their imperative. I may decide or be leaning to respect this absolute imperative only "if", under certain condition, but the "normative expectation" also by the others doesn't say "only if the others do so".

(ii) On the other side, it is true that the fact that several guy systematically violate that N encourages ignoring it, to consider that it is possible and not so terrible to violate it. We live in a rude world and we adapt/belong to it.[21]

> Agents are relevant in two ways: for modeling the complexity of such a dynamic and immergent/emergent process, by Agent-based Social Simulation; but also because we need non-passive normative and moral agents in Hybrid Societies where Artificial Intelligences (Agents, robots,) will work and cohabit with humans. In particular *N change* processes (internal and external) should be present in both MAS with cognitive Agents, and in Hybrid Societies. We have even to allow and exploit violations of rules and practices in organization, coordination, and work, but only when it is the case and by understanding "why" (reading behavior and mind) [17]. Actually there is a strong and advanced tradition in AgMAS on Agent architecture for Ns, in N based MAS and organization, in MAS simulation of Ns efficacy[22], however – in my view – we still need some advancements in theoretical modeling of cognitive and collective aspects of Ns dynamics. This work is a partial attempt in this direction.

References

[1] Andrighetto,G., Governatori,G., Noriega, P., van der Torre, L.W. (2013a) Normative Multi-Agent Systems. Dagstuhl Follow-Ups 4, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik 2013.
[2] Andrighetto G., Brandts, J., Conte, R. , Sabater-Mir, J., Solaz, H., Villatoro, D. (2013b) Punish and Voice: Punishment Enhances Cooperation when Combined with Norm-Signalling. PLOS.
http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0064941
[3] Andrighetto G. and Castelfranchi, C. The Prescriptive Power of Normative Actions. *Paradigmi*, 2, 2013.
[4] Alexander Artikis (2009) *Specifying norm-governed computational societies*, Published by ACM 2009
[5] Bendor, Jonathan and Piotr Swistak. 2001. *The evolution of norms*. American Journal of Sociology 106 (6):1493–545.
[6] Bicchieri, C. (2006) *The Grammar of Society: Nature and Dynamics of Social Norms*, N.Y.: Cambridge U.P.
[7] Bicchieri, C. and Mercier, H. (2009) Norm and Beliefs: How Change Occurs. In B. Edmonds(ed.) The Dynamic View of Norms, Cambridge University Press, 2009
[8] Bicchieri, C. & Xiao, E. (2009) Do the Right Thing: But Only if Others Do So. *Journal of Behavioral Decision Making*, 22: 191–208 (2009)
[9] Bicchieri, C. Xiao, E., and Muldoon, R. (2011) Trustworthiness is a social norm, but trusting is not. *Politics, Philosophy & Economics* 10: 170
[10] Guido Boella, Pablo Noriega, Gabriella Pigozzi, HarkoVerhagen:Introduction to the special issue on NorMAS 2009. J. Log. Comput. 23(2): 307-308 (2013)
[11] Boyd, R., Gintis, H., Bowles, S., Richerson, P.J. (2003) The evolution of altruistic punishment. *PNAS*, v. 100 (6); Mar 18, 2003

[21] It is even possible that a meta-norm emerges: the idea that to conform to this N (for example, of politeness) by antiquate, ridiculous, or snob, and this elicits negative attitudes in the others, that I want to avoid. A sort of *meta N of not conforming to the traditional N* is emerged; sometime even justified by new value (for example, "do not give precedence to women" as sign of women discrimination). In this case, for those people the previous N is no longer there, is no longer considered and accepted as a social N. The emergence or formulation of a *meta-N* about the violation (and then abandon) of a previous specific N is one of the processes of N abandoning and N innovation. It requires specific mental changes and contents; including a value-based justification of the "criticism" to the previous impinging N.
[22] See for example: [1], [39], [10], [26] [27], [36], [43], [28] [29].

[12] Geoffrey Brennan, Lina Eriksson, Robert E. Goodin, & Nicholas Southwood. *Explaining Norms*, by Oxford: Oxford University Press, 2013
[13] Campennì, M., Andrighetto, G., Cecconi, F., Conte, R. (2009) Normal = Normative? The role of intelligent agents in norm innovation. *Mind & Society* (2009) 8:153–172
[14] Castelfranchi, C. (1998) Through the minds of the agents, *Journal of Artificial Societies and Social Simulation*, 1(1), 1998
http://www.soc.surrey.ac.uk/JASSS/1/1/contents.html
[15] Castelfranchi, C. (1999). Prescribed Mental Attitudes in Goal-Adoption and Norm-Adoption. *AI &Law*, Special Issue on Norms in MAS. 7, 1999, 37-50.
[16] Castelfranchi, C. (2001). The theory of social functions. Challenges for multi-agent-based social simulation and multi-agent learning. *Journal of Cognitive Systems Research* 2, 5-38. Elsevier. http://www.cogsci.rpi.edu/~rsun/si-mal/article1.pdf
[17] Castelfranchi C. (2004). Formalising the informal?. *Journal of Applied Logic, N° 1,*
[18] Castelfranchi, C. (2013) *Cognitivizing Norms. Norm Internalization and Processing*. In *Informatica e Diritto*, Vol. XXII n. 1; Special Issue in "Law and Computational Social Science", S. Faro & N. Lettieri (Eds.); pp. 75-98.
[19] Castelfranchi, C., Tummolini, L. (2003). Positive and Negative Expectations and the Deontic Nature of Social Conventions, in *Proceedings of the 9th International Conference of Artificial Intelligence and Law* (ICAIL 2003), ACM Press: 119-125.
[20] Cialdini, R. (2007). Descriptive social norms as underappreciated sources of social control. *Psychometrika*, 72(2), 263-268.
[21] Conte R., Andrighetto G., Campennì M., Paolucci M. (2007), "Emergent and immergent effect in complex social systems", *Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence*, Washington.
[22] Conte, R. e Castelfranchi, C. (1995). *Cognitive and social action*. London, UCL Press.
[23] Conte, R. & Castelfranchi, C. From conventions to prescriptions. Towards a unified view of norms. *Artificial Intelligence & Law*, 1999, 3, 323-40.
[24] Conte, R., Castelfranchi, C. (2006). The Mental Path of Norms. *Ratio Juris*, Volume 19, Issue 4, Page 501-517, Dec 2006
[25] Conte, R., Castelfranchi, C., Dignum, F. (1999) Autonomous Norm Acceptance. In J. Mueller (ed) *Proceedings of the 5th International workshop on Agent Theories Architectures and Languages,* Paris, 4-7 July, 1999.
[26] Natalia Criado, Estefania Argente, Pablo Noriega, Vicente J. Botti. MaNEA: A distributed architecture for enforcing norms in open MAS. Eng. Appl. of AI 26(1): 76-95 (2013)
[27] Natalia Criado, Estefania Argente, Pablo Noriega, Vicente J. Botti: Corrigendum to 'Human-inspired model for norm compliance decision making' [Inform. Sci. 245(2013) 218-239]. Inf. Sci. 258: 217 (2014)
[28] Dignum, Frank (1999) "Autonomous agents with *norms*." Artificial Intelligence and Law 7.1 (1999): 69-79
[29] F. Dignum, V. Dignum (2009). Emergence and enforcement of social behavior. In Anderssen, R.S., R.D. Braddock and L.T.H. Newham (eds) 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, July 2009, pp. 2377-2383.
[30] Fehr, E., Fischbacher, U., & Gächter, S. (2002). Strong reciprocity, human cooperation, and the enforcement of social norms. *Human Nature, 13*, 1–25.
[31] Galoob, Stephen and Hill, Adam, "Norms, Attitudes, and Compliance". *Tulsa Law Review,* (January 23, 2015*).* Forthcoming.
[32] Garfinkel, H. (1963). A conception of, and experiments with, 'trust' as a condition of stable concerted actions. In O. J. Harvey (Ed.), Motivation and social interaction (pp. 187–238). New York: The Ronald Press.
[33] Gneezy U. & Rustichini, A. (2000) A fine is a price. *Journal of Legal Studies, Vol. 29, No. 1, January 2000*

[34] Heckathorn D (1988) Collective sanctions and the compliance norms a formal theory of group-mediated social-control. Am J Soc 94:535–562
[35] Hechter, M. & Karl-Dieter Opp, eds. (2001). *Social Norms*, New York: Russell Sage Foundation.
[36] Jomi Hubner, Eric Matson, Olivier Boissier, Virginia Dignum (Editors): Coordination, Organizations, Institutions, and Norms in Agent Systems IV, Proceedings of COIN IV, LNAI 5428, Springer, 2009
[37] Kahneman, D. & Miller, D.T. (1986). Norm Theory: Comparing reality to its alternatives. *Psychological Review, 80*, 136–153.
[38] Miceli, M. & Castelfranchi, C. (1989) A Cognitive Approach to Values *Journal for the Theory of Social Behaviour*, Volume 19, Issue 2, Page 169-193, Jun 1989.
[39] Pablo Noriega, Amit K. Chopra, Nicoletta Fornara, Henrique Lopes Cardoso, Munindar P. Singh (2013)Regulated MAS: Social Perspective. Normative Multi-Agent Systems 2013: 93-133
[40] Schultz, Nolan, Cialdini, Goldstein, Griskevicius. (2007). The constructive, destructive, and reconstructive power of social norms. *Psychological Science,* 18(5), 429–434
[41] Tummolini, L., Castelfranchi, C.: Trace Signals: The Meanings of Stigmergy. E4MAS 2006: 141-156
[42] Ullmann-Margalit, E. (1977). *The Emergence of Norms*. Oxford: Oxford University Press.
[43] Wamberto Weber Vasconcelos, Andrés García-Camino, Dorian Gaertner, Juan A. Rodríguez-Aguilar, Pablo Noriega: Distributed norm management for multi-agent systems. Expert Syst. Appl. 39(5): 5990-5999 (2012)

Simulating Normative Behaviour in Multi-Agent Environments using Monitoring Artefacts

Stephan Chang and Felipe Meneguzzi

School of Computer Science, Pontifical Catholic University of Rio Grande do Sul – Porto Alegre, Brazil
stephan.chang@acad.pucrs.br,felipe.meneguzzi@pucrs.br

Abstract. Norms are an efficient way of controlling the behaviour of agents while still allowing agent autonomy. While there are tools for programming Multi-Agent Systems, few provide an explicit mechanism for simulating norm-based behaviour using a variety of normative representations. In this paper, we develop an artefact-based mechanism for norm processing, monitoring and enforcement and show its implementation as a framework built with CArtAgO. Our framework is then empirically demonstrated using a variety of enforcement settings.

1 Introduction

Multi-Agent Systems are often used as a tool for simulating interactions between intelligent entities within societies, organisations or other communities. This Agent-based Simulation is useful for studying social behaviour in hypothetical situations or situations that may not be easily reproduced in the real world. The entities being simulated, human or otherwise, are represented by programmable intelligent agents, which must present reactive, pro-active and social behaviour [1].

When working with social simulations, we must consider that agents should be free to act in their own best interest, even though their actions might produce negative effects to other agents. For this reason, rules are established to ensure that certain actions, which would otherwise harm the society's performance, are prohibited. These rules, referred to as "norms" in multi-agent environments, allow agents to reason and act freely, while still being subject to punishment in the event that a norm is violated [2]. Agents are able to reason whether following a norm brings more positive results by avoiding the penalties associated to its violation. Some mechanisms [3][4][5][6][7][8][9][10] exist that makes reasoning about norms possible. In order to simulate norm-based behaviour, a structure must be defined that allows the specifying of norms in the form of prohibitions and obligations. Once these norms are active, agent interactions shall be observed by a monitoring mechanism and analysed by a norm-enforcing agent, which will then punish agents caught violating norms.

Although there are multiple frameworks for simulating agent-based behaviour, such as MASSim [11] or the agent programming languages Jason [12] and JADE

[13], relatively less attention has been focused on frameworks for norm-based behaviour simulation [14, Chapter 1]. In this paper, we aim to bridge this gap by developing a scalable norm processing mechanism that performs monitoring and enforcement in multi-agent environments. Our contributions are a mechanism to monitor agents's actions in an environment, described in Section 4.4 and a mechanism for norm maintenance and enforcement, described in Section 4.5. In Section 5 we show empirical results of applying our mechanism to a Multi-Agent System.

2 Multi-Agent Simulation

When intelligent agents [1] share an environment, competition between them becomes inevitable [15]. This idea becomes clear when we think of multi-agent systems as societies. Each person in a society has their own goals and plans to achieve them, and it is in their best interest to do so by spending as little effort as possible. Take for an example a person interested in eating an apple and another interested in selling one. For the buying person, its goal is to acquire the apple from the seller for the lowest cost possible, preferably with no cost at all. For the seller, the goal is to sell the apple for as high a price as affordable by the buyer, maybe even higher than that. Now, considering that in this hypothetical world no notion of ethics is known yet, the buyer soon realizes that instead of paying for the apple he wants to eat, he could simply grab it and eat it on the spot.

Competition between agents is often intended when working with agent-based simulations, as we desire to see how agents perform under such circumstances. However, to prevent the system as a whole from descending into chaos, rules must be established in order to control agent interactions while still allowing them to be autonomous. Nevertheless these rules must be limited to directing agents, rather than restraining them, otherwise, much of the benefit from autonomous agents is lost. When rules are set, agents that disregard them are subject to punishment for potentially harming the environment. In our buyer/seller system, we could establish a rule that guarantees items sold at shops must be paid for. If one is caught stealing, it will need to pay for the seller's injury. By doing so, we allow the buyer to reason about the advantages and disadvantages of obeying rules, letting it decide on an appropriate action plan. In multi-agent systems, we refer to these rules as norms.

Usual mechanisms for controlling agent interactions include interaction models, used by simulators such as NetLogo [16], MASON [17] and Repast [18]; strategies, commonly used in Game Theory; and regimented normative systems, such as $\mathcal{M}$oise [19]. The disadvantage of these methodologies is that agents are constrained to the rules of their environment. They are not allowed to break rules because the system is rule compliant by design, also known as the regimentation approach [20]. However, unlike environmental constraints, perfectly enforcement (regimentation) for social norms is both undesired, because it prevents agents from occasional violations for the greater good, and unrealistic, as it is not achievable in the real world.

3 Normative Scenario - Immigration Agents

To facilitate explanation and exemplification of our approach, as well as to highlight its capabilities, we present the scenario that was used to test our solution. This scenario helps understand what norms are and how they control interactions in an environment. First, we present a short story that connects the environment to its agents, then we outline the norms that constrain them.

In a fictional emerging nation[1], an immigration program was started by the government to accelerate development through the hiring of foreigners. Besides landed immigrants, visitors are also welcome to the country, since money from tourism greatly boosts local economy. At the border, immigration officers are tasked with the inspection of immigrant's passports. The foreigner acceptance policy is quite straightforward, and immigrants with valid passports and no criminal records are to be accepted immediately, while John Does and refugees are to be outright rejected. It is believed that the more immigrants accepted, the better. Each officer's responsibility is to accept as many immigrants as possible, while still following the guidelines that were passed to them. Each accepted able worker nets the officer 5 credits, which eventually turn into a bonus to the officer's salary. There are no rewards for rejecting immigrants. It becomes clear that the bonus each officer accumulates depends entirely on chance, and some officers may accumulate more than others, if at all. As such, some officers might feel inclined to accept immigrants they should not, only to add to their personal gain.

To ensure officers act on the best interests of the nation only, an enforcement system is introduced to the offices at the borders. Among the officers working in the immigration office, one is responsible for observing and recording the behaviour of those working in booths. This officer is known as the "monitor". His job is to write reports about what the officers do and send these reports to another officer, known as the "enforcer". The enforcer then reads the reports that are passed to him and look for any inconsistencies, such as the approval of an illegal immigrant. As this represents a violation of a rule, or norm, the enforcer then carries out an action to sanction the offending officer. The penalties for approving an illegal immigrant are the immediate loss of 10 credits and suspension of work activities for up to 10 seconds. Considering that immigrants arrive at a rate of 1 per 2 seconds, in a 10-second timespan 5 immigrants would have arrived at a given booth, meaning that a violating officer potentially loses 25 credits. Added to the other portion of the sanction, the potential loss rises up to 35 credits.

The enforcement system, however, is not cost free. Each monitor and enforcer has an associated cost and it is within the interests of the nation to spend as little as possible with such a system. Therefore, the government wants to know how intensive the system must be to cover enough cases of disobedience so that officers will know violating norms is a disadvantage rather than an advantage.

[1] Inspired by the game "Papers, please": `http://papersplea.se`

There are two norms that can be extracted from this scenario. These defined in Examples 1 and 2, which are detailed in Section 4.2, where we describe the mathematical representation of norms in our system. These norms concern the stability of the immigration program by assuring valid immigrants are accepted and discouraging corrupt officers to accept those who should not be.

Example 1. "All immigrants holding valid passports must be accepted. Failure to comply may result in the loss of 5 credits."

Example 2. "All immigrants holding passports that are not valid must not be accepted. Failure to comply may result in the loss of 10 credits and suspension from work activities for up to 10 seconds."

4 NormMAS Framework

In this section, we develop our normative monitoring and enforcement framework for agent simulation. We start by reviewing the agent approaches that underpin our framework in Section 4.1. We follow with the formalisation of the norms processed in our system in Section 4.2, as well as the way actions are represented in the environment in Section 4.3. With the formalisation in place, we proceed to explaining the monitoring and enforcement systems in Sections 4.4 and 4.5, respectively.

4.1 Jason and CArtAgO

In order to show the feasibility of the mechanism proposed in this paper, we use two programming approaches: agent-oriented programming and environment-oriented programming. The former is provided by the Jason interpreter [12], while the latter is achieved with the Common Artifact infrastructure for Agents Open environments (CArtAgO) [21].

Jason provides us with a means to program agents using the *AgentSpeak* language [22] in a Java environment. Agents are built with the BDI [23] architecture, and so their behaviour is directed by beliefs, goals and plans. Beliefs are logical predicates that represent an agent's considerations towards its environment. Predicates such as `valid(Passport)` and `wallet(50,dollars)` indicate that the agent believes the given passport variable is valid and that his wallet currently contains 50 dollars. In *AgentSpeak* variables start with an upper-case letter, while constants start with lower-case.

Goals are states which the agent desires to fulfil, and these can be either *achievement goals* or *test goals*. Achievement goals are objectives or milestones that agents pursue when carrying out their duties. To represent these in *AgentSpeak*, the goal's name is preceded by the '!' character. Test goals, on the other hand, are questions an agent may ask about the current state of the environment. These can be identified by a '?' preceding the goal's name.

To achieve these goals, agents need to perform sequences of actions that modify the environment towards the desired states. This sequence of actions

is referred to as a *plan*. A plan is not necessarily composed solely of actions, however, it can also contain sub-plans. This allows complex behaviours to be built, creating flows of actions that vary and are influenced by agent beliefs and perceptions.

As with any other programmed system, multi-agent systems must be tested before being effectively deployed to their end environments. To do so, test environments can be programmed for agents to be observed and any faulty behaviour addressed before release. Jason allows the programming of test environments in Java language, by providing an interface between agents and the programmed environment. These environments, however, are centralised, and so they are meant for small systems or specific test scenarios. This hinders scalability, which is an important aspect to consider when working with complex, more realistic scenarios or simply more robust structures. To address this limitation, we use the CArtAgO framework for environment programming.

In CArtAgO, environments are seen as composed by different artefacts. These artefacts represent objects in the environment through which agents interact with one another indirectly. *E.g.* a table in an office, which an agent may put reports on and from where another agent may pick these reports up to read them. The environment then becomes an abstraction, composed of different artefacts, which may be introduced to or removed from the environment whenever convenient. In our work, this allows us to create artefacts specifically for monitoring and enforcement tasks. These normative artefacts are then shared between normative agents so that more monitors and enforcers may be added to the system as it scales up.

4.2 Norms

In order to keep competition between agents manageable, norms are established to direct agent behaviour so that an environment's stability is maintained. This is achieved by specifying obligations and prohibitions [6]. Here, obligations are behaviours that agents must follow in a given context to comply with the norm, and prohibitions behaviours that jeopardise the environment's stability, and so must be avoided. Violating prohibitions is just as harmful as violating obligations, hence both cases must be addressed when detected. We expect that, when agents are punished for transgression, they are able to learn not to misbehave. Examples 1 and 2, in Section 3, correspond to an obligation and a prohibition, respectively.

While norms in the real world are expressed in natural language, they must be translated to a multi-agent environment so that agents are able to reason about them. This requires the extraction of necessary information related to a norm and composition of a mathematical representation. Agents should not have to reason how or why a certain norm came to be, but rather what the norm is about and what are the consequences of violating it. The format can also be extended to include other important information, such as the sanction function associated with a norm's violation, or the conditions for automatic activation

and expiration of the norm [6]. In this paper, norms as specified according to the tuple of Definition 1.

Definition 1. *A norm is represented by the tuple* $\mathcal{N} = \langle \mu, \kappa, \chi, \tau, \rho \rangle$*, where:*

- $\mu \in \{obligation, prohibition\}$ *represents the norm's modality.*
- $\kappa \in \{action, state\}$ *represents the type of trigger condition enclosed.*
- χ *represents the set of states (context) to which a norm applies.*
- τ *represents the norm's trigger condition.*
- ρ *represents the sanction to be applied to violating agents.*

Using Definition 1, we can proceed to formalising the norms from our example. We can formalize the first norm of our scenario from Example 1, as shown in Example 3.

Example 3. $\langle obligation, action, valid(Passport), accept(Passport), loss(5) \rangle$

The process can be repeated for Example 2. By identifying the context of a norm, it is possible to define it solely with predicates and atoms, as shown in Example 4, below.

Example 4. $\langle prohibition, action, not\ valid(Passport), accept(Passport), loss(10) \rangle$

4.3 Action Records

Similarly to norms, actions must also be stored as tuples containing essential information. Actions captured by monitors must only be accessed by agents of the enforcer type, and therefore only the pieces of information that can be associated with norms are deemed essential. These are: what was done; who did it; and under what context it was done. Example 5 shows how a monitor reports its observations to an enforcer:

Example 5. "Officer John Doe approved Passport #3225. The passport was known to be valid."

From this report, we can extract the following details:

Example 6. $\langle johndoe, approve(Passport), valid(Passport) \rangle$

In this example, an officer approves the entry of an immigrant holding a valid passport. The next report reads:

Example 7. "Officer John Smith approved Passport #2134. The passport was not known to be valid."

From this report, we can extract the following details:

Example 8. $\langle johnsmith, approve(Passport), notvalid(Passport) \rangle$

As such, we define Action Records:

Definition 2. *An Action Record, stored within the Action History, is represented by the tuple:* $\mathcal{R} = \langle \gamma, \alpha, \beta \rangle$*, where:*

- γ *represents the agent executing the action;*
- α *represents the action being executed by the agent* γ*; and*
- β *represents the agent* γ*'s internal state at the moment of execution.*

4.4 Monitoring System

The monitoring system has two roles: capturing agents's actions and forwarding reports, employing a producer/consumer mechanism. An action is captured whenever any agent successfully executes an action. In CArtAgO, this means that each operation executed successfully is saved to the Action History. This function is agent independent and thus it is implemented directly in the simulation engine's architecture. In our framework, we use an adapted agent architecture for Jason agents acting in CArtAgO environments and extend it so all successful actions are stored in a separate data structure referred to as the *Action History*. Should an action fail for any reason, it is ignored by the capturing system.

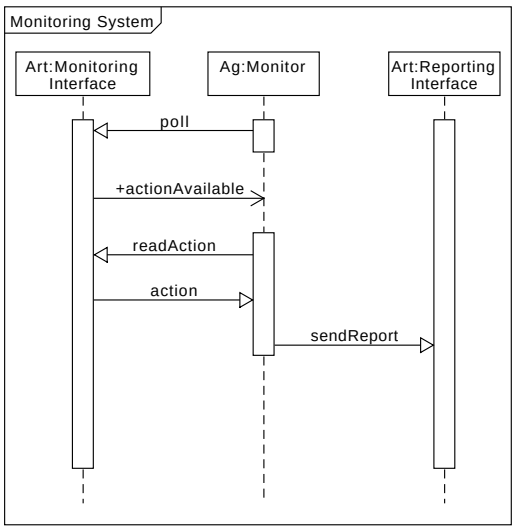


Fig. 1. Monitoring System Sequence Flow.

For these actions to be analysed, they must be sent to an enforcer agent in the form of a report. To achieve that, we use producer/consumer dynamic, in which an agent is tasked with continuously providing information through a channel, while another agent consumes this information. With this in mind, we can identify four components that are necessary for this set-up: a Producer, a Consumer, a channel for communications and the information itself. In our context, the role of Producer is given to the Monitor Agent; the role of Consumer is given to the Enforcer Agent; the communications channel is an interface called "Reporting Interface"; and the information that transits through this channel are reports containing the actions executed by agents. This process is illustrated in Figure 1.

Monitoring tasks are not cost-free, and monitoring costs grow with its intensity [24]. For this reason, it must be possible to adjust monitoring intensity so that enforcement can be performed effectively at a cost considered affordable by the society. Adjustments can be made either by configuring the monitor's enforcement intensity (inducing some desirable probability of reading actions) or by creating monitoring strategies. In this paper, we use a purely probabilistic strategy to study the general behaviour of our simulation.

The **Action History** is a queue-like data structure that stores captured actions, from which monitors gather the information that is sent to the norm enforcers. Actions are stored in the format discussed in Section 4.3 and are removed from the queue as soon as a monitor attempts to read them, regardless of the monitor's success when doing so. This represents the chance a violation will go unpunished.

4.5 Enforcement System

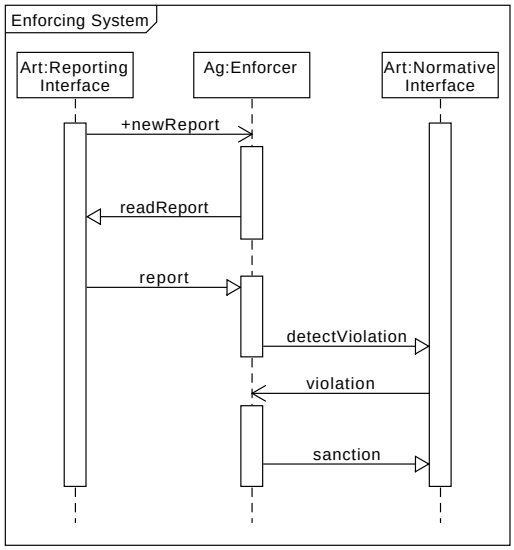


Fig. 2. Enforcement System Sequence Flow.

The enforcement system represents the Consumer entity in the normative mechanism's Producer/Consumer scheme. An enforcer agent connects to the Reporting Interface and awaits the arrival of new reports to analyse. The arrival of new reports is perceived by the enforcer, and in our implementation this perception is mapped to the `+newReport` signal. Once the report submission is

perceived, the enforcer accesses the Normative Interface in search of currently activated norms and checks for any possible violations by the reported action.

During the violation detection routine, the perception of violations is also mapped to a signal, represented in the sequence diagram of Figure 2 as the `+violation` event. When a violation is perceived, it falls to the enforcer to apply associated sanctions. The sanctioning step is the last in this process, and it starts as soon as detection finishes.

In order to sanction violating agents, the normative mechanism must be able to recognise them. It does not make sense to be told "John has approved an invalid passport. He violated a norm." if we do not know who John is in the first place. Therefore agents must be registered to the normative system prior to execution of their designed plans, similar to how people are registered for government issued IDs. In CArtAgO, this is accomplished through an operation in the Normative Interface that adds the agent's ID to a list, so that they may be found when needed. The ID they are registered with should be the same that appears in Action Records.

Normative Base When norms are created, they must be stored within the system so that they may be accessed by an enforcer attempting to detect violations. The Normative Base structure holds all the norms that exist in the system, active or not. Every time a norm is created, it is stored in a list structure with a unique identification. Norms may be activated or deactivated through the Normative Interface. Every time a norm is created, activated, deactivated or destroyed, agents connected to the Normative Interface perceives the event.

Detecting Violations The detection operation runs for each action report received by an enforcer agent. Each action read is verified against the normative base, along with the context under which the action was executed. Since it is possible for an action to violate more than one norm, we utilize a list structure to take note of all violations detected so they will be properly addressed at a later time. At first, no norm is seen as violated and thus the list is empty. A norm is only added to the list when all verification steps finish with the variable's $isViolated$ value set to $True$. The procedure for detecting violations can be seen in Algorithm 1 and is explained further.

Detection of violations can be achieved in two steps: context analysis and trigger condition analysis. Context analysis is about making sure that the action's execution context is the same as the one predicted by a norm. If it is, then there is a possibility of violation and further analysis is required. Otherwise, violation is considered an impossibility and the routine carries on. Formally, we define the norm's context as χ and the acting agent's belief-base as β. Hence, the context analysis returns $True$ value if $\chi \subseteq \beta$. Algorithm 2 is used for comparing sets of predicates. It checks if all the predicates defined in context χ are present in the agent's belief-base β, one by one. If a predicate in χ is negated (*e.g* `not valid(Passport)`), then the algorithm checks for its absence in belief-base β in-

stead. This is to reflect how `not` operator works in Jason. If the trigger condition is satisfied, the routine returns $True$ value, and $False$ otherwise.

Algorithm 1 Violation detection algorithm.

```
1: function DetectViolation(⟨γ, α, β⟩)
2:     V ← [ ]
3:     for each n = ⟨μ, κ, χ, τ, ρ⟩ ∈ ActiveNorms do
4:         if ContextApplies(χ, β) then
5:             if ConditionApplies(κ, τ, α, β) then
6:                 if μ = prohibition then
7:                     V ← V ∪ {n}    ▷ Violation detected! Adds to the list of violated norms.
8:             else
9:                 if μ = obligation then
10:                    V ← V ∪ {n}    ▷ Violation detected! Adds to the list of violated norms.
11:    for each n ∈ V do
12:        SignalViolation(n, γ)
```

A trigger condition of a norm can be either an action that was executed or a state the agent has reached. This is specified by the norm's trigger condition type and directs the way in which the detection algorithm executes. If we are working with an action trigger, then we must compare the action that was executed with the one specified by the norm. However, if we are working with a state trigger, then two contexts must be compared: the agent's belief-base and the norm's state trigger condition. These are compared using the context analysis algorithm of Algorithm 2. The pseudo-code for the trigger analysis procedure can be seen in Algorithm 3.

Algorithm 2 Context comparison sub-routine.

```
1: function ContextApplies(χ = [l_1, ..., l_n], β = [l_1, ..., l_n])
2:     Require count(χ) ≤ count(β)
3:     for each p ∈ χ do
4:         isPresent ← False
5:         checkAbsence ← False
6:         if p is of the form ¬φ then
7:             p ← φ
8:             checkAbsence ← True
9:         for each l ∈ β do
10:            if l = p then
11:                isPresent ← True
12:                break
13:        if checkAbsence = isPresent then
14:            return False
15:    return True
```

When both context and trigger conditions are satisfied, we need only verify whether the norm is an obligation or prohibition to conclude if it was violated or not. A prohibition means that a certain action or state is undesired under the given context. If all the conditions up to now have been met, we conclude that said undesired state has been reached and the norm was violated. On the other hand, an obligation requires the flow specified by the norm to be followed strictly, and if this is the case, we conclude that the norm was complied with. By negating our conditions, we also negate its results: if in a prohibition context the conditions were not met, then we would be home free; if they are not met while in an obligation context, however, we would have just violated it.

Algorithm 3 Trigger condition analysis sub-routine.

```
1: function ConditionApplies(κ, τ, α, β)
2:     if κ = action then
3:         return τ = α
4:     return ContextApplies(τ, β)
```

Their modality notwithstanding, every norm that is violated is added to a list that is processed when all norms have been verified. Sanction functions are then executed and agents perceive their punishments. Penalties can be brought directly upon agents through perception or carried out by a third party, while records on agent transgressions can be maintained in a separate structure for greater consistency.

5 Evaluation

In order to test our solution, we programmed agents using Jason and deployed them in a CArtAgO environment following the scenario described in Section 3. To visualise the difference between compliant and non-compliant behaviours, two types of agents were used: the *normal* type and the *corrupt* type. The normal type is programmed to approve only those passports that are truly valid, whereas the corrupt one will approve passports indiscriminately for his own personal gain. By making it so, we can more easily tell the effectiveness of the norm enforcing mechanism. Therefore, the following results were expected:

- Corrupt agents attain more credits when under lower monitoring intensity.
- Standard agents maintain an average quantity of credits through all simulations.
- At some point, corrupt agents should start performing poorly due to higher monitoring intensity. This marks the point at which monitoring can change the environment.

We ran 35 simulations for 11 different values of monitoring intensity. Intensity values range from 0 to 100, with a step value of 10. Each simulation was run for 10 minutes. In this timespan, with our set-up, around 1048 immigrants attempt to cross the border. In what follows, we refer to an agent's obtained credits, or their performance measure, as their utility. We use that measure in the graph of Figure 3, which illustrates how the environment's monitoring intensity affects the utilities of corrupt agents 1 and 2. The monitoring intensity is the probability as a percentage of a monitor to read an agent's action. A value of 100 means that all actions are read, while a value of 0 means no actions are read by the monitor. We notice that, as the intensity of the monitoring mechanism increases, the utility of corrupt agents decreases to the point where performing badly and not performing at all yield the same utility, whereas normal agents maintain their average utility. This allows us to conclude that, for a monitoring intensity value of 40 or more, following norms is a better decision than the contrary.

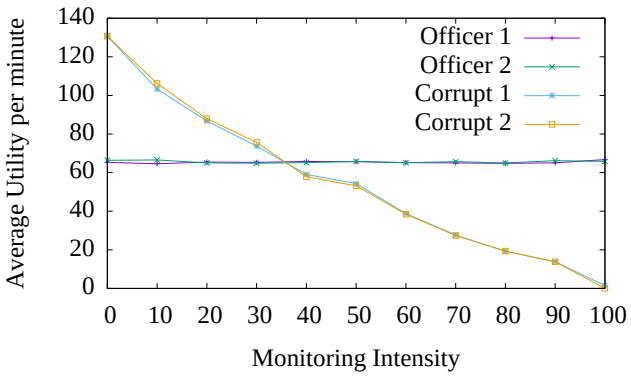


Fig. 3. Utility of corrupt agents is affected by monitoring intensity.

The data used to plot the graph of Figure 3 can be seen in Table 1. Values for μ and σ represent the arithmetic mean and standard deviation, respectively. These were calculated to show that utility values for normal agents are near constant. The μ values for corrupt agents show that, at the end of the simulation, their average performance is worse than those of normal agents, due to their constant violation of norms. A high σ value for these agents shows that their performance suffers between simulations. We can then see that through the analysis of recorded agent actions and successful identification of violation occurrences, violating agents are punished by the enforcement system and have their utilities affected.

Table 1. Agent Utilities × Monitoring Intensity.

Intensity	*officer1*	*officer2*	*corrupt officer1*	*corrupt officer2*
0	65,3285	66,3714	130,6571	130,7000
10	64,5871	66,5714	103,3000	106,2285
20	65,4428	65,0142	86,8000	87,9571
30	65,3142	64,8714	73,7571	75,6571
40	65,7857	65,1857	59,0571	57,8142
50	65,6714	65,7714	54,3285	53,1857
60	65,1571	65,1714	38,7714	38,4571
70	65,0142	65,6571	27,6428	27,3714
80	64,7857	64,9571	19,2285	19,3428
90	65,0714	66,1714	13,7857	13,8142
100	66,7571	65,8000	1,4714	0,0285
μ	65.3559	65.5948	55,3454	55,5051
σ	0.5569	0.5705	38,4836	39,1996

6 Related Work

There are multiple tools available for programming multi-agent environments, few of which provide mechanisms for norm specification. These tools range from programming libraries to model-based simulators. To name a few, NetLogo [16] and its distributed version HubNet [25] are of the model-based type and allow users to work with educational projects and, to some extent, professional ones. Other tools include MASON [17] and Repast [18]. MASON is a simulation library developed in Java that provides functions for modelling agents and visualising simulations as they run. As for Repast, it uses interaction models much like NetLogo does, although it is meant for professional use and thus offers more alternatives for agent programming. One final example worth mentioning is MASSim [11], which promotes multi-agent research and is used in the MAS Programming Contest[2] [26]. This one, however, provides only the tools related to the contests. Although it is possible to develop custom agents for operation within the simulator, the practice is not encouraged by its developers.

Building a full-fledged norm-based behaviour simulation engine is not a trivial task, and the "Emergence in the Loop" (EMIL) [27] project built a set of tools to accomplish this objective. A toolset which includes an extension of the BDI architecture that is capable of simulating the processes referred to as "immergence" and "emergence" of norms [28]; and an integration with multi-agent modelling tools such as NetLogo [16] and Repast [18]. In this way, agents are modelled in one of these environments and then simulated using the EMIL agent architecture. It is a very powerful tool for studying social behaviour in autonomous agents, since agents can reason about norms and, together, create conventions of what kinds of behaviours must be avoided or followed. EMIL's approach to normative simulation is more focused on agents and their experience

[2] `https://multiagentcontest.org`

with norms. This contrasts with our approach in that we are more focused on norm monitoring and enforcement tasks, and little is said about these matters in the EMIL literature. We also consider the environmental aspects of Normative Multi-Agent Systems, which is why we employ CArtAgO in our implementation.

7 Conclusions and Future Work

In this paper, we constructed a mechanism of norm processing and enforcement in a multi-agent environment. We show its feasibility with an implementation using Jason [12] and Cartago [21] technologies. By keeping track of agent activities and analysing actions against a normative base, it is possible to detect violations and enforce norms through the sanctioning of violating agents. With this framework, it is possible to evaluate different implementations [6,29,30,31] of normative behaviour. Statistics collection can also be customised so that results may be compared between simulations.

CArtAgO allows us to build environments in a distributed manner, therefore providing scalability for realistic simulation scenarios or complex multi-agent systems. The philosophy behind CArtAgO, which sees the environment as the composition of artefacts through which agents interact, also aided in the framework's construction. Artefacts are modular, they can be attached or detached to a multi-agent system seamlessly. Meaning that artefacts can be created to suit an agent's or group of agents's specific needs, and agents may connect only to those artefacts that are related to their designs. We took advantage of those features to build the interfaces for the monitoring system to access the Action History and Normative Base structures.

As future work, we aim to build improvements and extensions to the framework, such as: a mechanism to be added to the normative system that allows activation and expiration of norms following predefined conditions; agent architectures that can learn from normative environments, and with that avoid penalties by violation or minimising performance loss when violations are inevitable [6]; enable agents to learn about the enforcing intensity and use that information to their advantage [24]; and the introduction of agent hierarchies to control normative power [32].

References

1. M. Wooldridge, "Intelligent Agents," in *Multi-Agent Systems* (G. Weiss, ed.), pp. 3–50, The MIT Press, 2nd ed., 2013.
2. A. J. I. Jones and M. Sergot, *Deontic Logic in Computer Science: Normative System Specification*, ch. Chapter 12: On the characterisation of law and computer systems: the normative systems perspective, pp. 275–307. Wiley Professional Computing Series, Wiley, 1993.
3. M. Kollingbaum, *Norm-governed Practical Reasoning Agents*. PhD thesis, University of Aberdeen, 2005.

4. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre, "The BOID architecture: conflicts between beliefs, obligations, intentions and desires," in *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 9–16, 2001.
5. G. Governatori and A. Rotolo, "BIO logical agents: Norms, beliefs, intentions in defeasible logic," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 1, pp. 36–69, 2008.
6. F. Meneguzzi and M. Luck, "Norm-based behaviour modification in BDI agents," in *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pp. 177–184, 2009.
7. N. Criado, *Using Norms To Control Open Multi-Agent Systems*. PhD thesis, Universitat Politécnica de València, 2012.
8. N. Alechina, M. Dastani, and B. Logan, "Programming norm-aware agents," in *Autonomous Agents and Multi-Agent Systems* (W. van der Hoek, L. Padgham, V. Conitzer, and M. Winikoff, eds.), pp. 1057–1064, IFAAMAS, 2012.
9. S. Panagiotidi, J. Vázquez-Salceda, and F. Dignum, "Reasoning over norm compliance via planning," in *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII, 14th International Workshop, Held Co-located with the International Conference on Autonomous Agents and Multi-Agent Systems, Valencia, Spain, June 5, 2012, Revised Selected Papers* (H. Aldewereld and J. S. Sichman, eds.), vol. 7756 of *Lecture Notes in Computer Science*, pp. 35–52, Springer, 2012.
10. F. Meneguzzi, S. Mehrotra, J. Tittle, J. Oh, N. Chakraborty, K. Sycara, and M. Lewis, "A cognitive architecture for emergency response," in *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems*, pp. 1161–1162, 2012.
11. T. M. Behrens, M. Dastani, J. Dix, and P. Novák, "MASSim: Multi-Agent Systems Simulation Platform," *In Begehung des Simulationswissenschaftlichen Zentrums*, 2008.
12. R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
13. F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
14. R. Conte, G. Andrighetto, and M. Campennl, *Minding Norms: Mechanisms and Dynamics of Social Order in Agent Societies*. Oxford Series on Cognitive Models and Architectures, OUP USA, 2013.
15. M. Fagundes, S. Ossowski, and F. Meneguzzi, "Analyzing the tradeoff between efficiency and cost of norm enforcement in stochastic environments populated with self-interested agents," in *Proceedings of the 21st European Conference on Artificial Intelligence*, 2014.
16. U. Wilensky, "NetLogo." Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, 1999. http://ccl.northwestern.edu/netlogo/.
17. S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, "MASON: A Multiagent Simulation Environment," *Simulation*, vol. 81, pp. 517–527, July 2005.
18. M. North, N. Collier, J. Ozik, E. Tatara, C. Macal, M. Bragen, and P. Sydelko, "Complex adaptive systems modeling with repast simphony," *Complex Adaptive Systems Modeling*, vol. 1, no. 1, 2013.
19. J. F. Hubner, J. S. Sichman, and O. Boissier, "Developing Organised Multiagent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels," *Int. J. Agent-Oriented Softw. Eng.*, vol. 1, pp. 370–395, Dec. 2007.

20. A. J. I. Jones and M. Sergot, "On the characterisation of law and computer systems: The normative systems perspective," in *Deontic Logic in Computer Science: Normative System Specification*, pp. 275–307, John Wiley and Sons, 1993.
21. A. Ricci, M. Viroli, and A. Omicini, "CArtAgO: A Framework for Prototyping Artifact-based Environments in MAS," in *Proceedings of the 3rd International Conference on Environments for Multi-agent Systems III*, (Berlin, Heidelberg), pp. 67–86, Springer-Verlag, 2006.
22. A. S. Rao, "AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language," in *Agents Breaking Away, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996, Proceedings* (W. V. de Velde and J. W. Perram, eds.), vol. 1038 of *Lecture Notes in Computer Science*, pp. 42–55, Springer, 1996.
23. M. E. Bratman, *Intention, Plans and Practical Reason*. Harvard University Press, 1987.
24. F. Meneguzzi, B. Logan, and M. S. Fagundes, "Norm monitoring with asymmetric information," in Bazzan *et al.* [33], pp. 1523–1524.
25. U. Wilensky and W. Stroup, "HubNet." Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., 1999. http://ccl.northwestern.edu/netlogo/hubnet.html.
26. T. M. Behrens, M. Dastani, J. Dix, J. Hübner, M. Köster, P. Novák, and F. Schlesinger, "The Multi-Agent Programming Contest," *AI Magazine*, vol. 33, no. 4, pp. 111–113, 2012.
27. G. Andrighetto, R. Conte, P. Turrini, and M. Paolucci, "Emergence In the Loop: Simulating the two way dynamics of norm innovation," in *Normative Multi-agent Systems, 18.03. - 23.03.2007* (G. Boella, L. W. N. van der Torre, and H. Verhagen, eds.), vol. 07122 of *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum für Informatik, Schloss Dagstuhl, Germany, 2007.
28. G. Andrighetto, M. Campennì, R. Conte, and M. Paolucci, "On the Immergence of Norms: a Normative Agent Architecture," in *Proceedings of the Association for the Advancement of Artificial Intelligence Symposium, Social and Organizational Aspects of Intelligence*, Forthcoming, 2007.
29. J. Lee, J. Padget, B. Logan, D. Dybalova, and N. Alechina, "Run-time norm compliance in BDI agents," in Bazzan *et al.* [33], pp. 1581–1582.
30. W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman, "Normative conflict resolution in multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 2, pp. 124–152, 2009.
31. N. Criado, E. Argente, V. J. Botti, and P. Noriega, "Reasoning about norm compliance," in *10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan, May 2-6, 2011, Volume 1-3* (L. Sonenberg, P. Stone, K. Tumer, and P. Yolum, eds.), pp. 1191–1192, IFAAMAS, 2011.
32. N. Oren, M. Luck, and S. Miles, "A model of normative power," in *9th International Conference on Autonomous Agents and Multiagent Systems, Toronto, Canada, May 10-14, 2010, Volume 1-3* (W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, eds.), pp. 815–822, IFAAMAS, 2010.
33. A. L. C. Bazzan, M. N. Huhns, A. Lomuscio, and P. Scerri, eds., *International conference on Autonomous Agents and Multi-Agent Systems, Paris, France, May 5-9, 2014*, IFAAMAS/ACM, 2014.

Exploring the Effectiveness of Agent Organizations

Daniel D. Corkill, Daniel Garant, and Victor R. Lesser

University of Massachusetts Amherst, Amherst, MA 01003, USA

Abstract. Organization is an important mechanism for improving performance in complex multiagent systems. Yet, little consideration has been given to the performance gain that organization can provide across a broad range of conditions. Intuitively, when agents are mostly idle, organization offers little benefit. In such settings, almost any organization—appropriate, inappropriate, or absent—leads to agents accomplishing the needed work. Conversely, when every agent is severely overloaded, no choice of agent activities achieves system objectives. Only as the overall workload approaches the limit of agents' capabilities is effective organization crucial to success.

We explored this organizational "sweet spot" intuition by examining the effectiveness of two previously published implementations of organized software agents when they are operated under a wide range of conditions: 1) call-center agents extinguishing RoboCup Rescue fires and 2) agents learning network task-distribution policies that optimize service time. In both cases, organizational effect diminished significantly outside the sweet spot. Detailed measures taken of coordination and cooperation amounts, lost work opportunities, and exceeded span-of-control limits account for this behavior. Such measures can be used to assess the potential benefit of organization in a specific setting and whether the organization design must be a highly effective one.

1 Introduction

Organization is an important mechanism for improving performance in complex multiagent systems [1–6]. Designed agent organizations provide agents with organizational directives that, when followed, reduce the complexity and uncertainty of each agent's activity decisions, lower the cost of distributed resource allocation and agent coordination, help limit inappropriate agent behavior, and reduce unnecessary communication and agent activities [7–9].

When agents are mostly idle, agents can acomplish needed work whether or not they are well organized. This does not mean that effective organization does not affect how efficiently the agents work together, only that unorganized and even misorganized agents have sufficient time and resources to accomplish system objectives when lightly loaded. Conversely, when every agent is severely overloaded, no choice of agent activities achieves system objectives. In this situation, effective organization can help agents be more efficient while failing to

achieve objectives fully, but whether they are well organized or not, the system is unable to perform acceptably. Only as the overall workload approaches the limit of agents' capabilities does organization play a significant role in system performance.

2 Organizational "Sweet Spot"

We first explored this organizational-impact conjecture empirically using an previously implemented and described system of organizationally adept BDI[1] agents [11–13] operating in a well-instrumented and highly parametrized experimental platform adapted from the fire-extinguishing portion of RoboCup Rescue [14]. Organizationally adept *call center* agents direct *fire brigade* resources under their control to extinguish fires in important buildings as quickly as possible. There are no fire-brigade bases in the adapted RoboCup Rescue environment, and brigades typically move directly from fire to fire, remaining deployed if they become briefly idle. The objective is to minimize the total importance-weighted damage to buildings. A call center can use its fire brigades to execute plans to achieve its own goals of extinguishing building fires, and it can request temporary use of fire brigades from other call centers when necessary.

Our goal was to learn how the relative performance of previously evaluated agent organizations in this multiagent system changed when operating in environments well outside the conditions typically studied. Whether the existing agents and organization designs in this system were the best possible was not a concern, as better candidates would affect only the magnitude of the relative performances and not their qualitative characteristics. Some observations were intuitive, but there were also surprises, and we believe this to be the first systematic study of organizational impact in a multiagent system over such a broad range of conditions. We ran and analyzed thousands of controlled and repeatable simulation experiments involving dynamic environments in which new fires occur at various city locations throughout the entire duration of an experimental scenario. In such settings, call-center agents have an ongoing (but potentially changing) firefighting workload in which following organizational guidance offers potential advantages over unguided, reactive local decision-making.

Observation 1: Sweet-spot behavior ⇒ Figure 1 shows, as the firefighting workload increases, the performance benefit provided by call center agents that have been given an effective organization design that specifies a responsibility region for each call center (Org) relative to call-center agents operating without any responsibility-region directives (No Org). Call centers give priority to fighting fires in their responsibility regions when such regions are provided. Each of the four call centers controlled six fire-brigade resources. Performance attained in each of the 320 simulation runs is a raw score of the inverse importance-weighted fire damage in the city. We observed that the performance benefit achieved by organization (the raw score improvement) was greatest when the average

[1] Belief-desire-intention model of agency [10]

firefighting workload on brigades was near their capacity to fight important fires (approximately 2.2 fires per timestep). All figures illustrate trends as workload (e.g., ignition frequency) is varied. Trend lines are fit using a local linear model, with shaded regions representing a 95% confidence level in the fit. For example, each trend line in the firefighting experiments fits 320 separate simulation runs (drawn as individual dots).

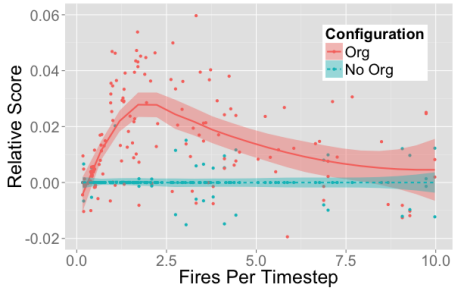


Fig. 1. Relative Score Achieved by Organization

Attenuation of organization benefit outside the sweet spot is a form of phase transition behavior. The transition occurs as the workload approaches the limit of agents' capabilities. The effect of phase boundaries has proved important in satisfiability problems [15–17] as well as to understanding problem difficulty in constraint satisfaction, number partitioning, and traveling salesmen tasks.[2] With multiagent organizations, it is important to determine where on the control complexity scale a system is operating (how important using an effective organization is to system performance) and more generally, when complex multiagent systems are operating within their organizational sweet spot. One may argue that organizations (multiagent or otherwise) will tend to be inevitably operated within the sweet spot region due to real-world economics that limit capabilities and resources to the minimum required to operate effectively.

Upon observing organizational sweet-spot behavior, we took a more detailed look into what was occurring as workload changed that accounted for the benefit attenuation.

3 Performance Factors

Why do we create agent organizations? One reason is that complex agent behavior becomes more structured and understandable through the definition of roles, behavioral expectations, and authority relationships [18]. Additionally, organizational concepts can be used to help design and build agent based systems

[2] For example, a typical phase-transition performance plot, such as Figure 4 in the classic Kirkpatrick and Selman SAT phase-change paper [16] shows the performance cliff that occurs at the phase boundary, which shifts laterally under different conditions. If such a figure is redrawn as relative difference curves from a baseline condition (such as the k=6/N=40 values in that figure), it reveals wide "sweet spot" curves similar to the curves shown in this paper. Relative plots highlight the span and magnitude of performance differences near the phase change, and we consider them more informative in highlighting sweet-spot regions than raw performance-value plots.

(organization-based multiagent system engineering). There is also a line of research that addresses *organizational membership* in open agent societies (incentives for organizational recruitment and retention and for the replacement of agents that leave the organization). Recent work in open and sociotechnical settings [19, 20] has this emphasis. Aligning agents' individual goals and objectives with those of the organization are among the issues addressed in that context. Our focus here is on *organizational control*; specifically, the organizational performance of the members ("how they do their jobs"), rather than on attracting participants from an open pool of agents ("obtaining members for the enterprise") or designing the agent system ("defining what the jobs (roles) are"). We assume here that we have acquired the agents we need, that they all share the organizational objectives (e.g., saving the most important buildings in the city), and that they are competent in their ability to perform tasks necessary to attain that objective. For example, there is no need to decide if an agent is able to play some role in the organization [21]. Furthermore, there are no non-cooperative agents trying to burn things down. Nevertheless, the cooperative agents sometimes do work at cross-purposes in attaining those objectives (such as all wanting to fight an important fire). This can occur whether the agents are organized or not, because agents have a limited local view of the situation. If unorganized agents did not have the same shared objective as when organized, then some performance gained through organization could stem from the changed objectives. Our assumptions eliminate such a cooperative-objective bonus.

We distinguish between *operational decision making*, the detailed moment-to-moment behavior decisions made by agents, and *organizational control*, an organization design expressed to agents through directives ("job descriptions") that limit and inform the range of operational decisions made by each agent in the organization. These directives contain general, long-term guidelines, in the form of parametrized role assignments and priorities (e.g., prefer extinguishing fires in region A over fires in region B), that are subject to ongoing elaboration into precise, moment-to-moment activity decisions by the agents [22, 2, 4]. Ideally, following organizational directives should be beneficial when agent directives can be designed that perform well over a range of potential long-term environment and agent characteristics.

3.1 Operational challenges

Without organizational directives, a call center must coordinate with other centers to avoid sending redundant fire brigades to the same fire (every call center receives all fire reports) using a highest estimated utility protocol to resolve conflicts. Coordination and *retractions* consume valuable time, delaying extinguishing operations. The designed organization only requires coordination if a call center wants to fight a fire outside its responsibility region. When region responsibilities are inappropriate and do not match workloads, fire-brigade *borrowing* requests from overloaded centers increase, again with a loss in performance. When the design is appropriate, retractions are diminished at the risk of more borrowing (as we will demonstrate when we discuss Figure 6) . Call centers must

consider all borrowing and loaning options in the context of estimated opportunity costs that are based on potential new fires and uncertainty in the duration of fighting current fires. These are challenging decisions even when agents are well organized.

The call-center agents are highly competent and can make skillful operational decisions to extinguish fires without organizational guidance. Norms, functions, protocols, etc., are implicitly represented in the plan templates used by these call-center agents. Centers follow these norms (organized or not) and know how to work together to fight fires and share fire-brigade resources.

Appropriately organized call-center agents, when operating in the sweet spot, should function better than unorganized centers, which must consider of all potential activities and explicitly coordinate them. The organizational complexity in the firefighting system is quite simple. Each call center can perform only two roles: 1) extinguishing fires by directing fire brigades to fight them and 2) loaning fire brigades to another call center. Perhaps counter intuitively, organizational design and control of split roles in homogeneous multiagent systems is more challenging than assigning discrete functional roles to specialized agents in heterogeneous multiagent systems because specialization reduces the space of reasonable choices [8]. The organizational "simplicity" in the firefighting setting means that observed organizational performance differences stem from a relatively small set of organizationally-biased behaviors and are not obscured by complex role and agent interactions.

3.2 Factors affecting organizational performance

We analyzed a number of general factors that influence organizational performance. As these factors change, a designed organization may become highly effective or less effective. In the discussion that follows, we provide an intuitive description of each factor, why it is important, and how it can affect organizational performance. We adjusted each factor individually while holding other environmental settings constant in order to observe its effect on organizational performance independent of the other factors. In total, we conducted a broad analysis that included over 5000 simulation runs with over ten terabytes of simulator output to determine how the general factors of coordination requirements, cooperation benefits, lost opportunity, workload imbalance, and span of control impact the effectiveness of organization. We begin with coordination.

Coordination Requirements Typically, complex tasks performed by multiple agents require coordination, and often a well-coordinated system will perform much better than a system where agents work at cross purposes from only their local, selfish perspectives. In firefighting, coordination is necessary to ensure that call-center agents share responsibility for extinguishing a building only when necessary, and otherwise fight important fires independently (i.e., they do not blindly work on the same fire when more utility could be gained by working on separate fires).

Coordination is not without associated costs, often involving delays while beliefs, desires, and intentions are communicated. The time required for agents to communicate this information and reconcile it with information from other agents can be significant, especially in cases where agents control resources which must be held in reserve while an agent decides whether it wishes to pursue some goal. Even more significantly, when agents take uncoordinated actions that involve operating in the world, they must deal with the consequences of physically moving resources and then withdrawing them (or having wasted them if they are consumables) once they discover their actions are in conflict with those of another agent. In our analyses, this has been the largest contributor to coordination "cost."

The amount of coordination required is not organization-independent. Organizational directives influence agents to assume specific roles and responsibilities pertaining to certain goals, and assume less responsibility for other goals. The best-case organization for a specific situation would be a perfect partitioning of responsibility regions so that agents select the fires for which they are responsible over those that are the responsibility of others. This ideal situation results in minimal *goal conflicts*, where two agents needlessly pursue the same goal (e.g., extinguish the building at 5^{th} and Madison). It is important to note that even this organization is not coordination-free, but when each goal is managed and committed to by the agent with the highest expected utility, the committing agent is best suited for reaching out for assistance if necessary. In the context of firefighting, this assistance comes in the form of lending and borrowing fire brigades, an effective remedy for temporal workload imbalances. However, as we will note shortly, excessive resource borrowing leads to inefficiencies in resource provisioning and is often a sign of a more permanent resource imbalance. The worst-case organization (in terms of coordination complexity) would influence every agent to select the same goals (No Org configuration). We analyzed many organization configurations to explore the full spectrum between these two extremes, where organization sometimes cannot prevent agents from selecting the same goals, and at other times, is effective in preventing a goal conflict (which we will also discuss later in conjunction with Figure 6).

This coordination phenomena occurs in firefighting because call centers need to negotiate with other call centers about which fires to fight. In order to come to a resolution for a contested goal, call centers need to compute and share their expected utility with peers. The call center with the highest expected utility will then be responsible for managing fighting the fire, and for borrowing fire-brigade resources from peers if necessary. To investigate the effect of adjusting this coordination cost, we adjusted the *resolution period*, during which call centers reserve resources to fight a fire while waiting for and considering bids from other call centers intent on fighting the same fire. Only after the resolution period has elapsed will the call center with the highest utility commit to fighting the fire. By increasing the resolution period, we increase the cost of coordination while simultaneously making centers more "globally aware" of the utility expected by other agents. By lowering the resolution period, we lower the cost of coordination

but make call centers more selfish in that they are less open to considering bids from other centers. Figures 2 and 3, to be discussed shortly, show the effects of "Low-Cost" (short) resolution and "High-Cost" (long) resolution times.

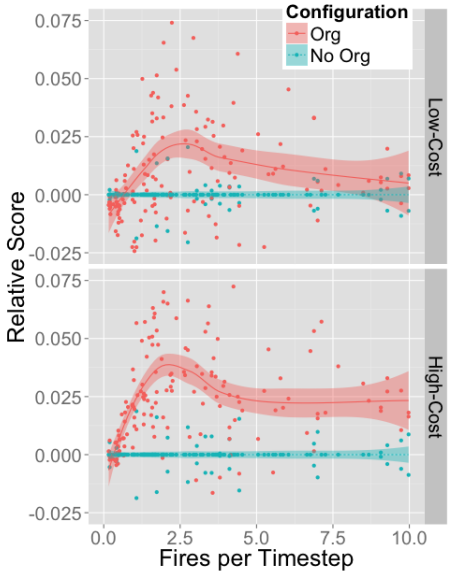


Fig. 2. Varying Coordination Requirements: Score Relative to No Org

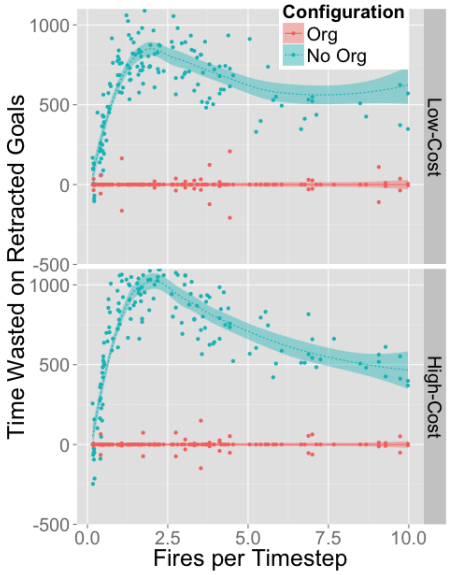


Fig. 3. Varying Coordination Requirements: Cost Effects Relative to No Org

Observation 2: The performance separation of effective organization increases with coordination requirements, without shifting the sweet spot laterally ⇒ We analyzed several organizational designs: 1) a specific responsibility region for each call center (Org) and 2) all centers are responsibible for the entire city (No Org). It seems reasonable to believe that when fires are uniformly distributed, Org would perform best, minimizing goal conflicts while still providing each agent with sufficient beneficial opportunities in its responsibility region. In practice, this is generally true, however, we have found that in cases where, when the conflict resolution period is very short (corresponding to low coordination cost and more selfish agents), the directives supplied to the organized agents do not improve on the No Org baseline. As coordination cost grows, the performance of the organized agents (which need to coordinate less frequently) improves increasingly on the No Org configuration (see Figures 2 and 3). Figure 3 shows the total retraction time relative to No-Org, which has the most retractions. In both Figures 2 and 3, the 0- and 10-time-steps resolution period results are relative to comparable 0- and 10-time-steps resolution No Org baselines.

Note that with low coordination cost (0-timestep resolution), the difference in performance between the Org and the No Org configuration is only statistically significant within a small window, centered at about 2–2.5 fires per timestep. Correspondingly, the scenario with high coordination cost (10-timestep resolution) achieves a prominent global maximum centered at this time window. From this analysis, it can be seen that when coordination does not incur significant costs, organization is not nearly as beneficial as in cases where coordination (or the absence of needed coordination) is costly. At moderate workload levels, the performance gains afforded by organization reach the maximum. When the simplicity of the scenario does not require coordination, the performance of the Org configuration and the No Org configuration are statistically indistinguishable. Extremely overloaded work scenarios are marked by either statistically indistinguishable performance differences or diminished returns.

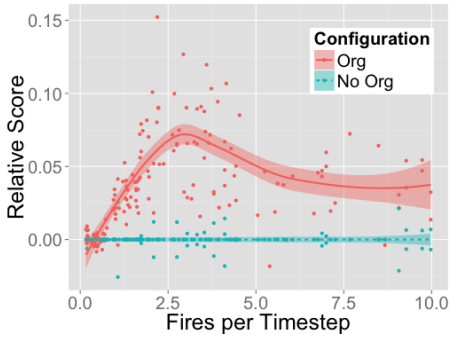


Fig. 4. Relative Score with Twice as many Fire Brigades

Observation 3: Increasing call-center capabilities by adding resources results in a lateral shift and widening of the sweet spot ⇒ The width and position of the sweet-spot window is not fixed, as it depends on the agent's capabilities in servicing goals at either end of the workload range. Call centers become more capable when they have more fire-brigade resources. Figure 4 shows the result of doubling the number of fire brigades controlled by each call center from six to twelve. Now, the organizational sweet spot occurs at a higher workload level: at approximately 2.7 fires per timestep. In addition, the sweet spot is wider as call centers can handle greater task loads before the situation becomes hopeless.

By holding the conflict resolution period constant and varying the number of call centers in the system, we see that coordination complexity is also a function of how "well partitioned" the centers' responsibilities are. In experiments with four call centers, we can see that fewer goal conflicts arise in the Org case than the No Org case. However, if we increase the number of call-center agents to twelve, each with two rather than six fire brigades and responsibility regions that overlap with two other centers, the environmental responsibilities are too precisely partitioned to handle temporal responsibility differences even if, on average over the course of the run, each center's responsibilities are roughly uniform. In Figures 5 and 6, this behavior is reflected in the fact that the number of goal conflicts in the organized, 12-call-center configuration approach the number of conflicts without organization. Correspondingly, the differences in performance between the two configurations are significant. Any advantages to organization under the

4-call-center scenario are lost with the increase in coordination complexity in the 12-call-center scenario. This observation is consistent with the notion that there is an "ideal" number of call centers given the centers' capabilities and the environmental conditions. We do not know for certain that a 4-center organization is the best choice for the environmental conditions that we simulated, but it is certainly better than a 12-center organization, as the 4-center organization provides a better balance between the partitioning of responsibility regions and coordination complexity [23].

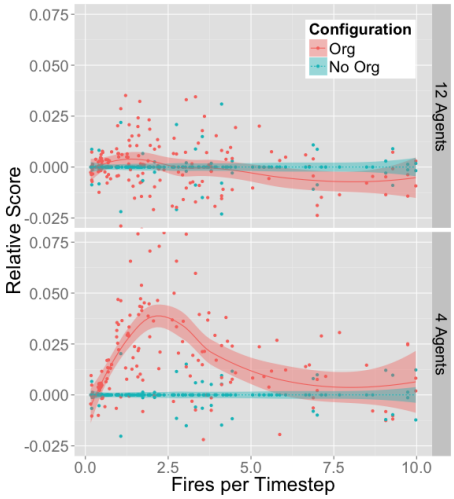


Fig. 5. Varying the Number of Call Centers: Relative Score

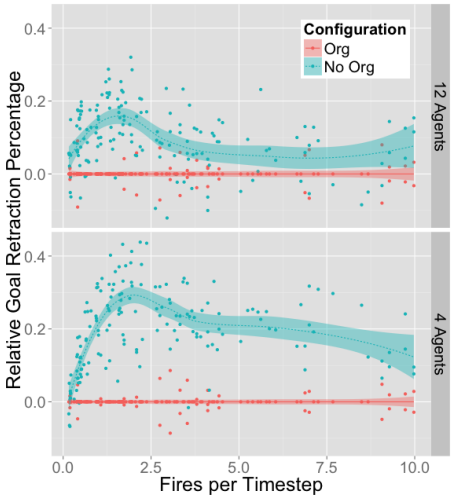


Fig. 6. Varying the Number of Call Centers: Relative Goal-Conflict Rate

Workload Imbalance Organizational directives influence agents to assume responsibility over particular goals and tasks. This reduces the amount of coordination involved in meeting these demands, as there is some expectation of which agent will perform or manage a task. In order for this organizational influence to improve performance, the per-agent workload that is suggested by the organizational directives must be consistent with the distribution of tasks in the environment. Otherwise, some agents have too little work and others have too much. As such, highly beneficial tasks may go without consideration by underloaded agents while overloaded agents struggle to complete all of the tasks they are responsible for. Workload imbalance occurs in firefighting when the distribution of fires throughout the city is not consistent with the size of each of the centers' responsibility region. For instance, if 60% of fires occur in the northwest corner of the city, a partitioning of the city into four equally-sized quadrants would result in a significant average workload imbalance, with the call center in

Fig. 7. Varying Workload Balance: Mean Goal Benefit

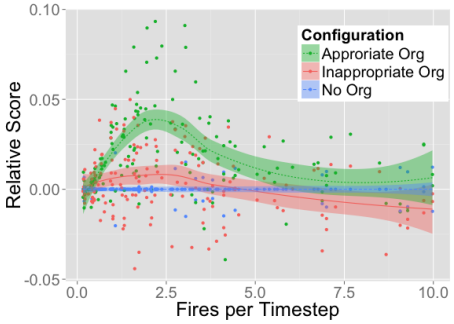


Fig. 8. Varying Workload Balance: Relative Score

the northwest corner of the city having almost six times the workload of other centers. In this setting, an appropriate organization would assign a much smaller responsibility region to the call center responsible for the northwest corner of the city, and expand the responsibilities of other call centers to make up the difference in coverage.

Observation 4: The performance separation of effective organization increases with increased workload imbalance ⇒ When workloads are imbalanced in this way, call-center agents are not necessarily idle, but instead they work on less beneficial goals. Thus, the penalty occurred by providing these call centers with an inappropriate organization comes in the form of "lost opportunity," where the agent could have performed much more beneficial tasks if it had not been discouraged from doing so by organizational directives. Correspondingly, Figure 7 shows that, as the organizational influences becomes less appropriate, the mean benefit of selected goals becomes lower. A surprising observation shown in Figure 7 is that the No Org case has the highest mean goal benefit of all of the configurations (but not the highest relative score). This is due to No Org agents' preference to selflishly commit to attractive goals which other agents may already be working on, introducing additional goal conflicts and coordination cost.

Observation 5: Extreme workload imbalance, high or low, causes organizationally guided performance to converge to non-organized performance ⇒ On the other end of the spectrum, both Appropriate and Inappropriate Org's less beneficial goals result in a direct lowering of overall score. Figure 8 indicates that this behavior essentially lowers the Appropriate Org curve onto the No Org curve, while still maintaining a window in the workload spectrum where organization is especially advantageous.

Span of Control An important factor in determining if and how agents should be organized is span of control. Simply adding resources (or performers) to a task does not result in constant gain per added resource, and can even result in a net

loss of utility. This phenomena is found in many real-world settings [23] where organizations attempt to scale the number of performers without correspondingly scaling management capacity (e.g., hundreds of construction workers cannot be managed by a single foreman). In the firefighting simulator, per-resource effectiveness is diminished above a parameterized call center span-of-control limit.

Observation 6: Increasing the number of call-center agents beyond what is necessary given their span-of-control capabilities adds coordination requirements (to keep them out of each other's way), decreasing the organizational benefit separation compared to a suitable number of centers ⇒ Span-of-control limits are both important and ubiquitous, since centralization is not generally tractable or realistic. When exceeded in RoboCup Rescue firefighting, performance per brigade is attenuated, counteracting coordination reductions from centralization. Otherwise, one center could handle all brigades.

We explored span of control using a configuration where a single call center agent is responsible for managing all 24 fire-brigade resources in the system, but with a span-of-control limit imposed after 6 utilized brigades. Then, we increased the span-of-control capability of the center to 24 (no span-of-control-limit attenuation) to understand how the single call-center agent would perform with no span-of-control limit. We compared these two cases with the baseline configuration where the fire brigades are distributed evenly across four call centers, each controlling 6 of them. Because no call center coordination is needed when there is a single center, in cases where fewer than 6 brigades are needed to execute all of the tasks in the environment, both of the single-agent configurations outperform the multiagent configuration (Figure 9).

At a workload level of one fire per timestep, the limited resource effectiveness incurred by the span-of-control penalty becomes more significant than the coordination cost in the multiagent case. Further, since the single-agent case incurs no coordination complexity, there is a noticeable peak in the single-agent configuration without a span-of-control penalty, corresponding to the coordination-cost peak discussed previously.

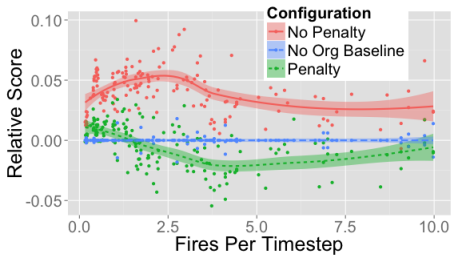


Fig. 9. Span of Control Analysis

Observation 7: Coordination requirements that exceed an agent's span-of-control capabilities leads to an inverted performance curve ⇒ Figure 9 shows that the sweet spot obtained when running under the best case scenario of a single call center with no coordination requirements becomes a "sour spot" when span of control is considered. Intuitively, the sweet spot drops below the No Org baseline in the region of the workload spectrum where it is important that fire-brigade resources

are managed effectively. With span-of-control limits imposed, fire-brigade effectiveness is diminished.

4 MARL Organizations

We next looked for sweet-spot behavior using another previously described and implemented system involving agent organizations. This second system operates in a very different setting: organizing agents that are learning task-assignment policies that optimize service time for tasks arriving in a network [24, 25]. Tasks arrive according to a Poisson distribution, and have variable difficulty (measured as time units) governed by an exponential distribution. Every task is spawned at some vertex v, augmenting agent v's *routing queue* with the new task. Agent v can then decide whether to work on a task locally, adding that task to v's *work queue*, or to forward the task to one of v's neighbors. At every timestep, task at the head of v's work queue is decremented, indicating that it is one timestep closer to being completed. Once the number of remaining timesteps has reached 0, the task is removed from the queue and agent v may proceed to complete the next task in the work queue. Agents receive the inverse of task service time as a reward when a task is completed. To operate effectively in this setting, agents must construct estimates of task service time given locally observable state information such as the size of neighbor's work queues and historical completion times when forwarding tasks.

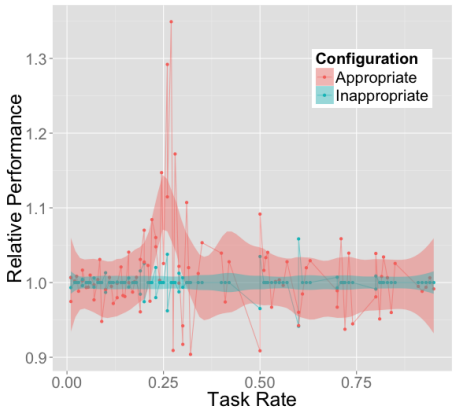


Fig. 10. Relative Performance of MARL Organizations

In this domain, each agent is either a subordinate or a supervisor. Supervisors are responsible for transferring experiences between subordinates that are experiencing similar environmental conditions. Appropriate organizations in this task allocation domain are those that arrange supervisors in a way that exploits similarities between agents. If a group of subordinates frequently experience the same environmental conditions, a great deal of transfer learning can take place. If subordinate groups experience vastly different environmental conditions, transfer learning can occur less frequently, thus not taking advantage of the benefits that organization provides. As in firefighting, an organizational arrangement of supervisors that is appropriate given a particular task distribution may be inappropriate under a different task distribution, so the organization is only ef-

fective if the actual distribution is consistent with the expectations assumed in the designed supervisor arrangement.

We used a completely different agent implementation and environment simulator in exploring sweet-spot behavior in multiagent reinforcement learning (MARL) organizations. For our experiments, we used a 100-agent lattice network and considered two agent organizations. The first organization arranges 4 supervisors such that agents are assigned to supervisors based on their distance from the border of the lattice. The second organization arranges 4 supervisors according to quadrants of the lattice. Tasks are then distributed on the lattice originating from the boundary. Under this model, the former organization is considered "Appropriate" since it partitions agents in a manner that maximizes the similarity of agents in supervisory groups. The latter organization is considered "Inappropriate" since it arranges agents in a way that prohibits effective experience sharing. Given this setup, we experimentally varied the difficulty of the learning problem by increasing the mean of the Poisson distribution governing task distribution on the range [0, 1], where 1 represents a very heavy task load (averaging one task per time unit). One hundred values of λ were sampled uniformly along this range for each supervisory configuration, resulting in a total of 200 runs. Evaluation was performed in terms of area under a learning curve (AUC), modeled as an exponential moving average of system-wide task service time. When the system converges more quickly to an optimal policy, the area under this curve will be smaller. To characterize relative performance differences across a wide array of problem difficulties, AUC was normalized relative to the Inappropriate Org configuration.

Observation 8: The MARL system also has a sweet spot $\Rightarrow$ Figure 10 shows more performance variability than occurred with firefighting, but a statistically significant sweet spot arises around a per-agent task rate of 0.25 tasks per timestep. At this workload, the Appropriate Org's performance dominates the Inappropriate Org's. Elsewhere, the two are statistically indistinguishable. The results in the MARL domain are particularly clear. When tasks arrive so frequently that agents cannot compute meaningful policies and the learning process diverges, a supervisor structure that is highly effective in the sweet spot does not help in transferring reasonable policies. On the opposite end of the workload spectrum, when tasks arrive so infrequently that agents do not need to act intelligently in order to service the requests in a timely manner, policy transfer is not important. It is clear from this analysis that even with a completely different set of system dynamics and agent behaviors, an organizational sweet spot exists.

5 Closing Thoughts

Although we have measured and analyzed agent-organization performance under widely varying conditions using only two previously implemented and studied systems (each operating in a different problem domain), we believe that the qualitative behaviors we observed are general and apply to multiagent organizations in *any* domain. We hope our observations encourage those working with

more complex heterogeneous agent organizations to investigate and report their performance over a wider range of conditions. Recognizing when a multiagent system will be operating in its organizational sweet spot is helpful in deciding how much effort should be spent in designing and using an agent organization as well as for explaining situations where using an agent organization results in little observed benefit (because the system is operating outside the sweet spot). We have observed that coordination and cooperation amounts, lost work opportunities, and span-of-control capabilities all contribute to sweet-spot performance benefits.

Understanding a multiagent system's organizational sweet spot is important, not just for understanding organizational control opportunity and effectiveness, but when considering if organizational adaptation might be worthwhile [26–28, 12]. Sweet-spot understanding is also important in open, sociotechnical settings when designing an organization (and sizing that design appropriately) for agent recruitment. Identifying where a multiagent system is operating in relation to its organizational sweet spot is important to any discussion or analysis of organizational suitability, performance, or effectiveness.

Acknowledgment This material is based in part upon work supported by the National Science Foundation under Awards No. IIS-0964590 and IIS-1116078. 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. M. S. Fox. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1):70–80, Jan. 1981.
2. D. D. Corkill and V. R. Lesser. The use of meta-level control for coordination in a distributed problem-solving network. In *IJCAI-83*, pages 748–756, Karlsruhe, Federal Republic of Germany, Aug. 1983.
3. L. Gasser and T. Ishida. A dynamic organizational architecture for adaptive problem solving. In *AAAI-91*, pages 185–190, Anaheim, California, July 1991.
4. E. H. Durfee and Y. pa So. The effects of runtime coordination strategies within static organizations. In *IJCAI-97*, pages 612–618, Nagoya, Japan, Aug. 1997.
5. K. M. Carley and L. Gasser. Computational organization theory. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 7, pages 299–330. MIT Press, 1999.
6. B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, 19(4):281–316, Dec. 2004.
7. 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.
8. 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.

9. J. Slight and E. H. Durfee. Organizational design principles and techniques for decision-theoretic agents. In *Proceedings of the Twelveth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2013), pages 463–470, May 2013.
10. 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.
11. D. D. Corkill, E. Durfee, V. R. Lesser, H. Zafar, and C. Zhang. Organizationally adept agents. In *Proceedings of the 12th International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems* (COIN@AAMAS 2011), pages 15–30, May 2011.
12. D. Corkill, C. Zhang, B. D. Silva, Y. Kim, X. Zhang, and V. Lesser. Using annotated guidelines to influence the behavior of organizationally adept agents. In *Proceedings of the 14th International Workshop on Coordination, Organization, Institutions and Norms in Agent Systems* (COIN@AAMAS 2012), pages 46–60, June 2012.
13. D. Corkill, C. Zhang, B. D. Silva, Y. Kim, D. Garant, V. Lesser, and X. Zhang. Biasing the behavior of organizationally adept agents (extended abstract). In *Proceedings of the Twelveth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2013), pages 1309–1310, May 2013.
14. H. Kitano and S. Tadokoro. RoboCup-Rescue: A grand challenge for multi-agent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.
15. P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *IJCAI-91*, pages 331–337, Sydney, Australia, Aug. 1991.
16. S. Kirkpatrick and B. Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264:1297–1301, May 1994.
17. R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, July 1999.
18. L. Gasser. An overview of DAI. In *Distributed Artificial Intelligence: Theory and Praxis*, pages 9–30. Springer, 1993.
19. L. Sterling and K. Taveter. *The Art of Agent-Oriented Modeling.* MIT Press, 2009.
20. O. Boissier and M. B. van Riemsdijk. Organizational reasoning agents. In S. Ossowski, editor, *Agreement Technolgies*, volume 8, chapter 19, pages 309–320. Springer, 2013.
21. M. B. van Riemsdijk, V. Dignum, C. Jonker, and H. Aldewereld. Programming role enactment through reflection. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (WI-IAT'11), pages 133–140, Lyon, France, Aug. 2011.
22. J. G. March and H. A. Simon. *Organizations.* John Wiley & Sons, 1958.
23. B. Horling and V. Lesser. Analyzing, modeling and predicting organizational effects in a distributed sensor network. *Journal of the Brazilian Computer Society,* special issue on Agent Organizations, 11(1):9–30, July 2005.
24. C. Zhang, S. Abdallah, and V. Lesser. Integrating organizational control into multi-agent learning. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2009), volume 2, pages 757–764, Budapest, Hungary, May 2009.
25. D. Garant, B. C. da Silva, V. Lesser, and C. Zhang. Accelerating multi-agent reinforcement learning with dynamic co-learning. Technical Report UM-CS-2015-004, School of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, Jan. 2015.

26. J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2009.
27. A. Staikopoulos, S. Saudrais, S. Clarke, J. Padget, O. Cliffe, and M. D. Vos. Mutual dynamic adaptation of models and service enactment in ALIVE. In *Proceedings of the Third International Models@Runtime Workshop*, pages 26–35, Toulouse, France, Sept. 2008.
28. T. B. Quillinan, F. Brazier, H. A. Frank, L. Penserini, and N. Wijngaards. Developing agent-based organizational models for crisis management. In *Proceedings of the Industry Track of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2009), pages 45–51, Budapest, Hungary, May 2009.

SIMPLE: a Language for the Specification of Protocols, Similar to Natural Language

Dave de Jonge and Carles Sierra

IIIA-CSIC, Bellaterra, Catalonia, Spain
{davedejonge,sierra}@iiia.csic.es

Abstract. Large and open societies of agents require regulation, and therefore many tools have been developed that enable the definition and enforcement of rules on multiagent systems. Unfortunately, most of them are designed to be used by computer scientists and are not suitable for normal users with average computer skills. Since more and more tools nowadays are running as cloud services accessible to anyone (e.g. Massive Open Online Courses and social networks), we feel there is a need for a simple tool that allows ordinary people to create rules and protocols for these kinds of environments. In this paper we present ongoing work on the development of a new programming language for the definition of protocols for multiagent systems, which is so simple that anyone should be able to use it. Although its syntax is strict, it looks very similar to natural language so that protocols written in this language can be understood directly by anyone, without having to learn the language beforehand. Moreover, we have implemented an easy to use editor that helps users writing sentences that obey the syntax rules, and we have implemented an interpreter that can parse such protocols and verify whether they are violated or not.

1 Introduction

In open multiagent systems (MAS) where any agent can enter and leave at will and the origin of the agents is unknown one needs a mechanism to regulate the behavior of those agents. Just like in human societies, rules need to be imposed in order to prevent the agents from misbehaving and abusing system resources. A good example is that of an auction taking place under a specific protocol. An English auction protocol for example, requires the buyers to make increasing bids, and stops when the auctioneer says so, after which the buyer with the highest bid wins the auction. In a Dutch auction on the other hand, bids are decreasing, and the first buyer to accept a bid wins the auction.

Many systems for the implementation of such regulatory systems have been developed, such as ANTE [6], MANET [27], S-MOISE+ [16], and EIDE [11]. They allow users to define a set of rules and then impose those rules on the agents in a MAS (the term 'agents' may here refer to software agents as well as to human beings). This enforcement of rules may happen either by punishing misbehaving agents, or by simply making it impossible to violate them, which is called *regimentation*.

One common characteristic of these systems is that they are mainly designed with computer scientists as their target users. They require knowledge of multiagent systems,

programming languages and / or formal logic. For people with no more than average computer skills they are unfortunately too complicated.

We expect however that agent technologies will become more and more common in the near future, creating a demand for simple tools to maintain and organize such systems and that can be used by ordinary people. We can compare this for example with the evolution of web development. In the early days of the Internet, developing a web page was considered an advanced task that would only be undertaken by computer experts, and hence web development languages such as HTML, PHP and SQL were developed to be used by professional programmers. However, as web pages became more and more abundant and every shop, social club, or sports team wanted to have its own web page, many tools such as DreamWeaver and WordPress were introduced to make the creation of web pages a much simpler task. We strive for a similarly easy tool for the development of multiagent systems.

A good example of where such a tool would be useful is the organization of online classes, because teachers often want to put restrictions on their students. Teachers may for example require that students only take a certain exam after they have passed all previous exams. In this way teachers make sure they do not waste their time correcting exams of students that do not study seriously anyway. Another example could be the process of organizing a conference, where one requires authors to submit before a deadline, or one requires the program chair to appoint at least 3 reviewers to each paper. Also, one can think of a tool that allows users to set up their own social networks, with their own specific rules, as suggested in [17].

Therefore, in this paper we present ongoing work on the development of a new language to define protocols for multiagent systems. This language is so close to natural language that it can be understood directly by anyone without prior knowledge of any other programming language. We call this language SIMPLE, which stands for SIMple Protocol LanguagE. Although it *looks* very similar to natural language, it has in fact a strict syntax. Together with this language we also present two tools: an editor that makes it very easy for users to write well-formed sentences, and an interpreter that parses the source file and makes sure that the rules defined in it are indeed enforced. The fact that the language comes with an editor is very important, because it enables the users to write correct protocols without having to know the rules of the language by heart. In fact, it even makes it impossible to write syntactically incorrect sentences.

We would like to stress that this language is not meant to program the agents themselves. It is only meant to program the organizational structure between the agents. That is: it puts restrictions on the agents in their actions, but does not dictate entirely what they ought to do; the agents still have the freedom to make autonomous decisions, as long as these decisions comply with the protocol. The protocol written in this language does not specify what the agents *must* do, but only specifies what the agents *can* do.

We have developed SIMPLE according to the following guidelines:

- The language should stay as close as possible to natural language.
- The syntax should remain strict: sentences must be well formed, and every well formed sentence can only have one correct interpretation.
- Given a protocol written in this language anyone should immediately be able to understand what it means, even if he or she has never seen our language before.

– Users should be able to write a protocol in this language without having to spend any time learning the language.

The only thing we require from the user is that he or she be familiar with the English language. The language as presented here is only the very first version, and we plan to extend it much further in the future.

The rest of this paper is organized as follows: in Section 2 we give a short overview of previous work done in this field. Next, in Section 3 we explain the assumptions that we have made about the set-up of any MAS to which our language is applied. In Section 4 we describe the syntax rules of our language. Next, in Section 5 we explain how our interpreter parses text files written in our language and enforces its rules upon the agents. Then, in Section 6 we give two examples of protocols written in SIMPLE, and for which we have tested that they are successfully parsed and enforced by our interpreter. And finally, in Section 7 we describe the further extensions that we are planning to add to our language.

2 Related Work

Regulatory systems have been subject of research for a long time and a number of frameworks have been implemented that often consist of tools for implementing, testing, running and visualizing protocols. Examples of such frameworks are ANTE [6], MANET [27], S-MOISE+ [16], and EIDE [11]. A comparative study of some of those systems has been made in [12].

ANTE [6] has been implemented as a JADE-based platform, including a set of agents that provide contracting services. It integrates automatic negotiation, trust & reputation and Normative Environments. Users and agents can specify their needs and indicate the contract types to be created. Norms governing specific contract types are predefined in the normative environment. Although ANTE has been targeting the domain of electronic contracting, it was conceived as a more general framework having in mind a wider range of applications.

The MANET [27] meta-model is based on the assumption that the agent environment is composed of 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. In the MANET meta-model it is assumed that the normative system can be composed of three structural components: agents, objects and spaces.

In the EIDE framework agents interact with each other in a so called *Electronic Institution*. The agents are grouped in to conversations, which are called *Scenes*. The institution has a specification that defines how agents can move from one scene to another and defines a protocol for each scene. Within a scene the agents interact by sending messages to one another. Each agent in the system has a special agent assigned to it, called its Governor, which checks whether the messages sent by the agent satisfy the protocol, and blocks them when they do not. The EIDE framework comes with a graphical tool called *Islander* [10] that allows people to create institution specifications in a visual manner. Protocols in Islander are represented as finite state machines, drawn

as a graph in which the states are the vertices and the state-transitions are the edges. Every message sent triggers a state transition.

In order to define rules and norms for multiagent systems, a vast amount of languages and logics have been proposed. It would be impossible to list all the relevant work in this field here, so we just mention some of the most important examples. A logical system to define norms and rules is called a *deontic* logic. The best known system of deontic logic is called Standard Deontic Logic (SDL) [30]. Important refinements of this logic are Dyadic Deontic Logic (DDL) [20] and Defeasible Deontic Logic [25]. Furthermore, an extension of this taking temporal considerations into account was proposed in [14]. In [22] A system to formalize norms using input/output logic was proposed, while in [15] the authors provide a model for the formalization of social law by means of Alternating-time Temporal Logic (ATL). In [19] the author proposes the use of Linear Time Logic (LTL) to express norms. Other important approaches are based on Propositional Dynamic Logic (PDL) [23], on See-to-it-that logic (STIT) [4] and on Computational Tree Logic (CTL) [5]. Models for the verification of expectations in normative systems are proposed in [8] and [1], and in [26] the authors introduce the *nC+* language for representing normative systems as state transition systems.

The above mentioned systems however mainly focus on the theoretical properties of regulatory systems. Work that is more focused on the actual implementation of such systems is for example [21] which proposes a model to define rules in the Z language, while in [3] the authors propose the use of Event Calculus for the specification of protocols. A programming language designed to program organizations, called 2OPL, was introduced in [9]. Other important examples of languages and frameworks for the implementation of norms and rules are described in: [29], [2], [28], [13], [18], and [7].

Although some of the above mentioned languages are more user friendly than others, it still seems that they all require the user to be a computer scientist or at least has some knowledge of programming, logic or mathematics. To the best of our knowledge no work has been published on the specification of protocols that aims for truly inexperienced users and tries to stay as close as possible to natural language.

There do exist a number of programming languages that claim to be similar to natural language such as hyperTalk[1] and PlainEnglish[2], but most of them still aim at real programmers, albeit that they aim for *beginning* programmers. The only exception that we know of, is a language called Inform 7 [24]. This is a language that in many cases truly reads like natural language, but the main difference with SIMPLE is that it is developed for an entirely different domain. Inform 7 is a language to write *Interactive Fiction*: an art form that lies somewhere in between literature and computer games.

We think that one of the main reasons that Inform 7 can stay very close to natural language, is that it is highly adapted to a very specific domain. This restricts the possible things a programmer may want to express and hence keeps the language manageable. We have taken a similar approach: our language is only intended to be used as a language for implementing protocols for multiagent systems, and although it could possibly be useful for other domains too, we restrict our attention to this domain.

[1] `http://en.wikipedia.org/wiki/HyperTalk`
[2] `http://www.osmosian.com`

3 Basic Ideas

We assume a multiagent system in which agents exchange messages according to some given protocol. These agents may be autonomous software agents, or may be humans, acting through a graphic user interface. The agents are however not in direct contact with one another. Every message any agent sends first passes a central server that verifies whether the message satisfies the protocol. If a message does not satisfy the protocol, then it is blocked by the server and it will not arrive at its recipients. Note that this is a form of regimentation. In this paper we will not consider any forms of punishment, and assume protocols are only enforced by means of regimentation. We assume that the life-cycle of the MAS is as follows:

1. A user (the *protocol designer*) writes a protocol in our language and stores it in a text file.
2. He or she launches a communication server, with the location of the text file as a parameter.
3. The interpreter, which is part of the server application, parses the text file.
4. Agents connect to the server through a TCP/IP connection and send messages to one another.
5. Every such message is checked by the interpreter. If it does not satisfy the protocol, it is blocked. If it does satisfy the protocol it is forwarded to its intended recipients.
6. The agent that intended to send the message is notified by the server whether the message has been delivered correctly or not.

The text file is not compiled, but is directly parsed by the interpreter, so the language is human-readable and machine-readable at the same time.

Protocols written in SIMPLE have a closed-world interpretation: every message is considered illegal by default, unless the protocol specifies that it is legal. In order to determine which messages are legal, we use a system based on the notion of 'rights' and 'events', meaning that an agent obtains the right to send a specific message if a certain event has (or has not) taken place. The assignment of such rights is determined by if-then rules in the protocol.

We currently assume agents can send messages following one of these two patterns:

- ('say', `x`)
- ('tell', `y`, `z`)

in which the sender can replace `x`, `y` and `z` by any character string (we will see later that the 'tell' message has the interpretation that the value filled in for `z` will be assigned to a variable of which the name is the string filled in for `y`). The current version of the language does not yet allow users to specify the recipient of a message, so for now we assume that any message is always sent to all the other agents in the MAS. We plan this to change in future versions of SIMPLE. Also, we expect that future versions will support more types of messages.

The interpreter keeps a list of **rights** for each agent in the MAS. A right is a tuple of one of the two following forms:

- ('say', `v`)

– (‘tell’, w)

We say that a right (‘say’, v) **matches** a message (‘say’, x) if and only if x is equal to v, or v is the key word ‘anything’. A right (‘tell’, w) matches a message (‘tell’, y, z) if and only if y equals w. For example: if the agent has the right (‘tell’, ‘price’) then it matches the message (‘tell’, ‘price’, ‘$100’). A message is considered legal if the agent sending the message has at least one right that matches the message. Whenever the interpreter determines that a message is legal, it stores a copy of that message, together with the name of its sender, in the interpreter’s **event history**.

One concept that we have borrowed from EIDE is the concept of a *role*. The rules in the protocol never refer to specific individuals, because we assume that at design time the designer cannot know which agents are going to join the MAS at run time. Instead, the protocol assigns rights to agents based on the roles they are playing. Every agent that enters the MAS (i.e. connects to the communication server) must choose a specific role to adopt, from a number of roles that are defined in the protocol. An auction protocol for example, could define the roles *buyer* and *auctioneer*. The protocol could then define a rule saying that a buyer can only make a bid after the auctioneer has opened the auction.

4 Description of the Language

A protocol is written as a set of sentences that look like natural language, but follow a strict syntax. Although in this paper we will often start sentences with a capital, this is not necessary, as the language is entirely case-insensitive. Like in natural language, the end of a sentence is marked with a period. Unlike most other programming languages, variable names are allowed to contain spaces. Another important property of this language, as we will see at the end of this section, is that it is impossible to write inconsistent protocols.

Definition 1. *A **role definition sentence** is a sentence of the form:*

This protocol defines the role r1 *(plural:* r2*).*

Where the protocol designer can replace r1 *and* r2 *by any character string. The string* r1 *is called the **singular role name** and* r2 *is called **plural role name**.*

For each role in the protocol there must also be exactly one such role definition sentence. For example:

This protocol defines the role buyer (plural:buyers).

Definition 2. *A **role constraint sentence** is a sentence of one of the following forms:*

– *There can be any number of* r.
– *There must be at least* x r.
– *There can be at most* x r
– *There must be at least* y *and at most* x r.
– *There must be exactly* x r.

Where x *and* y *can be any positive integer with* y $<$ x *and* r *is a plural role name from one of the role definition sentences, except in the case that* x=1*, in which case* r *must be a singular role name.*

The following sentence is an examples of a role constraint sentence:

There must be at least 2 buyers.

For each role in the protocol there must be exactly one such role constraint sentence. The interpreter makes sure that these role constraints are not violated. That is, when an agent tries to connect to the communication server with a role for which there are already too many participants, the connection will be refused. If there are not yet enough participants for every role, then every message is considered illegal. In other words: the agents can only start sending messages to one another when there are enough participants for every role.

The main idea of the language, as explained above, is that rights are assigned to the agents by means of if-then rules. An example of such a rule could be:

If the auctioneer has said 'open' then any buyer can tell his bid price.

In order to precisely define which sentences are well formed we first need to introduce a number of terms, namely: *quantifiers*, *identifiers*, *conditions*, and *consequences*.

Definition 3. *A **quantifier** is any of these keywords: no, any, every, a, an, the, that*

Definition 4. *An **identifier** is a sequence of characters of one of the following forms:*

- q r
- *no one*
- *anyone*
- *everyone*
- *he*

Where q *can be any quantifier and* r *can be any singular role name. Identifiers of the form no* r *as well as the identifier 'no one' are called **negative identifiers**. All other identifiers are called **positive identifiers**.*

Definition 5. *A **past-event condition** is a string of characters of one of the following forms:*

- id *has said '*x*'*
- id *has told* x
- pid *has not said '*x*'*
- pid *has not told* x

where id *can be any identifier and* x *can be any character string, and* pid *can be any positive identifier. A past-event condition is called negative if it contains the keyword 'not' or if it contains a negative identifier. A past-event condition is called positive otherwise.*

A past-event conditions is a specific type of condition. Other types of condition are defined later. The idea behind this is that a positive past-event condition is considered true if there is any message in the event history that matches the condition. For example the condition *any buyer has said 'hello'* is considered true if there exists a message in the event history of the form ('say', 'hello') which was sent by an agent playing the role *buyer*. A negative past-event condition is considered true if there is no message in the event history that matches the condition.

Definition 6. *A **right-update consequence** is a string of characters of one of the following forms:*

- `pid` *can say '*`x`*'*
- `pid` *can tell* `x`

where `pid` *can be any positive identifier and* `x` *can be any character string.*

A right-update consequence is a specific type of consequence. Other types of consequences are defined later on.

We can now construct sentences ('rules') of the form *If A then B*, where A is a conjunction of conditions and B is a conjunction of right-update consequences. We say that a rule is **active** if all its conditions are true. Then the idea is that an agent has the right to send a specific message if and only if there is an active rule with right-update consequence that matches that message.

Identifiers are used inside conditions and consequences to determine to which set of agents these conditions and consequences apply. We would like to remark that the quantifiers 'a', 'an', 'any' and 'the' all have exactly the same meaning, so the language contains some redundancy. However, we do consider it very useful to have all of them in the language because they help the protocol designer to write more natural sentences. For example, if an auction protocol contains only one auctioneer it makes much more sense to talk about 'the auctioneer' than about 'any auctioneer'.

Also note that we have included the quantifier 'that'. This quantifier refers to any agent that was also referred to by the last quantifier earlier in the sentence. For example, suppose that a buyer called Alice says 'hello' and then a buyer called Bob says 'hi', then the condition:

any buyer has said 'hello' and any buyer has said 'hi'

is true. However, the condition:

any buyer has said 'hello' and that buyer has said 'hi'

is false, because 'that buyer' refers to the same agent as the one that said 'hello' (which is Alice). This second condition would only be true if the messages ('say' 'hello') and ('say' ,'hi') had been sent by the same agent. Likewise, we have included the identifier 'he', which refers to the same agent as the last identifier that appeared earlier in the sentence. For example:

If any buyer has said 'hello' and he has said 'hi'

We may not want the rights of an agent to depend only on past events, but also on values of variables. Variables in SIMPLE are called **properties**. A property can be assigned to the protocol, or can be assigned to individual agents. For example, an auction may have a property 'highest bid' and each buyer may have a property 'bid price' to represent the price he or she has bid. If we have for example properties 'the price' and 'the account balance' then we can say things like:

If the price is lower than the account balance then any buyer can say 'buy'.
If the price is higher than 10 then the auctioneer can say 'sold'.

Properties can be added to a protocol by including property initialization sentences.

Definition 7. *A **property initialization sentence** is a sentence of one of the following forms:*

- *Initially,* `x` *is* `v`.
- *Every* `r` *has a* `x`*, which is initially* `v`.
- *Every* `r` *has an* `x`*, which is initially* `v`.

where `x` *can be any character string,* `v` *can be any character string, number, or identifier and* `r` *can be any singular role name.*

For example:

Every buyer has an age, which is initially 0.

Definition 8. *A **property condition** is a clause of one of the following forms:*

- `x` *is* `v`
- `x` *is not* `v`
- `x` *is higher than* `n`
- `x` *is lower than* `n`

where `x` *can be any character string,* `v` *can be any string, number or identifier, and* `n` *can be any number. The string* `x` *is called the **property name**, and* `v` *and* `n` *are called the **value**.*

Note that the current version of SIMPLE supports three types of properties: strings, numbers and identifiers. The type of a property is determined implicitly. That is: if the parser of the protocol is able to interpret the initial value of a property as a number, then the property is considered to be of type number, and likewise for identifiers. In all other cases the property is considered a string.

Definition 9. *A **property-update consequence** is a clause of the form:*

- `x` *becomes* `y`
- `x` *is* `v`
- `x` *is increased by* `n`
- `x` *is decreased by* `n`

where `x` *and* `y` *can be any character string,* `v` *can be any string, number of identifier, and* `n` *can be any number.*

Definition 10. *A **current-event condition** is a string of characters of one of the following forms:*

- `id` *says* '`x`'
- `id` *tells* `x`

where `id` *can be any identifier and* `x` *can be any character string.*

In order to change the values of properties (either assigned to the protocol or to an individual agent) we can use property-update rules.

Definition 11. *A **property-update rule** is a sentence of the form:*

- *When* `x` *then* `z`.

Where `x` *is a current-event condition and* `z` *is a conjunction of property-update consequences.*

Examples of property-update rules are:

> *When any buyer says 'bid!' then his bid price is increased by 10.*
> *When the auctioneer says 'sold' then the last bidder becomes the winner.*

Note that the clause *x becomes y* means that the value of property `y` is overwritten with the value of property `x`. This can be understood as follows: suppose we have a property called *Carol's sister* and a property called *Bob's wife*. Furthermore, suppose that *Carol's sister* is initialized to the value *'Alice'*. Then the clause *Carol's sister becomes bob's wife* means that the value *'Alice'* is copied into the property *Bob's wife*. Note that when a property is assigned to an agent we use the key word 'his' to refer to the agent that owns the property. To be precise: it refers to the last agent that appears earlier in the sentence. So in the above example, 'his bid price' refers to the property named 'bid price' assigned to the agent that said 'bid!'.

Another way that values of properties are updated is when a message of type ('tell', `x`, `y`) is sent. In that case the value `y` is assigned to a property with name `x`. For example, whenever an agent sends the message ('tell', 'the price', 100), the value 100 is automatically assigned to a property with the name 'the price'. The protocol does not need to contain any property-initialization sentence for such a property.

Definition 12. *A **right-update rule** is a sentence of the form:*

- `id` *can always say* `v`.
- `id` *can always tell* `v`.
- *If* `x` *then* `y`.
- *If* `x` *then* `y`*, as long as* `w`.

where `id` *is an identifier,* `v` *can be any character string,* `x` *and* `w` *are conjunctions of past-event conditions and/or property conditions and* `y` *is a conjunction of right-update consequences (the conditions in* `w` *are also referred to as **constraints**).*

Note that we allow such a rule to have no conditions at all, so that it is always active. In that case the protocol designer needs to include the keyword 'always' after the keyword 'can'. Also note that right-update rules are written in past tense, while property-update rules are written in present tense. This is because they are interpreted in a fundamentally different way, which we will explain in Section 5. Furthermore, we see in this definition that right-update rules may contain so-called *constraints*. A constraint is similar to a property condition, but is written at the end of the sentence, and indicated by the keywords *as long as*.

> *If the auctioneer has said 'open' then any buyer can tell his bid price, as long as his bid price is higher than the current price.*

A rule containing constraints is considered active if and only if all its conditions and constraints are satisfied. The difference between constraints and conditions, is that constraints refer to property values inside the consequences, whereas other conditions may only refer to past events or properties that do not appear inside the consequences. This distinction means that the truth of a condition is independent of any message, and therefore can already be determined before a specific message is sent, while the truth value of a constraint on a message X can only be determined after the participant has submitted message X, when the interpreter is verifying whether message X is legal. In the example sentence above for instance, the constraint says that the bid price told by the buyer, must be higher than the current price. This can of course only be checked *when* the buyer is telling his bid price, and not before.

Furthermore, we would like to remind the reader that right-update consequences can only have positive identifiers. This is important, because it means that a consequence can only *give* rights to an agent, but not take them away. Nevertheless, we can still make agents lose rights, but we do that by using negative conditions, rather than negative consequences. Take for example the following rule:

> *If the auctioneer has not said 'sold!' then any buyer can say 'bid!'.*

Here, every buyer initially has the right to say 'bid!', but loses that right once the auctioneer says 'sold!', because the condition becomes false (assuming there is no other active rule that gives the buyer the right to say 'bid!'). The big advantage of only allowing positive consequences, is that this makes it impossible to write inconsistent rules. An inconsistency would mean that there is one rule that specifies that you can do something, while another rule says you cannot do that. This is serious problem that one often encounters, for example in law. However, since we only allow positive consequences, this could never happen in our language.

Lemma 1. *A protocol written in SIMPLE is guaranteed to be free of inconsistencies.*

Proof. The proof is easy: in our language an agent has the right to do something if and only if there is an active rule with a consequence that gives this right to the agent. This can never lead to inconsistencies: either such a rule exists or not.

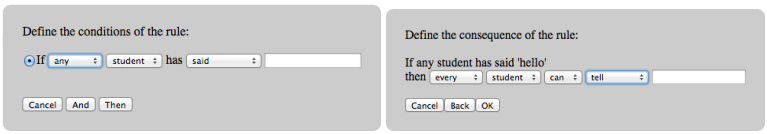


Fig. 1. Two screen shots of the SIMPLE editor. Users write sentences simply by selecting available options, and they can only write free text whenever the syntax rules indeed allow that. Therefore it is impossible to write malformed sentences.

5 The SIMPLE Interpreter

We will now describe the software component that interprets and enforces the protocols.

Whenever an agent tries to send a message, this message is first analyzed by the interpreter. The interpreter verifies if the agent sending the message indeed has the right to say that message and, if so, updates its internal state and forwards the message to the other agents connected to the server. If the sender of the message does not have the right to send that message he or she is notified that the message has failed. The message will in that case not be forwarded to the other agents and the internal state of the interpreter is not updated. In fact, we consider this message as not sent.

The internal state of the interpreter is defined as a list of all messages that have so far been sent successfully (the *event history*), a table that maps the name of each property to the current value of that property, a table that maps the name of each agent in the MAS to the role it is playing, and a table that maps the name of each agent in the MAS to a list of rights for that agent. Every time an agent tries to send a message, the interpreter follows the following procedure:

1. The list of rights of that agent is made empty.
2. For each right-update rule in the protocol, the interpreter verifies if its conditions are true:
   - If the condition is a property condition then it checks whether that property currently has the proper value to make the condition true.
   - If the condition is a past-event condition, the interpreter tries to find an event in the event history that matches the condition. If such an event is indeed found, then the condition is considered true.

   A rule for which all conditions are true is labeled as 'active'.
3. For each right-update consequence in each active rule, the interpreter checks whether the identifier matches the sender of the message and, if yes, adds the right corresponding to this consequence to the sender's list of rights. If this consequence has any constraints assigned to it, they are stored together with the right.
4. After all the rights of the sending agent have been determined the interpreter verifies whether any of them matches the message that the agent is trying to send.
5. Next, if the agent indeed has that right the interpreter checks whether its constraints (if any) are satisfied.
6. If the sending agent has the proper right, and all its constraints are satisfied then the interpreter determines if there are any property-update rules in the protocol

for which the condition matches the message. If yes, the properties in the rule's consequences are updated accordingly.

7. Finally, if the agent has the right to send the message and its constraints are satisfied, a copy of the message is stored in the event history, together with the name of the sender, and the message is forwarded to all other agents in the MAS.

It is important to note here that property-update rules and right-update rules are treated in a different way. To be precise: to verify whether a past-event condition is true, the interpreter compares the condition with all messages in the event history. Since messages are never removed from the event history this means that whenever a past-event condition becomes true, it remains true forever. For example, when a buyer says 'hello' then the condition *any buyer has said 'hello'* becomes true, and remains true forever. For negative conditions exactly the opposite holds: the condition *no buyer has said 'bye'* is initially true, but as soon as a buyer says 'bye' it becomes false, and will stay false forever.

The current-event conditions on the other hand are only considered true at the moment that the corresponding message is under evaluation of the interpreter. That is, the condition *when a buyer says hello* is considered to be true only while the interpreter is evaluating the message ('say', 'hello') sent by some agent playing the role of buyer. As soon as the interpreter handles the next message this condition is considered false again. The reason for this is that we consider that when you obtain a right, you keep that right for an extended period of time, until one of the negative conditions in the rule becomes false. Updating of a property on the other hand, is a one-time event that only takes place at the moment a certain message is sent.

6 Examples

We here provide two examples of protocols. Both have been tested and are correctly executed by the interpreter.

English Auction Protocol:

This protocol defines the role buyer (plural:buyers).
This protocol defines the role auctioneer (plural:auctioneers).
There must be exactly 1 auctioneer.
There must be at least 2 buyers.

Initially, the highest bidder is no one.
Initially, the winner is no one.
Initially, the current price is 0.
Every buyer has a bid price which is initially 0.

If the auctioneer has not said 'close' then he can say 'open'.
If the auctioneer has said 'open' then the auctioneer can say 'close'.
If the auctioneer has said 'open' and the auctioneer has not said 'close' then any buyer can tell his bid price, as long as his bid price is higher than the current price.

When a buyer tells his bid price then his bid price becomes the current price and he becomes the highest bidder.
When the auctioneer says 'close' then the highest bidder becomes the winner.

Dutch Auction Protocol:

This protocol defines the role buyer (plural:buyers).
This protocol defines the role auctioneer (plural:auctioneers).
There must be exactly 1 auctioneer.
There must be at least 2 buyers.

Initially, the price is 1000.
Initially, the winner is no one.

If no buyer has said 'mine' then the auctioneer can tell the next price, as long as the next price is lower than the price.
When the auctioneer tells the next price then the next price becomes the price.
If the auctioneer has told the price and no buyer has said 'mine' then any buyer can say 'mine'.
When a buyer says 'mine' then he becomes the winner.

7 Future Work

We consider that the language as it is, is still too limited to be of real practical use. We here list the shortcoming that we consider most important and that we plan to fix in the near future, as well as other improvements that we are considering.

Firstly, we will add the possibility to specify the recipient of a message. Currently every message is sent to all other agents in the MAS, which makes it impossible to send confidential information. This means we will allow to write sentences such as:

If the auctioneer has said 'welcome' to a buyer then that buyer can say 'hello' to the auctioneer.

Secondly, we would like the protocol designer to be able to express that a certain event must have taken place a certain number of times. For example:

If a buyer has told his bid price more than 5 times...

Thirdly, we would like to add more types of messages. and maybe even allow the protocol designer to define message types. That would make it possible to use certain domain-specific verbs. We could even take this a step further and allow the protocol designer to define new data types. Defining new types of objects is typically something that Inform 7 can handle well, so we may draw some inspiration from that language. Furthermore, we will add a system that determines at run time, whenever an agent tries to send an illegal message, which conditions first need to be fulfilled before the agent can indeed legally send that message. In this way the system can explain to the user why he or she made a mistake and will help the user to understand new protocols. In order to make the language more flexible and expressive, we will delve into literature about linguistics and apply some of its principles to our language. Finally, we are also considering the possibility to add support for model checking to our framework.

8 Acknowledgments

Supported by the Agreement Technologies CONSOLIDER project, contract CSD2007-0022 and INGENIO 2010 and CHIST-ERA project ACE and EU project 318770 PRAISE.

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(2-4), 133–157 (2006)
2. Argente, E., Criado, N., Botti, V., Julian, V.: Norms for agent service controlling. EUMAS-08 pp. 1–15 (2008)
3. Artikis, A., Kamara, L., Pitt, J., Sergot, M.: A protocol for resource sharing in norm-governed ad hoc networks. In: Leite, J.a., Omicini, A., Torroni, P., Yolum, p. (eds.) Declarative Agent Languages and Technologies II, Lecture Notes in Computer Science, vol. 3476, pp. 221–238. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/11493402_13
4. Belnap, N., Perloff, M.: Seeing to it that: A canonical form for agentives. In: Kyburg, Henry E., J., Loui, R.P., Carlson, G.N. (eds.) Knowledge Representation and Defeasible Reasoning, Studies in Cognitive Systems, vol. 5, pp. 167–190. Springer Netherlands (1990), http://dx.doi.org/10.1007/978-94-009-0553-5_7
5. Broersen, J., Dignum, F., Dignum, V., Meyer, J.J.C.: Designing a deontic logic of deadlines. In: Deontic Logic in Computer Science, pp. 43–56. Springer Berlin Heidelberg (2004)
6. Cardoso, H.L., Urbano, J., Rocha, A.P., Castro, A.J., Oliveira, E.: Ante: Agreement negotiation in normative and trust-enabled environments. In: Ossowski, S. (ed.) Agreement Technologies, Law, Governance and Technology Series, vol. 8, pp. 549–564. Springer Netherlands (2013), http://dx.doi.org/10.1007/978-94-007-5583-3_32
7. Cranefield, S.: A rule language for modelling and monitoring social expectations in multi-agent systems. Tech. rep., In: Selected and Revised papers from the AAMAS 2005 Workshop on Agents, Norms and Institutions for Regulated Multiagent Systems. Volume 3913 of Lecture (2005)
8. Cranefield, S., Winikoff, M.: Verifying social expectations by model checking truncated paths. Journal of Logic and Computation 21(6), 1217–1256 (2011), http://logcom.oxfordjournals.org/content/21/6/1217.abstract
9. Dastani, M., Tinnemeier, N.A., Meyer, J.J.C.: A programming language for normative multi-agent systems (2009)
10. Esteva, M., de la Cruz, D., Sierra, C.: Islander: en electronic institutions editor. vol. 3, pp. 1045–1052. ACM PRESS, Bologna, Italy (July 15-19 2002)
11. Esteva, M., Rodríguez-Aguilar, J.A., Arcos, J.L., Sierra, C., Noriega, P., Rosell, B., de la Cruz, D.: Electronic institutions development environment. In: AAMAS (Demos). pp. 1657–1658 (2008), http://www.iiia.csic.es/files/pdfs/eide.pdf
12. Fornara, N., Cardoso, H.L., Noriega, P., Oliveira, E., Tampitsikas, C., Schumacher, M.I.: Modelling Agent Institutions, chap. 18, pp. 277–307. No. 8, Springer-Verlag GmdH (2013)
13. García-Camino, A.: Ignoring, forcing and expecting simultaneous events in electronic institutions. In: Proceedings of the 2007 International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems III. pp. 15–26. COIN'07, Springer-Verlag, Berlin, Heidelberg (2008), http://dl.acm.org/citation.cfm?id=1791649.1791652
14. Governatori, G., Rotolo, A., Sartor, G.: Temporalised normative positions in defeasible logic. In: Procedings of the 10th International Conference on Artificial Intelligence and Law. pp. 25–34. ACM Press (2005)

15. van der Hoek, W., Roberts, M., Wooldridge, M.: Social laws in alternating time: effectiveness, feasibility, and synthesis. Synthese 156(1), 1–19 (2007), http://dx.doi.org/10.1007/s11229-006-9072-6
16. Hübner, J.F., Sichman, J.S.a., Boissier, O.: S-moise+: A middleware for developing organised multi-agent systems. In: Boissier, O., Padget, J., Dignum, V., Lindemann, G., Matson, E., Ossowski, S., Sichman, J.S.a., Vázquez-Salceda, J. (eds.) Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, Lecture Notes in Computer Science, vol. 3913, pp. 64–77. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11775331_5
17. de Jonge, D., Rosell, B., Sierra, C.: Human interactions in electronic institutions. In: Chesnevar, C., Onaindia de la Rivaherrera, E., Ossowski, S., Vouros, G. (eds.) 2nd International Conference on Agreement Technologies. vol. 8068, pp. 75–89. Springer, Springer, Beijing, China (01/08/2013 2013)
18. Kollingbaum, M.J.: Norm-governed practical reasoning agents. Ph.D. thesis, University of Aberdeen (2005)
19. Kröger, F.: Temporal Logic of Programs. Springer-Verlag New York, Inc., New York, NY, USA (1987)
20. Lewis, D.: Semantic analyses for dyadic deontic logic. In: Stenlund, S. (ed.) Logical Theory and Semantic Analysis: Essays Dedicated to Stig Kanger on His Fiftieth Birthday, pp. 1–14. Reidel, Dordrecht (1974)
21. Lopez y Lopez, F., Luck, M., Puebla, A.: A model of normative multi-agent systems and dynamic relationships. In: Regulated Agent-Based Social Systems. Volume 2934 of Lecture Notes in Artificial Intelligence. pp. 259–280. Springer (2004)
22. Makinson, D., Van Der Torre, L.: Input/output logics. Journal of Philosophical Logic 29(4), 383–408 (2000)
23. Meyer, J.J.C.: A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. Notre Dame J. Formal Logic 29(1), 109–136 (12 1987), http://dx.doi.org/10.1305/ndjfl/1093637776
24. Nelson, G.: Natural language, semantic analysis and interactive fiction. http://inform7.com/learn/documents/WhitePaper.pdf (2014)
25. Nute, D.: Defeasible Deontic Logic. Springer (1997)
26. Sergot, M., Craven, R.: The deontic component of action language nc+. In: Goble, L., Meyer, J.J.C. (eds.) Deontic Logic and Artificial Normative Systems, Lecture Notes in Computer Science, vol. 4048, pp. 222–237. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11786849_19
27. Tampitsikas, C., Bromuri, S., Schumacher, M.: Manet: A model for first-class electronic institutions. In: Cranefield, S., Vazquez-Salceda, J., van Riemsdijk, B., Noriega, P. (eds.) Coordination, Organizations, Institutions, and Norms in Agent Systems VII. Lecture Notes in Artificial Intelligence, vol. 7254. Springer Verlag (2012), http://link.springer.com/chapter/10.1007/978-3-642-35545-5_5
28. Uszok, A., Bradshaw, J.M., Lott, J., Breedy, M., Bunch, L., Feltovich, P., Johnson, M., Jung, H.: New developments in ontology-based policy management: Increasing the practicality and comprehensiveness of kaos. Policies for Distributed Systems and Networks, IEEE International Workshop on 0, 145–152 (2008)
29. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Implementing norms in multiagent systems. Multiagent system technologies pp. 313–327 (2004)
30. von Wright, G.H.: Deontic logic. Mind 60, 1–15 (1951)

Mind as a Service: Building Socially Intelligent Agents

Virginia Dignum

Delft University of Technology, The Netherlands,
M.V.Dignum@tudelft.nl

Abstract. The ability to exhibit social behaviour is paramount for agents to be able to engage in meaningful interaction with people. In fact, agents are social beings at the core. That is, agent behaviour is the result of more than just rational, goal-oriented deliberation. This requires novel agent architectures that start from and integrate different socio-cognitive elements such as emotions, social norms and personality. Current agent architectures however, do not support the construction of social agents in a structured, modular and computational- and design-efficient manner. Inspired by Service-Oriented Architectures (SOA), in this paper we propose MaaS (Mind as a Service) as a modular architecture for agent systems that enables the composition of different socio-cognitive capabilities into a running system. Depending on the characteristics of the domain, agent's deliberation will require different social capabilities. We propose to model these capabilities as services, and define a 'Deliberation Bus' that enables to design deliberation as a composition of services. This approach allows to define deliberation cycles that are situational and dependent on the available components in order to cope with the complexity of social and physical environments in parallel. We furthermore propose a Service Interface Descriptor language to encapsulate service functionalities in a uniform way.

1 Introduction

The potential of artificial intelligent systems to interact and collaborate not only with each other but also with human users is no longer science fiction. Healthcare robots, intelligent vehicles, virtual coaches and serious games are currently being developed that exhibit social behaviour - to facilitate social interactions, to enhance decision making, to improve learning and skill training, to facilitate negotiations and to generate insights about a domain. In all these cases, the ability to exhibit social behaviour is paramount for successful functioning of the system.

We informally define social intelligent agents as systems whose behaviour can be interpreted by others as that of perceiving, thinking, moral, intentional, and behaving individuals; i.e. as individuals that can consider the intentional or rational meaning of expressions of others, and that can form expectations about the acts and actions of others [27]. In this light, functionalities required

from social intelligent agents include the ability to reason about norms, beliefs and culture-specific contexts, to display and understand emotions, to balance between goal-directed and reactive behaviour, maintain a sense of identity, to form expectations about the other's acts and actions, etc. An important aspect of social behaviour is the capability to integrate and to choose between different types of behaviour, such as e.g. utility-based, mimicry or altruistic behaviours based on the physical and social context.

In the last years, many systems have been developed which possess some of these characteristics. In particular, work on Intelligent Virtual Agents and on Social Robotics has delivered many promising results. However, we are still lacking theories, tools and methodologies to guide and ground these developments. That is, current approaches often result in ad-hoc, unstructured solutions. Success and applicability are often more due to the expertise and art of the developers, rather than on robust engineering principles. Moreover, in most cases, social aspects are 'added-in' on top of existing architectures, such as BDI, which does not allow to model the rich inter-dependencies between social capabilities needed to generate social behaviour [12].

The necessity to develop working real-world systems capable of exhibiting social behaviour for the purpose of interaction and collaboration with people requires engineering approaches to explore the full potential of social artificial intelligent systems on a larger scale, mandates a new understanding of social intelligent agents. Architectures, tools and methodologies are needed to realize this potential and engineer applications with a high level of robustness and quality. Only then can we reach certification guarantees acceptable by industry and society.

In this paper, we introduce the vision of MaaS (Mind as a Service), a framework to develop the 'minds' of social intelligent agents, based on the composition of different cognitive modules, or services. In particular, we focus on the *minds* of social intelligent agents, i.e. the representations and processes that enable social behaviour. In the context of this paper, the concept of 'mind' should be understood as an analogy of the human mind rather than as a faithful representation. We us the term 'mind' in order to stress the importance that in many application this mind will be connected to a physical or virtual body of the agent. It is important to notice that this approach should be seen as orthogonal to current research focus on emulating the human brain, such as that taken by the European flagship project "Human Brain". In contrast to this project, our aim is to develop synthetic models that exhibit behaviour that can be seen as social, and not understand the human brain in order to emulate its computational capabilities.

MaaS combines a service-oriented architecture with formal specification languages to verify behaviour. In this position paper, we outline the MaaS approach, present the grounding theories on which MaaS is based, and discuss its main challenges. The work presented here should be seen as a first proposal towards a comprehensive theory and tools to build and analyse social intelligent agents.

2 Related work

Understanding social behaviour is the first step towards building social minds [7]. Social intelligence is defined as an aggregate of different capabilities, including awareness, social beliefs and attitudes, and the ability to change [6, 16]. In his book, "The Society of Mind" Minsky explores the notion that the mind consists of a great diversity of mechanisms: every mind is really a rich and multifaceted *society* of structures and processes, different for every individual as result of genetics, millennia of human cultural evolution, and years of personal experience [23]. Societies of Mind are composed of agents with specific functionality that can be combined together to perform functions more complex than any single agent could, and ultimately produce the many abilities we attribute to minds. Despite the great popularity of this work, there have been few attempts to implement the Society of Minds theory, especially due to the fact that Minsky presents his ideas at different level of abstraction and provides few handles for construction of minds. In its main objective, that of providing a modular, compositional and adaptable architecture for intelligent systems, MaaS takes a similar view of mind as that proposed by Minsky and can be seen as providing a principled engineering framework to develop systems similar to the Society of Minds. However, the basic concepts behind MaaS and the Society of Minds are quite different.

Decision-making processes are influenced by individual and social sources [22]. Social influences are often described in terms of social rules that are followed, such as 'obey your parents' or 'mimic the behaviour of your peers'. Individual influences are usually expressed in terms of personal goals or utilities and lead to 'rational' decision rules. The social sciences describe many mechanisms or schemas used by humans to link these capabilities (e.g. salience, priming, motivation and regulation), determine how decisions are made and generate complex social behaviour [1, 15]. Similar processes occur in human-agent interaction because social signals (like emotional expressions) produced by computational agents are processed by humans in a similar manner as signals which are produced by humans [32].

Computational cognitive models, such as ACT-R [3] and SOAR [9] produce intelligent behaviour by employing quantitative measures, which means that different factors take the same form in the deliberation process. This makes it difficult to manage, control and vary different socio-cognitive aspects because these cannot easily be isolated in the decision rules. Moreover, once models get larger they lack transparency to link observed behaviour to the implementation. Existing architectures used to construct virtual agents and intelligent game characters, such as FAtiMA [11], GRETA [21] or CIGA [31] can achieve fairly realistic behaviours that are computationally efficient, but are generally developed for a very specific domain of application. Given this domain-oriented focus, their results are not easily reusable in applications that require slightly different social aspects. I.e. sometimes norms play a major role in a training application while in health care applications emotions might take precedence. Moreover, the focus of these approaches is geared to the visualisation of the behaviours by the virtual characters in terms of e.g. gestures, or facial expressions.

Recently Kaminka and Dignum et al [18, 12] discussed the many challenges of designing the social behaviour of agents. In this paper, we propose an initial architecture to build agents that can meet those challenges. MaaS takes a modular, service-oriented approach to build social intelligent agents, resulting in flexible and adaptable deliberation. Nevertheless, existing cognitive models in AI are often too simplistic, mostly suitable for well-defined problem domains, platform- or domain-specific, or computationally too complex [29, 20, 9].

Deliberative agent models, such as BDI [33], have formal logic-grounded semantics, but often require extensive computational resources to deal with social contexts, or use game-theoretic rules that are too simple to capture many of the rich interactions that take place in real-world scenarios [5]. BDI does use different modules for beliefs, desires and intentions. However, these are geared towards individual influences on decision making. These models thus lack an explicit representation for social influences. One can represent all these social influences in the beliefs or goals of an agent, but that leads to the same objection as against the cognitive models; the rules become convoluted and different aspects cannot easily be managed separately.

Other decision-theoretic approaches often used are(PO)MDPs - (Partially Observable) Markov Decision Processes, which capture many of the facets of real world problems, but unrealistically assume that whatever system is solving the MDP knows at every point what state it is in. Moreover, (PO)MDPs do not scale well and lack the modularity needed to analyse the results of large models [4].

The Subsumption Architecture [8, 30] takes a reactive perspective, through an hierarchy of task-accomplishing behaviours (simple rules) without necessarily a central control. Lower layers correspond to 'primitive' behaviours and have precedence over higher (more abstract) ones. This architecture is simple in computational terms, but is conceptually obscure due to its 'black box' character.

In an attempt to balance different aspects, and improve the separation of concerns, AOSE (Agent-Oriented Software Engineering) addresses adaptation, concurrency, and fault-tolerance issues [28, 13] of the development of agent systems. However, most current AOSE approaches see agents as an application layer software component operating on middleware platforms to gain access to standardised infrastructures. Specifically, such approaches provide syntactic constructs to represent domain knowledge and agent functionalities but lack the formal semantics to reason about agent behaviour at higher levels of abstraction, in terms of socio-cognitive concepts. This leads to results that are not generalizable to other frameworks and applications.

3 The MaaS Vision

As discussed in the previous section, many approaches exist to model different aspects of social intelligence. Our proposal is not to develop yet another model, but enable to integrate different models into working software systems, with variable levels of precision and realism. We therefore aim to build social intelligent

systems as a modular, service-based architecture which enables formal verification and conceptual clarity while making possible the integration of different reasoning architectures.

The *'Mind as a Service'* (MaaS) architecture proposed in this paper, represents social intelligent agents as a composition of software services, each designed to implement a specific socio-cognitive functionality. MaaS systems behave in human-like fashion by integrating individual considerations and social influences in their decision making process, and taking into account situational differences. This approach follows recent literature suggesting that rational behaviour requires the input from different socio-cognitive abilities [2].

This approach is based on three pillars. Firstly, models for social deliberation and interaction should be grounded in existing proven psycho-sociological theories, but also be computationally sound and sufficiently 'light' to be easily be embedded and reused into avatars, robots or other intelligent systems. By expressing algorithms in logical terms, explanation and synthesis of socio-cognitive behaviour is possible [24]. This view is orthogonal to the current AI research focus on emulating the human brain[1], in such that our aim is to develop synthetic models that exhibit behaviour that can be seen as social, and not understand the human brain in order to emulate its computational capabilities.

Secondly, development of a computational platform to build MaaS as a composition of socio-cognitive services. This platform will allow to build modular socio-cognitive deliberation architectures and to analyse the consistency of different compositions in terms of accuracy of real world behaviour. Given the explicit formal representation of MaaS models this allows for introspection of the drives of an agent's behaviour. By applying Service-Oriented Architecture (SOA) principles [14], the resulting systems are scalable and flexible, as services can be replaced by other services, and the system includes only those services required for its aims. Through composition, new services can be created from a set of existing services. Moreover, each socio-cognitive function can be modelled in many ways, resulting in several services for the same socio-cognitive ability with different levels of complexity and realism, which can be interchanged depending on the requirements of the application. MaaS services can be addressed in a uniform way through a standard interface that is platform- and domain-independent.

Thirdly, methodology to develop MaaS that can be embedded in artificial interactive systems. This methodology provides guidelines for domain analysis, evaluate the socio-cognitive functionalities required for interaction and their level of realism, construct and compose the relevant socio-cognitive services, and evaluate results. The use of the methodology and framework will be evaluated in the development of prototypes for three case studies.

Ultimately, we aim to develop a complete framework that integrates formal theory, software development tools, and methodology to build artificial minds in a structured, compositional way. Through this framework, social intelligent

[1] Such as is advocated e.g. by the Human Brain Project (https://www.humanbrainproject.eu/)

agents can be build that are modular, flexible, adjustable and verifiable. This aim leads directly to the following challenges that we face for the realisation of MaaS:

Modular: requires definitions and models to represent different theories (describing socio-cognitive capabilities) and verify the resulting computational models. To address this challenge we propose a meta-modelling approach to specify socio-cognitive capabilities.

Flexible: Each application domain requires different abilities at different levels of precision. Our approach to this challenge is twofold: 1) we provide procedures and guidelines to identify relevant socio-cognitive modules given the requirements of an application domain, and 2) we define uniform interface descriptions that enable the composition and encapsulation of different socio-cognitive models.

Adjustable: Which socio-cognitive capabilities are needed, at which level of realism and computational complexity, and how to integrate the different capabilities into a deliberation mechanism, is dependent on the characteristics of the domain. MaaS should provide an extendible library of socio-cognitive services. We aim at a *plug-and-play* mechanism to combine these services in many ways resulting in different decision-making paradigms (e.g rational or behavioural models of decision making).

Verifiable: To judge the appropriateness of the behaviour of a MaaS system, computational theories and tools are needed to analyse the composed effects of social capabilities. By specifying formal representations of socio-cognitive theories we will be able to use formal model-checkers to verify whether a MaaS satisfies some desired properties.

3.1 The MaaS development process

In order to integrate different models in a structured way we follow a Model-Driven Engineering (MDE) approach [19]. This enables to develop models that make sense from the point of view of a domain expert, and that can serve as a basis for implementing systems. Formal models of socio-cognitive functions are the basis for the meta-models which can then be used to generate generic service models, similar to a Platform Independent Model (PIM) used in MDE, which are refined into specific conceptual designs realizing the functional requirements and characteristics of the application domain. PIMs are used as a blueprint to develop and compose software services. Through the Deliberation Bus (section 4.2) these services are composed into an operational MaaS that can be embedded in social intelligent artefacts that interact with people, such as Embodied Virtual Agents (EVAs) or other avatars or cognitive robots. This process is illustrated in Figure 3.

Finally, the resulting MaaS systems can then be embedded in social intelligent artefacts that interact with people. The MaaS process outlined above, is depicted in Figure 3.

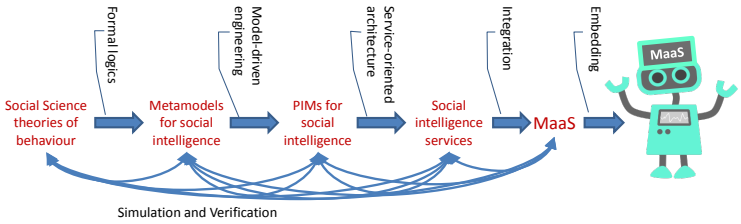


Fig. 1. The MaaS development process

3.2 Development Environment

Our aim is to develop MaaS system can then be embedded in interactive software environments, such as game characters, virtual assistants or robots. To this extent, we are developing MindBuilder, a computational platform to design and implement MaaS as service compositions, that integrates logical specification languages for socio-cognitive capabilities and the formal algorithms to check their behaviour, and the MindBuilder platform and methodology for the specification, integration, simulation and reuse of MaaS as a composition of services, and a methodology to analyse and develop MaaS systems for specific application domains taking into account ethical, social and technical considerations.

The objective is to generate computational models of socio-cognitive capabilities through a semi-automatic transformation of the formal models described above. MindBuilder, illustrated in Figure 2, includes the following functionalities:

- develop formal models of social sciences theories to be used as basis for socio-cognitive software services;
- compose services into MaaS systems using uniform service interfaces;
- provide library capabilities to store and search for services.
- verify the behaviour of MaaS systems, using simulation and model checking;

Resulting MaaS systems can be embedded in different interactive platforms to provide social intelligence capabilities.

The MindBuilder methodology supports the identification of the socio-cognitive capabilities required for the domain, and their level of realism, guides the development of domain-specific versions of existing models and services, and defines the parameters for analysis of results using simulation.

3.3 Example Scenario

To illustrate the MaaS vision, we describe its possible application to develop a virtual coach for children with overweight, JOGG. The socio-cognitive capabilities required by JOGG include the ability to show emotions, and to understand norms and values. E.g. the virtual coach should express happiness when the

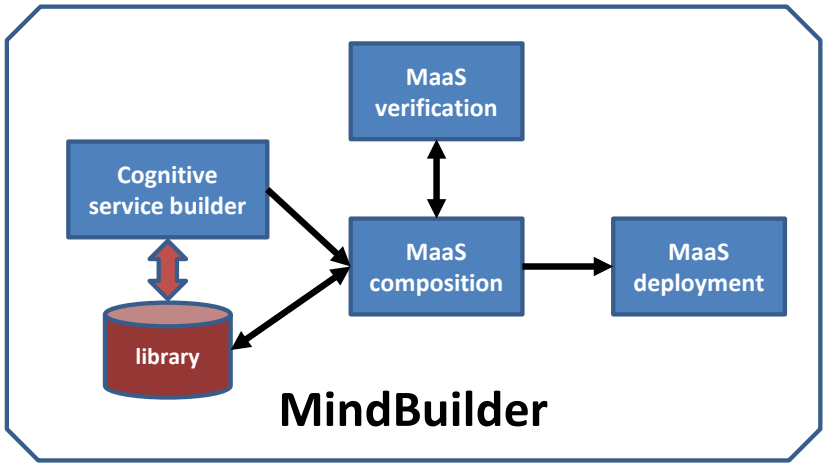


Fig. 2. The MindBuilder Architecture

user has successfully performed a task, should be persuading when suggesting a course of action, should monitor norms, such as the obligation to exercise daily, or the prohibition to snack too often, and enforce values such as privacy, but should also be able to decide when to break a norm, for example violate the norm of privacy and notify a doctor if the health of the user is perceived to be very poor.

Different social science theories exist to describe and analyse these socio-cognitive abilities. To name just a few, emotions can be described using e.g. the OCC model [26], or by simple rules that relate happiness to the fulfilment of one's goal, and norms can be modelled using e.g. deontic logics [34], or by the normative theory of Kahnemann [17]. The MaaS methodology will support the analysis of the domain to determine which base sociological theories are the most suitable, and what level of realism is required.

The MindBuilder Library may already contain services implementing these theories, or new services should be specified using MindBuilder Design. The required services are then tailored to the case, e.g. specifying specific norms on physical activity and nutrition, relevant values such as privacy, and suitable emotional expressions in the given cultural context of use. Using MindBuilder Composition component, services are composed into a MaaS. In order to determine the most adequate compositions, and which level of detail and realism of socio-cognitive services is required, MindBuilder Simulation is used to analyse different MaaS configuration options. Different configurations representing different deliberation mechanisms can be checked, e.g. to determine the effect of a norm on the emotion of the MaaS and vice-versa, to check how norm violations affect values, or to determine the effect of e.g. mimicry or goal-orientation as basis for the MaaS deliberation. The resulting MaaS can then be embedded in

an app to be used to support the user control their weight and maintain an active lifestyle.

4 MaaS Deliberation

Social deliberation in MaaS results from the integration of different socio-cognitive capabilities, modelled as software services. In order to realise the MaaS vision, we need both the means to describe these services in a uniform way (Service Interface Description), and the ways to combine them into meaningful deliberation (Deliberation Bus).

4.1 Service Interface Description

In order to ensure service integration into MaaS systems in a robust, resilient, dependable and scalable manner, we need to develop interfaces between services, and to identify and represent quality of service expectations. A service-oriented approach enables to separate service implementation from service specification. Service Interface Descriptors (SID) will describe the functionality offered by a service, independently from its implementation. As such, services can be seen as black boxes, where operational details are abstracted by the SID. Other services rely on SID to call the service.

As in Situation Calculus, we model the domain world as progressing through a series of states, as a result of various actions being performed within the world. A social state is defined as a set of fluents (properties whose truth changes over time). These fluents represent physical situations (agent is in place X), emotional aspects (agent is happy), relational aspects (agent A is friend of B), and other issues pertinent to the situation. A socio-cognitive service is then a transition from one (social) state to another. I.e., services take a state as input and result in an alteration of that state, that is a change in the value of some of the state fluents. SIDs describe which fluents are modifiable by the service, under which circumstances (i.e. fluents describing the preconditions for using the service).

A service-oriented approach enables to separate service implementation from service specification. We use Service Interface Descriptors (SID) to describe the functionality offered from a service, independently from its implementation. As such services can be seen as black boxes, encapsulated by SID. Other services rely on the SID to call the service. SIDs indicate which fluents are modifiable by the service, under which circumstances (i.e. fluents describing the preconditions for using the service).

Each service acts over a specific set of fluents. Several services may be active at the same time, and can call each other to perform some desired change. For example, in the scenario presented above, for the virtual coach JOGG to propose a possible activity to the user, it will employ services to determine the possible activities, to adapt its emotional expression (which calls a service to determine the user's current emotional state), to decide on the most appropriate way to propose those activities to the user (based on the user's culture, personality, and on holding norms of behaviour), and so on.

Quality of Service MaaS systems have different requirements concerning the socio-cognitive capabilities needed and the desired level of realism. Although all aspects will play a role in both decisions their relative importance is different. E.g. decisions on buying more organic products in the supermarket are mostly based on culture and personality, while and decisions buying cars might be more status driven. This characteristic demands design models that are scalable and can be flexibly adapted to the varying requirements of quality and scale of different use-cases.

The service-oriented approach taken in MindBuilder enables to specify and select services with different levels of precision and computational complexity to execute similar functionality. I.e., depending on the specific demands of an application domain, a socio-cognitive service for normative reasoning can, for example, be based on a temporal-deontic logic [34] or on Ostrom's 'ADICO' model [10]. Given the use of uniform Service Interface Descriptors, one can be replaced by the other in a MaaS system, resulting in a more or less rich normative reasoning as indicated by requirements of a given domain.

The quality of a service is described by the service's capability to handle fluents. That is, differences in quality of services are related to which fluents can be handled by the service, and how those fluents are perceived. Assuming an expressive domain representation language, many details can be given about a situation, however not all services are able to handle all the details. This results in different levels of complexity and realism for interchangeable services. Consider for instance, services that analyse the emotion of an user. Rich services can take into account vision, audio and biologic sensor information, while a simple emotion service is only able to take into account input from a dropbox question to the user ("How are you feeling? Choose from the following X options"). Obviously, the result of different emotion-services will be more or less detailed depending on the service option used. However, not all applications require the richer version.

4.2 Deliberation Bus

It is well-known that neither purely reactive nor purely deliberative techniques are capable of producing the range of behaviours required of intelligent agents in dynamic, unpredictable, domains, and specially when interaction with people is needed. I.e. real-time interaction requires both extensive reasoning as well as fast reaction. Therefore, socio-cognitive services have different expectations in terms of time and reaction rate, which demands the integration of goal-based planning and reaction over diverse temporal and functional scopes. At the heart of a MaaS we propose a Deliberation Bus consisting of a central deliberation bus to connnect and synchronise different services, and of memory and time management units. Besides socio-cognitive services, the Deliberation Bus also links to sensing and actuator services. These are dependent on the actual system or artefact in which the MaaS is embedded. The Deliberation Synchronisation Bus specifies and implements the communication between services using SID, and takes care of the synchronisation of the different service processes. We use the

term 'bus' to stress the fact that we do not assume a fixed deliberation cycle but rather parallel communication between services depending on the situation. In order to allow a uniform quantization of time throughout the model, yet permit different rates of reaction for services, it becomes necessary to interleave sensing and planning. The time management unit allows multiple state updates to occur during deliberation, while keeping in synch with an evolving world. The Delib-

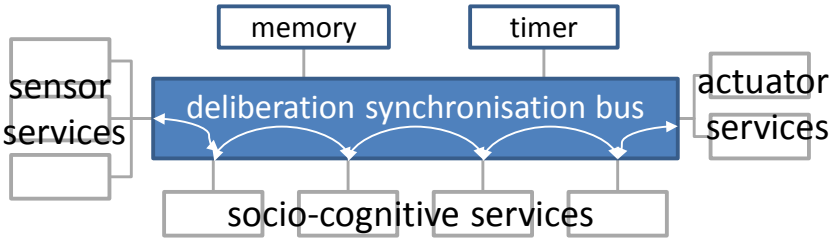


Fig. 3. Abstract Deliberation Bus Architecture

eration Bus architecture (cf. Figure 3) integrates deliberation and reaction in flexible and efficient ways. Existing deliberation paradigms such as goal-oriented (BDI) or reactive (Sense-Plan-Act) can be represented in the Bus, which is however expressive enough to specify many other deliberation possibilities.

4.3 Service meta-models and verification

Deep theoretical understanding of specific functionalities for social interaction is a pre-requisite to their use in artificial social intelligent systems, yet there is an awareness that current formalisms are not able to deal with the representation of social functionalities and their interrelations in a way that enables verification and proof. Nevertheless, formalisms abound that deal with specific aspects of reasoning, such as decision-making, norms, or emotions. However, such models are quite disparate and integration is not well understood. Our proposal is to start from existing logical formalisms to represent and reason about social-cognitive behaviour and develop formal interpretations of existing social science theories of social behaviour.

There is a long tradition in AI to use logical theories to provide insights into the reasoning problem without directly informing the implementation. The use of logical formalisms as a tool of analysis and knowledge representation, is at the basis of AI research [25]. We will use existing formalisms for different aspects of social behaviour (emotions, norms, culture, personality, ...) as a basis to develop formal theory and algorithms to specify social intelligent systems in a compositional way integrating different theoretical formalisms for socio-cognitive behaviour. To enable the integration and combination of different models we are exploring a meta-modelling approach.

Model checking is a well-known technique to verify properties of a formal model. An attractive feature of model checking is that it can be used to identify behaviours in which the properties do not hold, potentially generating insight in how certain problems can be solved. Well-known limitations of model checking include its inappropriateness to deal with infinite state spaces and branching/alternative time, and it enables only the verification of the model and not validation the process used to transform social science theories into the formal representation. Moreover, the main challenge in model checking is the state explosion problem that can occur if the system being verified has components that make transitions in parallel. However, the scale and complexity of the formalizations that are required for social behaviour are reaching beyond the traditional techniques of philosophical logic. We will explore the combination of logical methods with simulation models to enable the development of a more comprehensive and adequate theory of practical social reasoning than what pure logic can achieve. Simulation results can identify 'interesting' situations that can subsequently be formally checked by model checkers or theorem provers to verify whether the system satisfies certain desired (formal) properties. Simulations produce possible behaviours of the system, which enable to understand the meaning of the abstractions and see whether it corresponds to the system requirements.

5 Conclusions

In this paper, we introduced the *'Mind as a Service'* (MaaS) architecture. Inspired by SOA, we propose to build social intelligent agents as a composition of software services, each designed to implement a specific socio-cognitive functionality. We are at the initial stages of this research, which we are sure has the potential to realise a new paradigm for agents. This paper aims to highlight the main features and challenges of MaaS. We are currently developing a software environment to build and deploy social minds. This platform, MindBuilder, enables the specification, composition, simulation and reuse of MaaS, and provides functionalities for *a*) Design: design services constructed using meta-models based on those formal representations using a uniform interface structure; *b*) Composition: specify Deliberation Bus models to compose services into MaaS systems with different deliberation models; *c*) Simulation: simulate and verify the behaviour of those MaaS systems; *d*) Library: provides library capabilities to store and search for services. The impact of the resulting systems on the people interacting with them is potentially very high. It is therefore crucial to consider the ethical impact of social intelligent systems. We believe that realistic technical solutions are needed before we can fully address the moral and ethical issues inherent to artificial systems that provide care, change behaviour, and interact with vulnerable people across all age-groups. User participation and near-realistic experimentation environments are needed to explore and evaluate technical results and their ethical consequences in a controlled non-evasive way.

References

1. R. Adolphs. Social cognition and the human brain. *Trends in cognitive sciences*, 3(12):469–479, 1999.
2. C. Allen, W. Wallach, and I. Smit. Why machine ethics? *IEEE Intelligent Systems*, 21(4):12–17, 2006.
3. J. R. Anderson, M. Matessa, and C. Lebiere. Act-r: A theory of higher level cognition and its relation to visual attention. *Hum.-Comput. Interact.*, 12(4):439–462, Dec. 1997.
4. R. Aras and A. Dutech. An investigation into mathematical programming for finite horizon decentralized pomdps. *J. Artif. Int. Res.*, 37(1):329–396, Jan. 2010.
5. R. Beheshti. Normative agents for real-world scenarios. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1749–1750, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.
6. R. E. Boyatzis. Learning life skills of emotional and social intelligence competencies. *The Oxford Handbook of Lifelong Learning*, page 91, 2011.
7. C. Breazeal. Social interactions in hri: The robot view. *Trans. Sys. Man Cyber Part C*, 34(2):181–186, 2004.
8. R. A. Brooks. Cognitive simulators. In *Architectures for Intelligence: The 22nd Carnegie Mellon Symposium on Cognition*. Psychology Press, 2014.
9. A. J. Butt, N. A. Butt, A. Mazhar, Z. Khattak, and J. A. Sheikh. The soar of cognitive architectures. In *CTIT 2013*, pages 135–142. IEEE, 2013.
10. S. E. Crawford and E. Ostrom. A grammar of institutions. *American Political Science Review*, 89(03):582–600, 1995.
11. J. Dias, S. Mascarenhas, and A. Paiva. Fatima modular: Towards an agent architecture with a generic appraisal framework. In *WS. Standards for Emotion Modeling*, 2011.
12. F. Dignum, R. Prada, and G. J. Hofstede. From autistic to social agents. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14*, pages 1161–1164. IFAAMAS, 2014.
13. M. Dragone, H. Jordan, D. Lillis, and R. W. Collier. Separation of concerns in hybrid component and agent systems. *International Journal of Communication Networks and Distributed Systems*, 6(2):176–201, 2011.
14. T. Erl. *Soa: principles of service design*, volume 1. Prentice Hall Upper Saddle River, 2008.
15. S. T. Fiske and S. E. Taylor. *Social cognition: From brains to culture*. Sage, 2013.
16. D. Goleman. Social intelligence: The new science of social relationships. *New York, NY: Bantam*, 2006.
17. D. Kahneman and D. T. Miller. Norm theory: Comparing reality to its alternatives. *Psychological review*, 93(2):136, 1986.
18. G. A. Kaminka. Curing robot autism: A challenge. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*. IFAAMAS, 2013.
19. S. Kent. Model driven engineering. In *Integrated formal methods*, pages 286–298. Springer, 2002.
20. J. Laird. *The Soar cognitive architecture*. MIT Press, 2012.
21. M. Mancini and C. Pelachaud. Dynamic behavior qualifiers for conversational agents. In *Intelligent Virtual Agents*, pages 112–124. Springer, 2007.
22. J. G. March. *Primer on decision making: How decisions happen*. Simon and Schuster, 1994.

23. M. Minsky. *The Society of Mind*. Simon & Schuster, Inc., New York, NY, USA, 1986.
24. M. Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.
25. R. C. Moore. *Logic and representation*, volume 39. Center for the Study of Language (CSLI), 1995.
26. A. Ortony. *The cognitive structure of emotions*. Cambridge university press, 1990.
27. A. Schutz. *The phenomenology of the social world*. Northwestern University Press, 1967.
28. O. M. Shehory and A. Sturm. *Agent-oriented software engineering: reflections on architectures, methodologies, languages, and frameworks*. Springer, 2014.
29. B. G. Silverman, M. Johns, J. Cornwell, and K. O'Brien. Human behavior models for agents in simulators and games: part i: enabling science with pmfserv. *Presence: Teleoperators and Virtual Environments*, 15(2):139–162, 2006.
30. J. T. Turner, S. N. Givigi, and A. Beaulieu. Implementation of a subsumption based architecture using model-driven development. In *Systems Conference (SysCon), 2013 IEEE International*, pages 331–338. IEEE, 2013.
31. J. van Oijen, L. Vanhée, and F. Dignum. Ciga: A middleware for intelligent agents in virtual environments. In *Agents for Educational Games and Simulations*, pages 22–37. Springer, 2012.
32. J. Wagner, F. Lingenfelser, T. Baur, I. Damian, F. Kistler, and E. André. The social signal interpretation framework: multimodal signal processing and recognition in real-time. In *21st ACM Conf. Multimedia*, pages 831–834. ACM, 2013.
33. G. Weiss. *Multiagent Systems*. MIT Press, 2013.
34. R. Wieringa and J.-J. Meyer. Applications of deontic logic in computer science: A concise overview. In *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. John Wiley & Sons, 1993.

CÒIR : Verifying Normative Specifications of Complex Systems.

Luca Gasparini[1], Timothy J. Norman[1], Martin J. Kollingbaum[1], Liang Chen[1], and John-Jules Ch. Meyer[2]

[1] Department of Computing Science, University of Aberdeen, UK
{l.gasparini, t.j.norman, m.j.kollingbaum, l.chen}@abdn.ac.uk
[2] Information and Computing Sciences, Utrecht University, The Netherlands
j.j.c.meyer@uu.nl

Abstract. Existing approaches for the verification of normative systems consider limited representations of norms, often neglecting collective imperatives, deadlines and contrary-to-duty obligations. In order to capture the requirements of real-world scenarios, these structures are important. In this paper we propose methods for the specification and formal verification of complex normative systems that include contrary-to-duty, collective and event-driven imperatives with deadlines. We propose an operational syntax and semantics for the specification of such systems. Using Maude and its linear temporal logic model checker, we show how important properties can be verified for such systems, and provide some experimental results for both bounded and unbounded verification.

1 Introduction

The specification and verification of properties of normative systems is an important consideration for the design of complex distributed systems [1, 5]. Motivated by the need to capture the requirements of real world scenarios, research on the specification of normative systems has explored conditional [15], event-governed (e.g. activation/expiration condition) norms [13], collective imperatives [8, 11], imperatives with deadlines [6], and contrary-to-duty (CTD) norms [15]. A further focus has explored mechanisms for the analysis of systems of norms for the purpose of identifying and resolving conflicts between norms and plans [16]. Although such analyses are of benefit, for safety critical systems it is important to analyse the interactions between normative constraints and agents' actions as a system evolves. For these reasons the use of model checking [3] techniques to analyse liveness and safety properties of norm-governed systems has been explored [1, 5, 7]. To date, however, this research has focussed on restricted representations of norms such as labelling states or transitions as compliant/non-compliant. Ågotnes *et al.* [1], for example, study the complexity of this model checking problem for different robustness-related properties; e.g. whether a certain property is guaranteed in the event of a subset of agents violating a norm.

The focus of this paper is on how to efficiently apply model checking to analyse properties of normative systems specifications with richer representations of

norms. In particular, we consider event-governed conditional norms, deadlines for the fulfilment of obligations, and contrary to duty and group imperatives. The contributions we claim are as follows: (i) We propose a norm specification language that is sufficiently expressive to capture all the features discussed above, namely CÒIR[3]; (ii) a Structural Operational Semantics (SOS) [12] for a monitoring component that, given a description of the environment, keeps track of activation, expiration, fulfilment, and violations of norms; and (iii) a realisation of this component using the Maude [4] rewriting logic framework, which allows us to perform formal analysis of normative systems specifications. A particular challenge is that representing time explicitly (in order to reason about temporal deadlines) makes the problem undecidable. For these reasons we explore both the use of bounded model checking and model abstraction to obtain a finite Kripke structure for unbounded model checking. We present some results of both these approaches in an example domain that motivates the requirements for us considering such a rich representation of norms.

2 Motivating Example

Consider a coalition of agents of the sea-guard, consisting of a set of *Unmanned aerial vehicles (UAVs)*, *helicopters*, and *boats*. Their goal is to monitor and intercept unauthorized boats trying to access a restricted area. The norms that guide the behaviour of the coalition are: (**1**) At any moment at least one member of the coalition must monitor the area. Moreover, we prefer having UAVs monitoring the area over helicopters. We assume that only helicopters and UAVs are capable of monitoring. (**2**) Whenever an unauthorized boat enters the area, a member of the coalition must intercept it before a certain deadline expires. (**3**) If no one intercepts the boat, then at least one member of the coalition must send a report to head-quarters before a certain deadline expires. These are all examples of collective imperatives: they require *at least one member* of the coalition to act. Norm 3 is also a CTD obligation that is activated in the event of a violation of the obligation 2. Moreover, norms 2 and 3 require the agents to perform an action *before a certain deadline* (a liveness property), while norm 1 requires that *at any given moment* someone is monitoring the area (a safety property).

3 CÒIR Norm Specification

We now introduce a formalism for representing norms that satisfies our requirements, which we call CÒIR. We allow for the definition of *obligations with deadlines* and *prohibitions* and we assume that everything that is not prohibited is permitted. Compliance with norms is evaluated against a knowledge base KB that is dynamically updated to represent the environment and the observable properties of the agents acting within it. We rely on the closed-world assumption, which we believe to be reasonable in a verification setting. We include

[3] CÒIR is the Scottish Gaelic for obligation.

```
⟨constTerm⟩     = ⟨strTerm⟩ | ⟨numTerm⟩ ;
⟨term⟩          = ⟨constTerm⟩ | ⟨varTerm⟩ | ⟨predicate⟩ ;
⟨predicate⟩     = ⟨functor⟩ "(" ⟨term⟩ { "," ⟨term⟩ } ")" ;
⟨binding⟩       = "BIND( " ⟨varTerm⟩ "," ⟨term⟩ ")" ;
⟨violationCond⟩ = "VIOLATION-OF(" ⟨numTerm⟩ "," ⟨strTerm⟩ ")" ;
⟨formula⟩       = ⟨predicate⟩ | ⟨binding⟩ | ⟨violationCond⟩ | ⟨formula⟩ "/\" ⟨formula⟩
                | ⟨formula⟩ "\/" ⟨formula⟩  | "IN {" ⟨formula⟩ "} FILTER" ⟨boolExpr⟩ ;
⟨constant⟩      = "false" | "true" ;
⟨existsPattern⟩ = "EXISTS {" ⟨formula⟩ "}" ;
⟨equalsCond⟩    = "EQUALS (" ⟨term⟩ "," ⟨term⟩ ")" ;
⟨temporal⟩      = "TEMPORAL (" ⟨numTerm⟩ ")" ;
⟨violated⟩      = "VIOLATED" ;
⟨compareOp⟩     = "=" | ">" | "<" | "<=" | ">=" ;
⟨count⟩         = "COUNT(" ⟨varTerm⟩ "IN{" ⟨formula⟩ "})" ⟨compareOp⟩⟨numTerm⟩ ;
⟨boolExpr⟩      = "NOT" ⟨boolExpr⟩ | ⟨violated⟩ |  ⟨count⟩ | ⟨boolExpr⟩ "/\" ⟨boolExpr⟩
                | ⟨boolExpr⟩ "\/" ⟨boolExpr⟩ | ⟨existsPattern⟩ | ⟨equalsCond⟩ | ⟨constant⟩ ;
```

Fig. 1. EBNF grammar for the COIR language.

the description of previous violations in the knowledge base. These can then be used to activate CTD norms. An issue that has been discussed, for example, by Dignum *et al.* [6] is whether an obligation with a deadline should persist or be deactivated after a violation; i.e. after the deadline has expired without the obligation being fulfilled. CÒIR supports the specification of either of these alternatives. By default obligations don't expire when violated, but, thanks to the fact that violations are represented in KB, it is always possible to specify the expiration condition as being triggered by a violation of the current instance.

3.1 Syntax

A norm `nd`i is defined as a tuple $\langle id_i, mod_i, act_i, exp_i, goal_i, ddl_i \rangle$ where: id_i is a unique identifier; $mod_i \in \{O, F\}$ specifies whether the norm is an obligation with deadline or a prohibition; act_i (activation condition) describes a pattern that, when matched in KB, causes a norm instance to be detached; $goal_i$ represents the situation that needs to be brought about (for an obligation) or avoided (for a prohibition); exp_i (expiration condition) is a condition that, when met, causes the expiration of the instance; and the deadline for the fulfilment of the norm (ddl_i) can be temporal or symbolic and is defined only for obligations.

Fig. 1 shows the EBNF grammar of the operational language used to represent the components of a norm specification. `functor` and `strTerm` are identified by strings that start with a letter, `numTerm` by numbers and `varTerm` by strings that start with a `?` character. `?actTime`, `?violTime`, `?tick`, `?this-id`, `?violated` and `?flag` are reserved terms. The description of the environment, KB, consists of a set of ground predicates; i.e. predicates with no `varTerm`. Intuitively, a `boolExpr` represents a condition that is evaluated against KB returning a boolean result, while a `formula` is a pattern with a set of variables that is evaluated by returning the set of substitutions that make the pattern match a subset of KB. In a norm description, act_i is represented by a `formula`, while exp_i, $goal_i$, and ddl_i are `boolExpr`s.

The formula `VIOLATION-OF(`n, s`)` is matched when there is a violation of norm n and is used for the activation of CTD obligations. The meaning of the pa-

rameter s will be explained in Section 4. The meanings of EQUALS, EXISTS and the usual boolean operators are intuitive. TEMPORAL(n) is evaluated to true if a temporal deadline has expired, while VIOLATED can be used in exp_i and returns true if the instance being evaluated has been violated. COUNT (v IN { f }) > n evaluates to true if the number of different assignments of the variable v that matches the pattern f is higher than the number n.

3.2 Representing collective obligations

We now discuss how our formalism allows us to represent different types of collective obligations [11]. In contrast to Tinnemeier *et al.* [14], we allow $goal_i$, exp_i, and ddl_i to include variables that have not been bound at activation time. Through the use of the patterns EXISTS{f_i} and NOT EXISTS{f_i} we are able to express existential and universal quantification on these variables. Inspired by Norman and Reed [11] we discuss some common patterns of collective obligations and show how they can be expressed in our language (See [8, 11] for discussions of responsibility in collective obligations). In order to ease the presentation, we assume that agents are organized in groups, group membership is represented by predicates of the type `memberOf(agent,group)`, and an agent's performance of an action by `perform(agent,action)`.

Joint distributive obligations are obligations where all the members of group `g` are responsible for all the members of the group performing the action `a`. This can be expressed by an obligation where:

$$act_i = \texttt{memberOf(?add,g)}$$
$$goal_i = \texttt{NOT EXISTS \{IN \{memberOf(?ag,g)\} FILTER NOT EXISTS \{perform(?ag,a)\}\}}$$

$goal_i$ is met when there is no member of `g` that hasn't performed `t`; i.e. when all the members of `g` have performed `a`. As a result, if any of the members of the group don't perform the task, all the members will be responsible for the violation. Alternatively we could consider the group as an entity to be responsible for the fulfilment of the obligation by specifying the activation condition as:

$$act_i = \texttt{BIND (?add , g)}$$

and referring to the group as `?add` in the goal. Note that if a group has no members, such an obligation would be trivially fulfilled. It might be appropriate to add the constraint EXISTS{memberOf(?add,g)} in act_i or in $goal_i$.

Joint collective obligations specify that all the members of a group `g` are responsible for at least one member of the group performing the action `a`.

$$act_i = \texttt{memberOf(?add,g)}$$
$$goal_i = \texttt{EXISTS\{ memberOf(?ag,g) /\ perform(?ag,a)\}}$$

4 CÒIR Semantics

We define the semantics of CÒIR through a Structural Operational Semantics (SOS), a framework for the description of the semantics of programming and

specification languages. SOS consists of a set of transition rules that generate a transition system whose states are called configurations. Transition rules are of the form $\frac{P}{C \to C'}$ meaning that, whenever P holds, a transition from the configuration C to C' is applicable. We use SOS to describe how the active norm instances and violations are updated every time we detect a change in KB.

In formalising these semantics we assume two functions that evaluate `formula` and `boolExpr`; these will be summarised below. We define a substitution $\theta_j \in \Theta$ as a set of assignments $[v/c]$ where c is a `constTerm` and v a `varTerm`. Formulae are evaluated by means of a function $match : 2^P \times Q \to 2^\Theta$, where P is the set of all predicates, Q the set of all formulae, and Θ the set of all substitutions. Intuitively, $match(KB, f)$ returns all the substitutions θ_i such that $f \cdot \theta_i$ is entailed by KB. Boolean expressions (`boolExpr`) are evaluated by means of a function $eval : 2^P \times \Theta \times E \to bool$ where E is the set of all `boolExpr` and $bool \in \{$ `true` , `false` $\}$. A norm instance $[id_i, \theta_j, at]$ is detached at time at for each substitution $\theta_j \in match(KB, act_i)$. Then $eval(KB, e, \theta_j)$ is used to evaluate exp_i, $goal_i$, and ddl_i. The addressee of the norm, identified by the value assigned to `?add` in θ_j, is responsible for complying with the obligation (reaching a state where $eval(KB, goal_i, \theta_j) =$ `true` before the deadline) or with the prohibition (avoiding states where $eval(KB, goal_i, \theta_j) =$ `true` until the prohibition expires).

A further issue to address prior to detailing the transition rules of our operational semantics is that of "duplicate activations". Consider a simplified version of norm 3 from Section 2. We specify its activation condition as follows:

```
type(?add,coalition) /\ type(?boat,unBoat) /\ type(?area,rArea) /\ inArea(?boat,?area)
```

In other words, an instance of the obligation to send a report should be detached when an unauthorized boat is in the restricted area. Intuitively, if the same boat remains in the restricted area for more than one consecutive instant of time, we don't want the coalition members to send more than one report. However, if the boat exits and then re-enters the area, we would expect the coalition to be obliged to send another report. Formally, if we denote by KB_t the state of the knowledge base at time t, we capture this distinction by activating an instance of a norm `ndi`, associated with a substitution θ_j, at an instant of time t whenever $\theta_j \in match(KB_t, act_i)$ and $\theta_j \notin match(KB_{t-1}, act_i)$; i.e. when we find a substitution such that act_i goes from "unmatched" to "matched" in two subsequent instants of time. To do that we keep record of the instances $[id_i, \theta_j, at]$ such that the act_i was matched in the previous instant of time.

Following Dennis et al. [5], in order to enforce an order of execution among the transitions of the operational semantics, we organize the reasoning cycle in three stages: (**A**) Deactivate instances for which the expiration condition holds or the obligation has been fulfilled; (**B**) Check for violations of active obligations (if the deadline has passed, but the goal has not been achieved) and prohibitions (if the state to avoid is achieved).(**C**) Check for the activation of new norms and update the list of previously matched instances.

In the following we denote by $a_1 : a_2 : \ldots$ a list of elements and we use ϵ to indicate the end of a list. Moreover, we assume that KB contains a predicate `cT(n)`, where `n` is a `numTerm` that represents the current time of the system and

we denote by $time(KB)$ the value $\mathtt{n}$ such that $\mathtt{cT(n)} \in KB$. A configuration $Conf$ is defined as $\langle KB, \Delta, I, \Pi, \Phi, \Sigma, r\rangle$ where KB is the current state of the knowledge base, Δ is a list of norm descriptions, I is the list of active norm instances and Π the list of previously matched instances, which, as discussed above, is needed to avoid the problem of multiple activations. Φ is the set of violations detected in the current reasoning cycle[4], and a violation is represented as $v = [id_i, \theta, t]$, where t corresponds to the violation time. Σ is the stage of the computation and r is a flag that is set initially to `false`, and changed to `true` if we need to loop again through the reasoning cycle. This is necessary because, whenever we activate a new instance (stage **C**), we need to check whether this is instantly fulfilled or violated (**A** and **B**). Moreover, detecting a violation (**B**) could trigger an expiration or an activation (**A** and **C**).

The initial configuration is $\langle KB_0, \Delta, \epsilon, \epsilon, \epsilon, \mathbf{A}, \mathtt{false}\rangle$, where KB_0 describes the initial state and Δ the normative specification. We now illustrate the key rules of the operational semantics. For each rule we include only the components of the configuration that are involved in it.

Rule R1 applies when the first instance in I is such that its expiration condition holds. In this case we simply remove the instance from the list. Similarly another rule (not included) is defined for the case of a fulfilled obligation. Rule R2 accounts for the case where the first instance in the list is a prohibition and the expiration condition is not met. In this case we move the instance to the end of the list, after the ϵ symbol. We write a similar rule (not included) for an obligation instance that it is neither fulfilled nor expired. Rule R3 represents the end of stage A, which occurs when the first instance is ϵ.

$$\frac{\langle KB, \Delta, [id_i, \theta_j, at] : I, \mathbf{A}\rangle,\ nd_i \in \Delta,\ eval(KB, exp_i, \theta_j) = \mathtt{true}}{\langle KB, \Delta, [id_i, \theta_j, at] : I, \mathbf{A}\rangle \rightarrow \langle KB, \Delta, I, \mathbf{A}\rangle} \tag{R1}$$

$$\frac{nd_i \in \Delta,\ mod_i = F,\ eval(KB, exp_i, \theta_j) = \mathtt{false}}{\langle KB, \Delta, [id_i, \theta_j, at] : I, \mathbf{A}\rangle \rightarrow \langle KB, \Delta, I : [id_i, \theta_j, at], \mathbf{A}\rangle} \tag{R2}$$

$$\frac{\mathtt{true}}{\langle \epsilon : I, \mathbf{A}\rangle \rightarrow \langle I : \epsilon, \mathbf{B}\rangle} \tag{R3}$$

Rule R4 detects violated obligations; i.e. obligations whose deadline has expired before the goal is satisfied. Since fulfilled obligations have been deleted in stage **A**, we just need to check whether the deadline has expired. When we detect a violation we update the violations list, add the violation description (denoted by $d([id_i, \theta_j, \tau])$) to KB and we set the flag r to `true` since the violation predicate might trigger the expiration condition of that instance. $d([id_i, \theta_j, \tau])$ consists of a predicate $\mathtt{v}(id_i,\mathtt{p}(\theta_j),\tau)$ where $\mathtt{p}(\theta_j)$ is a representation of the substitution in the form of a predicate. In rule R5 if the first obligation in the list is not violated we move it at the end of the list. Similarly we add two rules (not included) for prohibitions, where we consider a prohibition to be violated if its goal condition evaluates to `true`. Φ is included to avoid infinite loops. In fact, since rule R4 sets r to `true`, detecting the same violation in each loop would cause infinite iteration.

[4] We refer to the whole updating procedure as a reasoning cycle, while **A**, **B** and **C** are the stages of a cycle.

Rule R6, together with the condition $[id_i, \theta_j, \tau] \notin \Phi$ of rule R4 ensures that each violation is detected only once for each reasoning cycle. Another rule similar to rule R3 (not included) is defined for the end of stage **B**.

$$\frac{\begin{array}{c} mod_i = O,\ [id_i, \theta_j, \tau] \notin \Phi, \\ eval(KB, ddl_i, \theta_j) = \texttt{true},\ KB^* = KB \cup d([id_i, \theta_j, \tau]) \end{array}}{\langle KB, \Delta, [id_i, \theta_j, at] : I, \Phi, \mathbf{B}, r\rangle \rightarrow \langle KB^*, \Delta, I : [id_i, \theta_j, at], [id_i, \theta_j, \tau] : \Phi, \mathbf{B}, \texttt{true}\rangle} \quad \text{(R4)}$$

$$\frac{nd.mod = O, eval(KB, ddl_i, \theta_j) = \texttt{false}}{\langle KB, \Delta, [id_i, \theta_j, at] : I, \mathbf{B}\rangle \rightarrow \langle KB, \Delta, I : [id_i, \theta_j, at], \mathbf{B}\rangle} \quad \text{(R5)}$$

$$\frac{[id_i, \theta_j, \tau] \in \Phi}{\langle [id_i, \theta_j, at] : I, \Phi, \mathbf{B}\rangle \rightarrow \langle I : [id_i, \theta_j, at], \Phi, \mathbf{B}\rangle} \quad \text{(R6)}$$

Rule R7 checks for the activation of new instances of the first norm nd_i in Δ. Let $\tau = time(KB)$, for each $\theta_j \in match(KB, act_i)$, we add a new instance $[id_i, \theta_j, \tau]$ at the end of Π (list Π_2), while we add to I only those instances that are not in Π (list I_2). The substitutions of the instances added to I_2 are integrated with the assignment of the variables `?actTime` and `?this-id` which are needed to evaluate the TEMPORAL and the VIOLATED conditions as we will show below. If we activate at least one new instance we set $r = \texttt{true}$. By adding new instances at the end of Π, we ensure that, at the end of the reasoning process, the instances added to Π during the current reasoning cycle will be those after ϵ. Formally the pattern $\Pi_3 : \epsilon : \Pi_4$ identifies with Π_4 all the instances added in the current step and with Π_3 all the instances added during the previous reasoning cycle. This is exploited in rule R8, where, at the end of stage **C**, if r is equal to `false`, we end the reasoning cycle (stage **end**) and discard Π_3 and Φ. We define another rule (not included) for the case where r is equal to `true`. In this case we move ϵ at the end of Δ and go back to stage **A**. In rule R7, when we check if a new instance is not in Π, we consider also instances added in previous loops of the current reasoning cycle. In this way it is guaranteed that we do not reactivate the same instances in each loop.

$$\frac{\begin{array}{c} I_2 = \langle [id_i, (\theta_j \cup \theta_k), \tau] : \ldots\rangle \ s.t.\ \theta_j \in match(KB, act_i) \text{ and } eval(KB, exp_i, \theta_j) = \texttt{false} \\ \text{and } \theta_k = [\texttt{?actTime}/\tau] \cup [\texttt{?this-id}/id_i] \text{ and } [id_i, \theta_j, \tau - 1] \notin \Pi, \\ \Pi_2 = \langle [id_i, \theta_j, \tau] : \ldots\rangle \ s.t.\ \theta_j \in match(KB, act_i),\ r* = \texttt{true} \text{ iff } (I_2 \neq \emptyset) \text{ or } (r = \texttt{true}) \end{array}}{\langle KB, nd_i : \Delta, I, \Pi, \mathbf{C}, r\rangle \rightarrow \langle KB, \Delta : nd_i, I_2 : I, \Pi : \Pi_2, \mathbf{C}, r*\rangle} \quad \text{(R7)}$$

$$\frac{\texttt{true}}{\langle \epsilon : \Delta, \Pi_3 : \epsilon : \Pi_4, \Phi, \mathbf{C}, \texttt{false}\rangle \rightarrow \langle \Delta : \epsilon, \Pi_4 : \epsilon, \epsilon, \mathbf{end}, \texttt{false}\rangle} \quad \text{(R8)}$$

With these transition rules in place, we now provide further details of the $match$ and $eval$ functions for querying KB. We denote by $\theta_i[v]$ the value c assigned by θ_i to the variable v. Given a formula f, $f \cdot \theta_i$ denotes the `formula` obtained by substituting, for each `varTerm` v with an assignment in θ_i, each occurrence of v in f with $\theta_i[v]$. Moreover we say that two substitutions θ_1 and θ_2 are compatible if and only if there is no variable v that is bound in both the substitutions such that its assigned values are different. Formally:

$$compatible(\theta_1, \theta_2) = \texttt{true} \text{ iff } \nexists\ v,\ ([v/c_1] \in \theta_1 \text{ and } [v/c_2] \in \theta_2 \text{ and } c_2 \neq c_1)$$

Let p denote a `predicate`, e a `boolExpr`, f_i a `formula`, v_i a `varTerm`, n and a `numTerm`, s_i a `strTerm` and t a `constTerm`. We denote by $s_1.\theta_k$ the substitution obtained by adding the string s_1 as a prefix to all `varTerm`s in θ_k. Fig. 2 summarizes the semantics of $match$ and $eval$.

$match(KB, p) = \{\theta_i : p \cdot \theta_i \in KB\}$
$match(KB,$ `BIND ( ` v_1 `, ` t `)` $) = \{[v_1/t]\}$
$match(KB, f_1$ `/\` $f_2) = \{(\theta_1 \cup \theta_2) : \theta_1 \in match(KB, f_1)$
and $\theta_2 \in match(KB, f_2)$ and $compatible(\theta_1, \theta_2)\}$
$match(KB, f_1$ `\/` $f_2) = match(KB, f_1) \cup match(KB, f_2)$
$match(KB,$ `VIOLATION-OF(`t_1, s_1`)`$) = \{s_1.(\theta_j \cup$ `[?violTime/`vt`]`$) : d($`[`t_1, θ_j, vt`]`$) \in KB\}$
$eval(KB,$ `EXISTS { ` f_1 `}`$, \theta_i) =$ `false` if $match(KB, f_1 \cdot \theta_i) = \emptyset$; `true` otherwise
$eval(KB,$ `EQUALS ( ` t_1 `, ` t_2 `)`$, \theta_i) =$ `true` iff $t_1 = t_2$. If t_1 or t_2 are `varTerm`, θ_i is used to replace them with their assigned constant terms.
$eval(KB,$ `VIOLATED`$, \theta_i) =$ `true` iff there exists a vt such that $d($`[`t_1, θ_i, vt`]`$) \in KB$ and θ_i`[?thisId]` $= t_1$
$eval(KB,$ `TEMPORAL(`n`)`$, \theta_i) =$ `true` iff θ_i`[?actTime]` $+ n <= time(KB)$
$eval(KB,$ `COUNT(`v_1 `IN{`f_1`}) > `$n, \theta_i) =$ `true` iff $|\{\theta_j[v_1] : \theta_j \in match(KB, f_1 \cdot \theta_i)\}| > n$.
same for the other `compareOp`.
`NOT`, `\/`, and `/\` have the usual meaning when applied to `boolExpr`
$match(KB,$ `IN { ` f_1 `} FILTER ` $e) = \{\theta_i : \theta_i \in match(KB, f_1)$and $eval(KB, e, \theta_i) =$ `true`$\}$.

Fig. 2. Semantics of *match* and *eval*

The construct `TEMPORAL(n)`, where `n` is a `numTerm`, will be used to evaluate a temporal deadline of `n` steps relative to the activation time of a norm instance. In defining its semantics we assume that the variable `?actTime` is bound in θ_j to the activation time (see Rule R7 of above). The construct `VIOLATION-OF(`n, s`)` presented in Section 3.1, can be used in the activation condition of a CTD norm to return the description of a detected violation of a norm with id n. For a violation $[id_j, \theta_j]$, with $n = id_j$, it returns the substitution obtained by adding the prefix s to all the variable names of θ_j. The prefix is added in order to allow the norm designer to distinguish between variables bound by the substitution of the violation and variables bound by the activation condition, even when they have the same variable name. The construct `VIOLATED` is used when we want to ask whether the current instance has been violated (e.g. for the expiration condition of an obligation). It is evaluated to `true` if *KB* contains the description of a violation of the instance being evaluated. Note that, since, for each instance, we bind the activation time in the substitution, `VIOLATED` is able to distinguish between violations of different instances associated with the same pair $(\texttt{nd}j, \theta_j)$.

5 The *Seaguard* Example

We now show how we can capture the norms described in our motivating example (Section 2) using the CÒIR formalism. Norm 1 states that at any instant of time, at least one agent must monitor the area. This may be captured by a prohibition from achieving a state where no agent is monitoring the area (a safety property). The fact that a UAV monitoring the area is preferred to a helicopter can be represented by separating the norm in two as shown in Fig. 3 (`nd1` and `nd4`). Norm `nd1` is a prohibition that is violated if no UAV is monitoring the area. Norm `nd4` is violated if neither a UAV nor a helicopter is monitoring the area.

Therefore, a situation where a UAV is monitoring the area would comply with both the norms, while having a helicopter monitoring would violate only nd1.

```
nd1 = ⟨1, F, act1, false, goal1, false ⟩
   act1 = type(?add,coalition) /\ type(?ar,rArea)
   goal1 = NOT EXISTS{ memberOf(?ag1,?add) /\ type(?ag1,uav) /\ monitoring(?ag1,?ar) }
nd4 = ⟨4, F, act1, false, goal4, false ⟩
   goal4 = NOT EXISTS{ memberOf(?ag1,?add) /\ monitoring(?ag1,?ar) /\
      ( type(?ag1,uav) \/ type(?ag1,heli) ) }
nd2 = ⟨2, O, act2, exp2, goal2, ddl2⟩
   act2 = IN{ type(?add,coalition) /\ type(?ar,rArea) /\ inArea(?ag1,?ar) } FILTER
      NOT EXISTS{ type(?ag1,?type) /\ subType(?type,authAgent) }
   exp2 = VIOLATED \/ NOT EXISTS { inArea(?ag1,?ar) }
   goal2 = EXISTS{ intercepting(?ag2,?ag1) /\ memberOf(?ag2,?add) }
   ddl2  = TEMPORAL(3)
nd3 = ⟨3, O, act3, exp3, goal3, TEMPORAL(3) ⟩
   act3 = IN { type(?add,coalition) /\ VIOLATION-OF(2,v) } FILTER EQUALS(?add,?v:add)
   exp3 = NOT EXISTS{ inArea(?v:ag1,?v:ar) } \/ EXISTS { intercepting(?ag2,?v:ag1) }
   goal3 = EXISTS{ reporting(?ag2,?v:ag1) /\ memberOf(?ag2,?add) }
```

Fig. 3. Specification of norms nd1, nd2 and nd3.

Norms nd2 and nd3 capture the specification of norms 2 and 3 from our motivating example respectively. An instance of the obligation nd2 is activated, for a coalition, every time an unauthorized boat ?ag1 enters the restricted area ?ar. The obligation is fulfilled if one member of the coalition ?ag2 intercepts ?ag1 before a deadline of three time steps, while it expires if ?ag1 exits ?ar or the obligation is violated. Obligation nd3 is activated by a violation of norm nd2, and is addressed to the same coalition. It requires at least one member of the coalition to report the unauthorized access.

6 Formal Verification

In this section we explore the problem of verifying properties of multi-agent systems specified using CÒIR. Firstly we discuss our implementation of the operational semantics in Maude [4], a rewriting logic framework that allows us to specify the semantics of a system by means of rewriting rules. We chose Maude because its syntax for specifying rewriting rules is very close to that for SOS. Moreover, by implementing our system in Maude, we obtain a specification which is executable and on which we can perform formal verification using the Maude Linear Temporal Logic (LTL) model checker. In this way we can: (i) Validate our normative specification; for example by verifying that a specified non compliant behaviour always results in a detected violation; and (ii) Verify how robust a multi-agent system is to violations; for example by verifying if a certain property is guaranteed under certain compliance assumptions [1, 7].

We discuss the reasons why, by representing our model as explained in Section 4, we obtain an infinite state model. We show how we can use the LTL model checker to perform bounded model checking of the infinite state system, and then show how we can modify our model in order to make the state space finite and apply unbounded model checking.

6.1 Maude Implementation

Maude modules can contain *conditional equations*: simplification rules used to define data-types and language constructs and to specify how they are evaluated by the system. Modules may also contain *conditional rewriting rules*: transition rules that describe how the state of a system can evolve over time. We defined the CÒIR language (Fig. 1) and we implemented the $match$ and $eval$ functions. We then implemented our operational semantics by means of an operator `reason` that takes as arguments a configuration and returns the configuration resulting from the application of the reasoning cycle. The reasoning process is described by a set of conditional equations, which are a direct (syntactical) translation of the rules of Section 4 into the Maude syntax. The dynamics of the system is specified by a set of rules that follow the pattern:

```
crl C => reason( tick(C', n) ) if condition .
```

where `c` and `c'` are two configurations and the only component that can change from `c` to `c'` is the knowledge base. `tick` is a function that takes a configuration `c` and an integer `n` as parameters and increases the time in `c` by `n` units. The meaning of this rule pattern is that, at each step, after applying the changes in the description of the environment, we invoke the `reason` operator to update the list of active instances, previous matches and violations accordingly. The Maude model checker, given one initial state i, and a set of transition rules T, generates a Kripke structure containing all the states that are reachable from i.

6.2 Bounded Model Checking

Properties of a norm-governed multi-agent system can be verified using the Maude LTL model checker. In order to do so we need to define a labelling function λ, specifying the set of atomic propositions $q \in Q$ that hold in some state $s \in S$ [4, Chap. 13]. We denote by $((s \models_\lambda q) = \texttt{true})$ the fact q holds in s and by $((s \models_\lambda q) = \texttt{false})$ the fact that q doesn't hold in s. The state of a multi-agent system is represented by the configuration $Conf$ of the monitoring component. Let Q be the set of all `predicate`s as defined in Fig. 1. Equations 1-4 defines λ.

$$\langle KB, \Delta, I, \Pi, \Phi, \Sigma, r\rangle \models_\lambda p = \texttt{true} \text{ if } p \in KB. \tag{1}$$

$$\langle KB, \Delta, I, \Pi, \Phi, \Sigma, r\rangle \models_\lambda violated(n) = \texttt{true} \text{ if } \exists\ \theta_j, \tau \text{ s.t. : } d([n, \theta_j, \tau]) \in KB \tag{2}$$

$$\langle KB, \Delta, I, \Pi, \Phi, \Sigma, r\rangle \models_\lambda violated(n, t) = \texttt{true} \text{ if } \exists\ \theta_j, \tau \text{ s.t. : } d([n, \theta_j \cup [\texttt{?add/t}], \tau]) \in KB \tag{3}$$

$$\langle KB, \Delta, I, \Pi, \Phi, \Sigma, r\rangle \models_\lambda p = \texttt{false} \text{ otherwise .} \tag{4}$$

Equation 1 makes it possible to use the predicates of KB as atoms of LTL properties. Equations 2 and 3 define properties about the normative state of a configuration, allowing us to query the model checker for states where a certain norm has been violated (optionally specifying an addressee).

The principal requirement to make the LTL model-checking decidable is for the transition system to have a finite number of reachable states. However, the fact that we represent time explicitly in KB means that the state space is infinite.

One way of dealing with this is to limit the state space to the states reachable in a fixed number of transitions, l. We can do this, for example, by modifying the specification of the system so that all the conditional rewriting rules that increase the time by n are applicable only to states where $time(KB) < l - n$. Ideally, however, we want to be able to verify system properties in the unbounded case.

6.3 Unbounded Model Checking

In order to make the unbounded model checking problem decidable, we need to remove any explicit reference to the current time from the semantics. We remove the predicate `cT(n)` from KB and the references to activation and violation time from instances and violations respectively (now represented as $[id_j, \theta_k]$). In order to represent temporal deadlines, we take an approach similar to the one proposed by Lamport [10]. When we activate an instance (Rule R7), instead of binding `?actTime`, we add the assignment `[?tick/n]` in the substitution of instances of norms that include a statement of type `TEMPORAL(n)`. Rule R7 is substituted with:

$$\frac{\begin{array}{c} I_2 = \langle [id_i, (\theta_j \cup \theta_k)] : \ldots \rangle \; s.t. \; \theta_j \in match(KB, act_i) \text{ and } eval(KB, exp_i, \theta_j) = \texttt{false} \\ \text{and } \theta_k = isTemp(ddl_i) \text{ and } [id_i, \theta_j] \notin \Pi, \\ \Pi_2 = \langle [id_i, \theta_j] : \ldots \rangle \; s.t. \; \theta_j \in match(KB, act_i), \; r* = \texttt{true} \text{ iff } (I_2 \neq \emptyset) \text{ or } (r = \texttt{true}) \end{array}}{\langle KB, nd_i : \Delta, I, \Pi, \mathbf{C}, r \rangle \rightarrow \langle KB, \Delta : nd_i, I_2 : I, \Pi : \Pi_2, \mathbf{C}, r* \rangle} \quad \text{(R7*)}$$

where $isTemp(ddl_i)$ checks whether a deadline is temporal and, in that case, returns the initialisation for the `?tick` variable.

$$isTemp(ddl_i) = \begin{cases} \texttt{[?tick/t]} & \text{if } ddl_i \text{ contains one and only one statement of the type } \texttt{TEMPORAL(t)} \\ \emptyset & \text{otherwise.} \end{cases}$$

We then modify the `tick(C, m)` operator so that, for each instance $[id_j, \theta_k]$, it will decrease all the values t such that $[\texttt{?tick}/t] \in \theta_k$ by a value equal to the minimum of t and m. The semantics of $eval(KB, \texttt{TEMPORAL(n)}, \theta_j)$ is then changed to return `true` if and only if the `?tick` variable reaches value zero:

$$eval(KB, \texttt{TEMPORAL(n)}, \theta_j) = \texttt{true} \text{ iff } [\texttt{?tick}/0] \in \theta_j.$$

In other words, for every instance of a norm with a temporal deadline, we activate a timer that is decremented by a call to the function `tick`. The deadline is considered expired when the timer reaches 0. Another consequence of removing the explicit reference to the current time is that, without a reference to the activation time, multiple instances or violations associated with the same pair $(\texttt{nd}i, \theta_j)$ become indistinguishable. This leads to a number of problems at the implementation level. Consider the example in Section 5. When the coalition fails to intercept an unauthorized boat `ub` (violation of `nd2`), an instance of `nd3` that binds to `ub` will be activated and included in the list Π. Subsequent violations will bind to the same substitution in the activation condition of `nd3`, preventing any new activation. In order to solve this problem we need to make sure that every new violation of `nd2` will match, for the activation condition of `nd3`, to a substitution that is not currently in Π. We do this by adding a boolean flag in the representation of the violation in the knowledge base. When the first violation

of nd2 associated with θ_j is detected, its description is added to KB with the flag set to false. At every subsequent violation associated with the same pair $(\texttt{nd2}, \theta_j)$ we change the value of the flag. We update the semantics of $match$ for the construct VIOLATION-OF(t_1, s_1) to include the variable ?flag bound to the flag value instead of the variable ?violTime. When, for example, the flag values goes from false to true, the previous match for the activation of nd3 is deleted while the instance with ?flag set to true gets activated. This mechanism guarantees that we can activate at least one CTD instance per step for each pair $(\texttt{nd3}, \theta_j)$. Further, to correctly interpret the VIOLATED expression, we need to check for a violation of the current instance. Again, without relying on the activation time, we are not able to distinguish between different violations associated to the same pair $(\texttt{nd}i, \theta_j)$. We solve this by adding to the substitution θ_j of each instance $[id_i, \theta_j]$ a variable ?violated which is initially unbound. We modify Rule R4 (and the equivalent for violated prohibitions) to set ?violated to true when a violation is detected, and update the semantics of $eval$ for VIOLATED as follows:

$$eval(KB, \texttt{VIOLATED}, \theta_j) = \texttt{true} \text{ iff } [\texttt{?violated/true}] \in \theta_j \tag{5}$$

As a result of these modifications, rule R4 becomes as follows:

$$\frac{\begin{array}{c} mod_i = O,\ \theta_k = \theta_j \cup [\texttt{?violated/true}] \\ [id_i, \theta_j] \notin \Phi,\ eval(KB, ddl_i, \theta_j) = \texttt{true},\ KB^* = addV(KB, [id_i, \theta_j]) \end{array}}{\langle KB, \Delta, [id_i, \theta_j] : I, \Phi, \mathbf{B}, r\rangle \rightarrow \langle KB^*, \Delta, I : [id_i, \theta_k], [id_i, \theta_k] : \Phi, \mathbf{B}, \texttt{true}\rangle} \tag{R4*}$$

where θ_k is the substitution obtained by setting the value of the ?violated flag and $addV$ updates the content of KB as discussed above:

$$addV(KB, [id_i, \theta_j]) = \begin{cases} KB \cup \texttt{v}(id_i\texttt{,p}(\theta_j)\texttt{,false}) & \text{if } \texttt{v}(id_i\texttt{,p}(\theta_j)\texttt{,}f) \notin KB \\ & \forall f \in \{\texttt{true}, \texttt{false}\} \\ KB \setminus \texttt{v}(id_i\texttt{,p}(\theta_j)\texttt{,}f) \cup \texttt{v}(id_i\texttt{,p}(\theta_j)\texttt{,}\neg f) & \text{if } \texttt{v}(id_i\texttt{,p}(\theta_j)\texttt{,}f) \in KB \end{cases}$$

6.4 Model Checking Results

We implemented our scenario in Maude and ran the LTL model checker to verify properties of the system for both bounded and unbounded cases.

Table 1 shows the results for bounded model checking[5]. The scenario implemented includes a single UAV a Helicopter and two unauthorized boats and is regulated by norms nd1, nd2 and nd4. In all these scenarios agents can perform, according to their capabilities, at most seven actions: start and stop monitoring, start and stop intercepting, start and stop reporting, and move to a different area. We checked the following property, which asks whether a state where uav does not monitor the restricted area area2 always results in a violation of nd1:

$$\Box((\neg monitoring(uav1, area2)) \rightarrow violated(1))$$

To prove that this property is always true the model checker has to observe the whole state space, giving us a worst-case scenario in terms of execution time. We can see that both the execution time and the number of states increase exponentially with the number of steps.

Table 1. Model checking results: bounded steps

	Step Limit				
	7	**8**	**9**	**10**	**11**
States	4647	12352	32336	81504	202007
Execution Time	10 s	29 s	78 s	3m 8s	8m

Table 2. Model checking result: unbounded

Part a: `ddl2 = ddl3 = TEMPORAL (3)`							
cA	**uB**	**nd1**	**nd2**	**nd3**	**nd4**	**States**	**Time**
2	2	✓	✓			5250	20s
2	2	✓	✓		✓	20012	2m
2	2	✓	✓	✓	✓	243994	1h,16m
3	2	✓	✓			19032	2m
3	2	✓	✓		✓	72327	15m
3	2	✓	✓	✓	✓	870165	25h

Part b: `ddl2 = ddl3 = TEMPORAL (1)`							
cA	**uB**	**nd1**	**nd2**	**nd3**	**nd4**	**States**	**Time**
1	2	✓	✓	✓	✓	5717	40s
2	2	✓	✓	✓	✓	17653	5m
3	2	✓	✓	✓	✓	75245	16m

Table 2 shows the results for unbounded model checking in different scenarios. **cA** is the number of coalition agents, **uB** the number of unauthorized boats, while for each `nd`i, a ✓ indicates that the norm was included in the scenario.

The scenario in row 2 (Table 2.a) is equivalent to that used to produce the results in Table 1. Note that the execution time for bounded model checking at 10 steps is higher than the unbounded case. This is due to the fact that, since we include the time value in KB, conceptually equivalent states are not recognized because their time values differ, making it impossible for the model checker to take advantage of optimizations that rely on state matching.

As we can see from Table 2.a, the scenarios where both `nd2` and `nd3` are enforced are those with higher execution times. We believe this is due to an interaction between temporal deadlines and CTD obligations: In fact `nd3` is a CTD of `nd2` and each of them has a temporal deadline of 3 steps. Values for the `?tick` variable range from 3 to 0 in instances of `nd2` and, whenever `nd2` is violated, the timer for `nd3` is initialized. Our intuition is confirmed by Table 2.b: by decreasing the deadline to 1, we obtain significantly smaller state spaces and execution times.

We now show how model checking can be used to verify that our normative specification is correct, by checking that non compliant behaviours are detected as violations. Let's consider a variation of `nd2` stating that, in order to optimize the allocation of resources, we want one and only one member of the coalition to intercept the unauthorized boat detected in the restricted area. Intuitively we would be tempted to express the norm with the following goal:

$goal2$ `= COUNT ( ?ag2 IN { memberOf(?ag2,?add) /\ intercepting(?ag2,?ag1) } ) = 1`

which holds true if the number of agents (`?ag2`) that are members of the coalition and are intercepting `?ag1` is equal to 1. We can now use model checking to verify whether this specification captures the meaning we intend. For example, we might ask whether it is true that having two agents intercepting the same boat results in a violation. We refer to `area2` to be the restricted area, `ub` the unauthorized boat, and `uav` and `heli` the UAV and the helicopter respectively. We check

[5] All tests ran on a Intel Core i5 2.7Ghz, 16 GB RAM.

the following property, which says that having both uav and heli intercepting ub always results in a violation of nd2.

$$\Box((\texttt{intercepting(uav,ub)} \wedge \texttt{intercepting(heli,ub)} \wedge \texttt{inArea(ub,area2)}) \rightarrow \texttt{violated(2)})$$

The model checker returns an execution trace that violates the property as a counter example. In fact, if the uav and heli start intercepting at two different instants of time, the obligation is fulfilled (and thus deleted) when the first agent starts intercepting. We can capture the intended meaning with an obligation to have at least one agent intercepting before the deadline and a prohibition from having multiple agents intercepting the same boat.

We now show, with an example, how model checking can be used to verify robustness-related properties. We want to verify whether compliance with nd2 and nd3 guarantees that an unauthorized boat cannot enter and exit the restricted area without being reported or intercepted. We denote by area1 and area2 an unrestricted and a restricted area respectively. The following property says that there is no path such that ub goes from area2 to area1 being neither intercepted nor reported and without triggering a violation of nd2 or nd3.

$$\neg\Diamond(\texttt{inArea(ub,area2)} \wedge \Diamond\texttt{inArea(ub,area1)} \wedge \Box(\neg\texttt{violated(2)} \wedge \neg\texttt{violated(3)} \wedge \neg\texttt{intercepting(uav,ub)} \wedge \neg\texttt{reporting(uav,ub)} \wedge \neg\texttt{intercepting(heli,ub)} \wedge \neg\texttt{reporting(heli,ub)}))$$

The model checker shows as a counterexample a path where ub moves from area2 to area1 before the deadline for it being intercepted, causing the expiration of nd2. We thus verified that our normative system does not guarantee that the specified critical situation will never occur, even if we consider only compliant paths. If we want to make sure that, in a situation of compliance, a boat that exits the area is at least reported, we can modify $exp2$, $ddl2$ and $exp3$ as:

$exp2$ = VIOLATED ; $exp3$ = false
$ddl2$ = TEMPORAL(3) \/ NOT EXISTS{inArea(?ag1,?ar)}

In this way, both the expiration of the temporal deadline or ub exiting area2 before being intercepted trigger a violation of nd2, thus activating an instance of nd3. By applying model checking we can see that compliance with revised norms nd2 and nd3 guarantees that the boat is intercepted or reported.

7 Discussion

The formalism we use to represent norms builds upon a number of approaches to formalise norms for practical applications. For example Tinnemeier et al. [14] describe the operational semantics of a normative language with support for norms with deadlines and CTD obligations. Hübner et al. [9] adopt an SOS-approach to formalise the norm lifecycle (activation, fulfilment, violation, etc.) for monitoring the execution of norm-governed systems, which provides the underpinning for a language (NOPL) for programming such systems. Alvarez-Napagao et al. [2] propose a semantics based on production systems for a norm monitoring component that supports norms with deadlines. We complement this existing research by addressing the issue of verifying temporal properties of such systems. CÒIR also permits the representation of collective imperatives, which are not considered in existing models defined using semantics at the operational level.

Existing research on the verification of properties of normative systems has been focussing on restricted representations of norms, considering only variations of conditional deontic logic, without considering deadlines, event-driven norms, or collective imperatives. Dennis *et al.* [5], for example, integrate the ORWELL normative language in the MCAPL verification framework in order to verify properties of agents' organisations. In ORWELL, however, norms are represented through *counts as* rules, which label states as compliant or non-compliant by saying that a *brute fact* counts as an *institutional fact* (e.g. a violation) in a certain context. Our results (Table 2), show that, despite using a more expressive representation, verification times are comparable to those reported by Dennis *et al.* [5]. An alternative approach is that proposed by, for example, Ågotnes et al. [1], where transitions of a Kripke structure are labelled as compliant or non compliant. It is then possible to use model checking to verify properties of the system under different compliance assumptions. While such a labelling might be expressive enough to represent the kind of norms captured by our formalism, it is not clear how to compute it from a declarative normative specification.

We believe that this mismatch between formalisms used to specify and monitor norms and those used to verify and analyse normative systems makes it difficult to ensure that the norms being implemented satisfy certain desired properties. Our work attempts to bridge the gap between norm specification, monitoring and verification, by providing an executable specification that is verifiable through model checking.

For future research we plan to explore techniques to exploit domain symmetries in order to improve performances and to extend our model to allow agents to issue imperatives at run-time.

8 Conclusion

In this paper we proposed CÒIR, a language for the specification of obligations and prohibitions with support for common features of real world norms, including deadlines, contrary to duty and event-based activation/deactivation. We showed how, thanks to the fact that we allow existential and universal quantification over variables, our formalism can be used to specify common patterns of collective obligations. We then formalized how norms are to be interpreted by means of an operational semantics which we then implemented in Maude. We discussed how the fact that we explicitly represent time in our model leads to an infinite state space, and hence proposed an abstraction of our model that preserves the semantics and makes unbounded model checking decidable. We then used the Maude LTL model checker to validate our normative specification and to verify its robustness to violations.

Acknowledgments This research was sponsored by Selex ES.

References

[1] Ågotnes, T., Van der Hoek, W., Wooldridge, M.: Robust normative systems and a logic of norm compliance. Logic Journal of IGPL 18(1), 4–30 (2010)

[2] Alvarez-Napagao, S., Aldewereld, H., Vázquez-Salceda, J., Dignum, F.: Normative monitoring: semantics and implementation. In: Coordination, Organizations, Institutions, and Norms in Agent Systems VI, pp. 321–336. Springer (2011)
[3] Clarke, E.M., Grumberg, O., Peled, D.: Model checking. The MIT press (1999)
[4] Clavel, M., Durán, F., Eker, S., Lincoln, P., et al.: All about Maude – a high-performance logical framework. Springer (2007)
[5] Dennis, L., Tinnemeier, N., Meyer, J.J.C.: Model checking normative agent organisations. In: Computational Logic in Multi-Agent Systems, pp. 64–82. Lecture Notes in Computer Science, Springer (2010)
[6] Dignum, F., Broersen, J., Dignum, V., Meyer, J.J.C.: Meeting the deadline: Why, when and how. In: Formal Approaches to Agent-Based Systems, pp. 30–40. Lecture Notes in Computer Science, Springer (2005)
[7] Gasparini, L., Norman, T.J., Kollingbaum, M.J., Chen, L.: Severity-sensitive robustness analysis in normative systems. In: Proc. of the 17th Int. Workshop on Coordination, Organisations, Institutions and Norms (2014)
[8] Grossi, D., Dignum, F., Royakkers, L.M., Meyer, J.J.C.: Collective obligations and agents: Who gets the blame? In: Deontic Logic in Computer Science, Lecture Notes in Computer Science, vol. 3065, pp. 129–145. Springer (2004)
[9] Hübner, J.F., Boissier, O., Bordini, R.H.: A normative organisation programming language for organisation management infrastructures. In: Coordination, Organizations, Institutions and Norms in Agent Systems V, Lecture Notes in Computer Science, vol. 6069, pp. 114–129. Springer (2010)
[10] Lamport, L.: Real-time model checking is really simple. In: Correct Hardware Design and Verification Methods, pp. 162–175. Springer (2005)
[11] Norman, T.J., Reed, C.: A logic of delegation. Artificial Intelligence 174(1), 51 – 71 (2010)
[12] Plotkin, G.D.: A structural approach to operational semantics. Tech. Rep. DAIMI FN-19, University of Århus (1981)
[13] Şensoy, M., Norman, T.J., Vasconcelos, W.W., Sycara, K.: OWL-POLAR: A framework for semantic policy representation and reasoning. Web Semantics: Science, Services and Agents on the World Wide Web 12–13(0), 148 – 160 (2012)
[14] Tinnemeier, N., Dastani, M., Meyer, J.J.C., van der Torre, L.: Programming normative artifacts with declarative obligations and prohibitions. In: Int. Joint Conf. on Web Intelligence and Intelligent Agent Technologies. pp. 145–152 (2009)
[15] van der Torre, L.: Contextual deontic logic: Normative agents, violations and independence. Annals of Mathematics and Artificial Intelligence 37(1-2), 33–63 (2003)
[16] Vasconcelos, W.W., Kollingbaum, M.J., Norman, T.J.: Normative conflict resolution in multi-agent systems. Autonomous Agents and Multi-Agent Systems 19(2), 124–152 (2009)

The Role of Knowledge Keepers in an Artificial Primitive Human Society: an Agent-Based Approach

Marzieh Jahanbazi, Christopher Frantz, Maryam Purvis, Martin Purvis

Department of Information Science, University of Otago, New Zealand
{Marzieh.Jahanbazi@Postgrad.Otago.ac.nz,
Chrsitopher.Frantz@Otago.ac.nz, Maryam.Purvis@Otago.ac.nz,
Martin.Purvis@Otago.ac.nz}

ABSTRACT. This paper discusses knowledge accumulation and diffusion mechanisms and their effect on social and institutional change in an artificial society. The focus of this paper is to model the role of *knowledge keepers* in the CKSW institutional meta-role framework. In literature this role has been associated with helping to maintain social order by spreading social awareness and resolving disputes. In addition to outlining the model of a complex, adaptive, and self-sustaining artificial society, we examine in this context the societal mechanism of violence control.

Keywords: Artificial social systems, Social Simulations, Institutions, Complex Social Systems

1 Introduction

An increasingly popular approach for understanding complex social interactions in the social sciences is agent-based modeling and simulation [1–5]. Most of the works in this area take a specific perspective on the complex world of human societies and model phenomena related to that perspective in isolation from any other aspects of the society. However, agent-based modelling affords the opportunity to see how multiple interconnected factors may interact and affect the overall outcome.

This paper describes a model of primitive human communities with thousands of agents across multiple generations. Apart from representing an archetypical primitive society, the model affords measuring changes of social relationships of agents over time and their effects on societal wellbeing. Furthermore, it demonstrates how these modelled people dynamically adapt to different levels of resource availability or different demographic compositions. The model introduces a set of specific social interactions, such as mutual sharing, maintaining personal relationships, and keeping up with social reputation changes. We deem those to be applicable to primitive societies in particular in order to measure their long-term effect on the society's structural makeup and socio-economic development.

A notable feature of our model is its representation of generic roles that characterize some of the fundamental social activities in the society and how they are coordinated. In particular the generic role of the knowledge-keeper will be shown below to be a key element in the coordination of the society's activities. It is our belief that such generic agent roles, such as that of the

knowledge-keeper, are crucial aspects of inter-agent coordination and are as fundamental to social sustainability as organizational structures, such as norms and institutions.

2 Background

As discussed in [6], primitive communities can be considered to be a good starting point for modeling human interactions and societies' structures. Agent-based models of such societies typically have agents operate according to simple rules that are derived from ethnographic field studies. We built our model based on the earlier extensive studies of primitive cultures that were initiated by Younger [6–10]. Younger's work was based on his observations of pre-contact Pacific Island societies, and they can serve as an archetype for pre-modern societies without advanced and explicit institutional structures. In order to structure both the society and internal agent's structure we follow the CKSW approach of Purvis and Purvis [11, 12] that identifies four fundamental meta-roles of social interaction that are believed to be found in every society. The CKSW Meta-Role Model consists of four basic meta-roles:

- **C** – the Commander role. It characterizes leaders and those who are in charge of decision-making and have access to coercive authority to control others.
- **K** – the Knowledge role. The Knowledge specialist role has the responsibility to create, maintain, control, and transmit institutional knowledge. Since its central feature lies in the management of knowledge, we refer to it as knowledge keeper in the remainder of the text.
- **S** – the Skill role characterizes know-how intelligence. Skilled people develop tools to enhance their operations, and so they have historically engaged in trade to exchange these tools with other groups.
- **W** – the Worker role represents the general working population which can use tools to engage in productive activities.

The reflection of the CKSW meta-role model in real human societies suggests that it can provide a natural structural scaffolding in agent-based models of artificial societies. Its application to our model of an evolving primitive society is particularly suitable, since it allows us to model and retrace structural development of a society both on an individual level, an intermediate level (classes of agents that are primarily dedicated to a particular role), and a macro level (the overall structural outcome). The internal (individual level) CKSW element defines different types of agents with varying preferences in the light of similar opportunities. For example an individual with a relatively high K (knowledge)-value would be more able to use and exploit knowledge that becomes available. In earlier work of Jahanbazi et al. [13], covering social interaction in primitive societies, only the C and W meta-roles were included in the social model. In general, however when societies become more organized, it is natural for them to start keeping track of and managing knowledge of general value, thereby shaping their value systems and culture. For example, a K-specific aspect is the interpretation of the natural environment and phenomena. Thus special social roles with a focus on K-management have arisen in early past societies, such as the "medicine man" or priest that managed and interpreted knowledge. Thus we believe that societies first emerged with C and W meta-role sectors (the most primitive societies) and then developed into societies with C, W, and K meta-role sectors. Only later were all four C, K, W, and S meta-role sectors present in more developed societies. The work presented here describes a model for early C-K-W societies that have agents that activate the C, K, and W meta-roles.

Work on the part of other social scientists and agent-based modellers has investigated building artifical societies, but without the CKSW scaffolding. Each uses a different approach and different angle to define the complex world in their model. The model developed by [3, 14] shares our objective for developing a model which allows endogenous progression of

institutional development. There are many works which only focus on singular aspects captured in our model, for instance population dynamics [15–17], mate selection [18–20], kinship [21], leadership and governance [3, 6], institutions [3, 14], economic development [2, 5, 22] or even modeling the society's history [23, 24].

Due to the multifaceted nature of our model and limited space, we can only briefly introduce the various elements of the model as well as features relevant to the knowledge-keeper role in the upcoming section.

3 Model Description

Our model consists of one or more villages of people, each with a leader. All agents have a finite lifetime (they can die of "old age") and need to eat food resources in order to sustain themselves. If an agent doesn't eat enough food, it can die of hunger. For this reason agents may sometimes be motivated to steal food from others. But agents may be killed for either stealing food or for reasons of revenge due to negative opinions of each other or previous negative experience. During their fertility ages, agents find mates (based on the matching of their mutual relationship values) and reproduce offspring that inherit (with a small possibility of mutation) their parents' characteristics.

Model Overview.

In summary, our core model follows the idea that ordinary worker agents live in a village that is ruled by a leader agent and undergo a regular daily life cycle. They gather food from the environment and bring it to storage locations controlled by the leader. In our model an agent's time schedule is based on its own characteristics. For example, while the length of day is a universal parameter and is the same for all agents, an agent's "productive time" depends on its loyalty and defines how many time units during the day that they must work for the village leader. During their productive time period, the follower (i.e. non-leader) agents are under the command of the leader and gather food from the surrounding area which they then deposit into a central storage controlled by the leader.

After an agent's productive time period has elapsed, it is free from obligations to the leader. At this point agent can keep the collected food. Agents can carry this food around, or can store it in their home. The stored food at home is accessible by all members of a household and is secure from theft, while the food that agents carry might be subject to theft. Beyond these activities, agents engage in other activities, such as sharing food (in order to increase their reputation and hence increase their chance of finding a mate), stealing, socializing (sharing what they know about other agents' reputations with other agents that they know), and taking revenge if they hold a negative relationship value toward another agent. An agent could have a negative relationship value toward another agent if it were to witness that agent's stealing or killing acts, or witness an out-group agent (an agent from another village) collecting food from the observer's village's food sources. Apart from these actions, agents also perform automated activities that do not require deliberation. Those include growing older, experiencing increase in the hunger level due to energy consumption, eating (if they carry food and their hunger level is high), observing other nearby agents, and mating (under the condition that they had already found a mate).

Leaders maintain order in the village, but they do not gather food. They have control over the village's storage, however. They issue orders to collect food. Furthermore, they might share food with hungry follower agents based on their own loyalty and altruism level. They also have the power based on their aggression level to arrest agents who commit crimes in their vicinity. The overall social climate is affected by the leader's behavior. For example, leaders with high

personal altruism levels tend to share more food with their followers, which can lead to social welfare without starvation (but also possibly to overexploitation with deleterious results). On the other hand, leaders with high personal aggression levels prevent more crimes and therefore decrease overall deaths due to crime. A schematic and high-level overview of the simulation is shown in Algorithm 1.

Schematic overview of the simulation run

```
1:  Initiate Global Parameters
2:  Set the physical environment
3:  Create the agents
4:  Assign Leader to each village
5:  for SimulationDuration do
6:      if clock < (Loyalty * LengthOfDay) and IsFollower and LeaderOrder = Col-
        lectFood do
7:      Move Toward Food Sources
8:      Collect Food
9:      Move Back to MyVillage
10:     Deposit Food Into CentralStorage
11: else
12:     if LeaderOrder = ShareFood do
13:         Get a share of food
14:     end if
15:     Eat Food at Food Source
16:     Share Food
17:     Move Back to MyHome
18:     Deposit Food into HomeStorage
19:     Steal
20:     Take Revenge
21:     Share Normative Reputation
22:     Observe Others
23:     Eat Food From Home Storage
24:     Procreate
25:     while Death Condition = False do
26:         Grow Older
27:         Consume Energy
28:         Forget Old or Unimportant relationships
29:         Find Mate
30:     end while
31: Update Food Resources
32: Update Statistics
33: Update Leaders
34: end for
```

Algorithm 1: high level schematics overview of the simulation.

Model Functional Aspects.

Our agent-based model is implemented in Netlogo [25], for which the location granularity is referred to as a "patch" and relationships between agents are demonstrated with "links". In the following we will give insight into the functional aspects of the model.

The individual agents in our model have the following feature categories:

- **Simulation related variables**. These track an individual's states, such as its needed food resource level, the amount of food resources it may be carrying, its current chosen goal, or the location of its home (its "patch"). This also includes a list of known resource locations.
- **Demographic related variables**. Age, sex, and fertility rate are part of this group.
- **Kinship related variables.** These include references to parents, children, mate, lineage, siblings, and their village.
- **Personal variables.** These include Altruism, Aggression, Loyalty, Physical Ability, and they are represented by a value between 0 and 1.0. These variables are adopted from [6].
- **Role related variables.** Agents can be Leaders or Followers (corresponding to Commanders (C) and Workers (W)). In addition there is a Leader Class of agents with family ties to the current leader (they are still follower agents but they may have special privileges) .In this connection there is a loyalty-level parameter. For the leader of a village, it determines the extent to which his ruling is coercive. But for followers, this parameter determines how likely they are to obey orders.
- **Agents' internal CKSW variables.** Each agent has C, K, S, and W attributes, and for each such attribute there are two values – a capability value and an achievement value. The capability variable reflects how an agent will react to various opportunities available in the environment. For example, if an agent must choose between (1) exploring ways to be able to collect more food resources, and (2) exchanging information with other agents about known resources, then its choice will be determined by whether its knowledge (K) capability or skill (S) capability is dominant. If its knowledge capability is dominant, then the agent will choose to exchange information. This achievement level can be enhanced over time according to defined individual learning rates.

Agents Interactions.

Agents keep track of their relationships with other agents. The relationships of agents are maintained using an internal interaction matrix maintained by each agent that holds information about other agents it has encountered. The matrix is modified based on the observation of 'good deeds', such as sharing, and likewise adjusted based on negative experiences with an agent, such as observing or being the victim of stealing. Associated with this is the essential action of socialization. Similar to the notion of gossiping, when agents socialize they align their interaction matrix values in congruence with shared common acquaintances.

Relationships are represented by Netlogo "links". Each agent has a set of incoming links which are carrying another agent's opinion of the agent. Additionally each agent has a set of outgoing links that hold its opinion about other agents. The reputation of one agent is the sum of all the observational values on incoming links.

Links have the following attributes:

- **Age**: the creation time of the link.
- **Frequency**: the number of interactions so far with the agent at the end of this link.
- **Material exchange value:** the amount of resources exchanged with this agent by sharing or stealing.
- **Observational Values**: the "strength" of the relationship based on *observing* the other agent's actions or from *being informed* about that agent from other sources (for example by gossiping about a known third agent reputation).

Agent's Decision-Making.

Agents choose actions based on their internal state, which can include their hunger level, levels of altruism or aggression, as well as external state, which can be changed by the presence of a leader or enforcer agents in their vicinity. In general, we aim to use a minimum of fixed-

behaviour parameters to determine an agent's actions, and instead make use of social comparison in most decision-making activities. For example aggressive agents are not necessarily just those with aggression levels higher than 0.5 (or any other hard-wired parameter); instead, they define a personal threshold based on self-comparison with other people that they know in their village. This implies that an agent with an aggression level of 0.6 who lives near another agent whose aggression level is 0.4 might act more aggressively compared to similar 0.6 aggression level agent who lives next to an agent with a 0.8 aggression level. (If an agent's aggression level is higher than those in its vicinity, then it is more likely to act aggressively.)

Another example of how an agent's activities can vary according to the social context concerns the conditions under which an agent might be motivated to steal. Ordinarily the conditions determining when an agent might commit a crime are dependent on whether a composite set of threshold conditions is met (the *MaxHunger* value is the level of hunger at which the agent will die of starvation):

1) There is no law enforcer (e.g. a leader) nearby.
2) The perpetrating agent is not carrying food.
3) Another agent is nearby who carries food.
4) *HungerLevel*/(*MaxHunger*) > *AltruismLevel*
5) *HungerLevel*/(*MaxHunger*) > (1- *AggressionLevel*)

In addition to such situations, however, there are other conditions that could prevail. A potential crime perpetrator could evaluate the risk of getting caught and decide that it is worth committing the crime, for example, when condition 1) is not met. In that case the perpetrator agent might temporarily elevate its aggression level and commit the crime, anyway.

3.1 The Incorporation of Knowledge into the Model

Having discussed the fundamental features, we now proceed with introducing new features added to the model. In order to make the model more comprehensible, we have classified its main features based on their related structural components, which we use as a rough guide for the introduction of the model additions. Figure 1 shows the defined model components. The *Physical Environment* covers infrastructural aspects related to the simulation environment, such as the locations of resources, growth rates, defining distances between different locations, the distances between villages, village settings, and the locations of distributed village storages. The *Institutional Structure* is the social structure we impose upon the agents; it defines the structure of the society in which agents live, including the norms and the rules they must consider in their decision-making. The *Individual Agent* covers anything related to features and capabilities of individual agents.

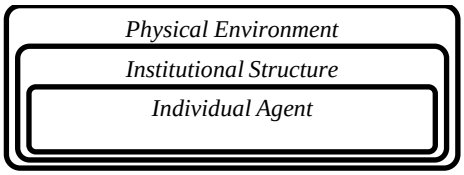


Fig. 1. Model components.

Physical Environment.

As shown in Figure 2 at the center of each village is a central storage area that the leader controls. In addition, each village has four distributed storage locations, which are also controlled by the leader and which make it easier for villages to deposit food so that there is less time spent commuting to and from food sources. There is also a common food source area between

the villages, over which each village makes a claim of ownership. Collecting food from this area may lead to revenge attacks or negative reciprocity relationships (since villagers look negatively on any other agent from another village who collects food from the common area that they claim as theirs). Note that our general model is designed so that different numbers of villages can be generated automatically.

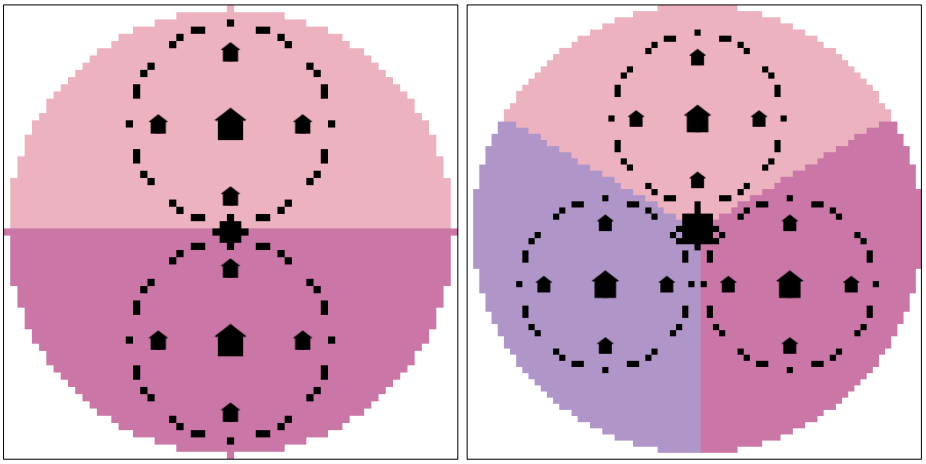


Fig. 2. Multi-Village Configurations, Each village has a central storage in the center and four distributed storage locations (small houses). Food sources, shown by black squares, are distributed in a circle with same distance from the center. A common food source area is located between villages with same distance to center of each village.

Institutional Structure.

As a society grows in size, it becomes increasingly difficult for a leader to maintain a monopoly on coercive control. So for social scalability we have introduced a class of people appointed by the leader who monitor and prevent crimes. Those agents are recruited from the "Worker Class" (i.e. regular villagers) and selection is done based on the strength of their kinship relationships to the leader. This is associated with the leader selection strategy, which is based on heredity. That is, when a leader dies, either his son, or his next closest kin will step up to become the new leader. And the new leader class will be selected based on the new leader's kinship relationships; members of the old "Leader class" will be converted back to regular workers.

The daily course of action of people in the "Leader class" group is similar to that of the normal worker class, they have all basic responsibilities; but they also have the authority to secure locations identified by the leader to prevent crimes. This is governed by a probability related to their *aggression* and *loyalty* levels. The *aggression* level determines the successful prevention of crimes, while the *loyalty* level determines how long (how many time units) these agents are under orders to maintain security at a location. They have the power to arrest agents who dare to commit a crime in their presence. Resulting prisoner agents are required to work full-time for the public good and collect food and deposit it into village storage. This strategy is in accordance with Boehm [26], who argues that in Pacific Island societies, instead of elimination of the offender, they have used another sort of temporary punishment which motivated the offender to regain group acceptance again and be able to return to life in the society. Whenever agents do get arrested, their reputation values will decrease significantly based on their current reputation level and the type of the crime they were caught committing. The secondary form of punishment is in accordance with [27], which discusses the effectiveness of combining material pun-

ishment (having to collect certain amount of food for the leader) with normative punishment (lowering one's reputation), which is a form of group punishment [28] in that it decreases the chance of the offending agent in finding a mate or receiving shared food (since an agent's reputation is publicly visible).

But this system of law enforcement will only work if knowledge about notable events is shared widely. Ordinarily whenever any notable event such as a crime occurs, nearby agents who have a high *Knowledge Capability* may observe this event and record it. But ordinary agents have only information about the areas that they visited and they don't have a big picture of the whole village. However, a group of agents with high loyalty have the opportunity to share their observations with the leader. This is in line with the notion of having a group of people who care more for their society's wellbeing and see themselves responsible to report crimes whenever they see them and take action in order to make their society safer [29]. Then the leader can decide on locations which need more control of violence. Since agents with a high *Knowledge Capability* have the motivation to share and distribute their knowledge, if they collocate with another agent with a similar *Knowledge Capability*, they can share information about their observations about events and agents they know.

Thus the distributed enforcement relies on three essential elements (see Figure 3): (1) distributed knowledge accumulation of K-agents, (2) transmission of this information by a loyal subset of K-agents to the leader who will accumulate a global overview of what's happening in his territory, and (3) the leader's decisions on whether to send enforcers to a certain area.

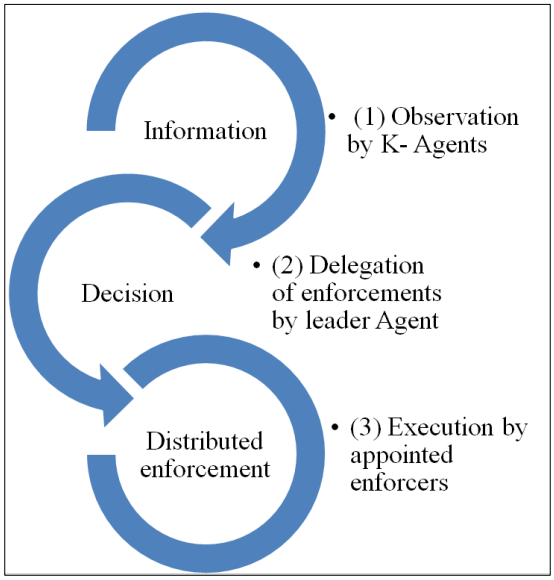


Fig. 3. Different elements and parties in distributed enforcement.

Conflict mediation.

Historians have observed that people living in small groups often go to an elder to resolve their disputes [26, 30]. An elder with good reputation can resolve the intragroup conflicts, whereas inter-group conflicts should be resolved by the leader himself. Different cultures qualify different individuals as the ones who can resolve disputes – sometimes a person with high verbal skills, a good warrior reputation or a warm personality can be considered to be a good candidate. In some other groups, wealth (or the ability to offer a material gift), generosity, aggres-

sion, self-assertion, and reputation are considered to be important. We employ the most often mentioned property, which is the reputation. In our model, reputation is also a signal of kindness, since it improves by sharing, and as kind agents grow old, they have more opportunities to share. If they have high *Knowledge Capability*, they have a higher chance of getting to know other agents and thereby have more knowledge to make judgment about contesting agents inasmuch as they know both parties involved in a dispute. Therefore high-reputation agents who have a high *Knowledge Capability* are good candidates for resolving intragroup disputes.

A significant aspect of dispute mediation is the procedure itself. In some cultures a material gift from the offending person will work, while is some other situations a duel, physical harm, or ostracism is needed to resolve the dispute [30, 31]. In our model, we used a practice of gift exchange. The amount required for this material exchange is the quantity of food units needed to make the relationship between two agents reach a neutral value.

In simulation runs which have this feature enabled, whenever an agent is collocated with another agent with whom he has a negative relationship and his aggression level is not sufficiently high to trigger revenge, then a dispute resolution mechanism will be sought. In this case the offended agent will identify another agent in the vicinity with a high reputation. Then both parties will move toward the identified mediator, and the "neutral" mediator will prescribe a penalty based on the relationship values. The target agent must pay the penalty amount to the other party to restore his reputation. The cost involved in this procedure is mostly the time both agents spend finding the mediator agent and moving toward him. The mediator agent, in turn, increases its own reputation in return, which makes him more likely to be chosen in connection with future disputes. Thus when this feature is active in our model, agents with higher reputation are expected to be experienced dispute mediators as well [13].

The combination of these two new features empowers our artificial society with a simplified version of both legal and civil justice. Legal justice aims to prevent crimes, and if enforcers catch someone committing a crime, there is a penalty of imprisonment and loss of reputation. On the other hand, civil justice attempts to resolve issues between agents by a reputable mediator agent without any actual penalty. Figure 4 shows different parties in both mechanisms.

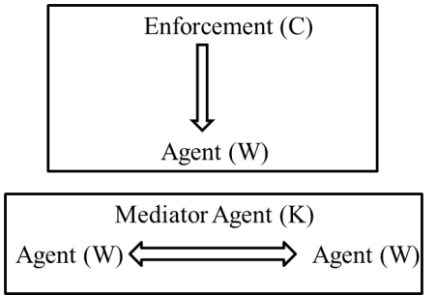


Fig. 4. Crime prevention and mediation mechanisms.

At this stage we have introduced the essential aspects of our relatively feature-rich agent model. Below, we present the results of our sensitivity analysis which we used to test the system for plausibility, but also to inform further parameter choices for selected scenarios.

4 Sensitivity Analysis

By using multi-agent modelling as a research tool, a repetitive process of defining and redefining model requirements based on extensive literature in different disciplines can be fol-

lowed in order to validate the model based on observational studies and reports from related literature. Thereafter simulations of different scenarios can help to gain deeper understanding of the causes of deviations or of optimal ways to trigger the desired outcomes [11]. We have followed a systematic approach in this fashion by tuning each model parameter to find the most reasonable value (or range of values). As defined by [32], reasonable parameters are those which help the model to reproduce patterns observed in reality. We tested hundreds of iterations for single parameters, even for the most trivial of them, such as the degree in which agents change their direction when exploring or the hunger level at which they start eating.

We began our simulation study by starting with similar parameters used as reported in previous work [7, 13, 33]. In our attempts to extend those models with new features, whenever we needed a new parameter, we have tested wide ranges of values for each one of them. Nevertheless, the selection of the range of possible values in itself is not straightforward. In order to illustrate how we went about it, we provide an example showing the steps we went through to define one of the parameters used. Although in this example we ended up with a different tactic (using social comparison instead of using a parameter), we basically followed similar steps for most of the used parameters.

Initially, by adopting a perspective similar to [7, 13, 33], we decided to use the *revenge threshold* parameter, which could be set at any negative value. Therefore we tested a range from 0 to -1000 (decremented by values of 20) to see how it affected the simulation outcomes. Each value was tested with 20 different *Random Seeds* which led to 1000 rounds for 1 village setting. The outcomes revealed that having high-magnitude values lead to collapse of the simulation (values higher than -100), due to high numbers of revenge killings (since revenge killing could start a vicious cycle of revenge attacks and thereby lead to a population collapse). On the other hand, by using very low-magnitude values, revenge attacks never happen (-600 and lower). However, since our overall approach was to employ a minimal number of parameters and by considering that not all the people have the same threshold, we took a step back and considered other factors which helped us to facilitate parameter estimation. We observed the minimum and maximum relationship values for each agent and used this range for each individual agent in the following way:

$$RevengeThreshold = Max - ((Max - Min) * (AggressionLevel))$$

Accordingly, an agent will take revenge if (a) the agent has a negative revenge threshold (indicating negative reciprocity) and (b) is collocated with an agent who has a lower-than-threshold relationship value toward him. In summary, we tested every single parameter with hundreds of experiments and used those which seemed more plausible and led to results closest to [33]. Of course the issue of "plausibility" can be subjective and is not objectively measurable, which is a framing consideration for all agent-based models.

In summary, as we stated earlier we avoided hard-wired thresholds to introduce new institutional activities that keep the social order intact or in agent decision-making processes. Instead we have used notions of social comparison among the agents to define their own views toward welfare at the societal level and at the individual level. This is also in accordance with theories of the social self and the idea that we are influenced by people around us, and the characteristics of those who are close to us will have an influence on our own [34]. We believe that this is missing in many agent-based models, inasmuch they mostly define arbitrary global parameters for such thresholds set at low, medium, or high values. We argue that it is preferable to look from a situated perspective and ask whether the effects of a particular parameter can be shown to emerge from the social and environmental context.

5 Experimental Design

We used 30 different random seeds for each pair of experiments in 2-village configurations with 100 agents as initial populations for each village. Agents can live up to 4000 time units, and we used 40,000 time units as the total duration of the each simulation run. There were three major scenario categories that we examined:

1) Scenarios without distributed control of violence (or distributed enforcement).
2) Scenarios with distributed control of violence but **without** the use of observation of events by agents with high *Knowledge Capability*. Instead we simulated global knowledge of criminal occurrences by storing the criminal events locally in the patch and making them globally visible to the enforcers.
3) Scenarios with distributed control of violence and with the use of observation of events by agents with high *Knowledge Capability*.

For each scenario we tested it with and without conflict resolution, which made a total of 6 experiments per random seed (180 in total).We considered scenario Types (2) and (3), above, in order to compare the relative efficiency difference between global knowledge about crime and knowledge about crime that is passed through knowledge-aware agents. For simulation efficiency it can be useful to store the criminal results in the patches, but it is less realistic. We found that Scenario (3), which employed criminal event observation and communication by high Knowledge-Capability agents to be almost as efficient as Scenario (2) and a more realistic representation.

6 Results and Discussion

In this section we summarize our experimental results with regard to specific features.

Effectiveness of distributed information gathering.

Before moving to our main features and their effects, the scenarios that test the accuracy of information will be discussed. The results show that the correlation between decline in death due to revenge and enabling enforcers who use crime information stored in the patches is -0.80, while the correlation between decline in death due to revenge and enabling enforcers who use the information collected by distributed knowledge gathering is -0.78. The results indicate that distributed information gathering is almost as effective as using accurate information stored in the history of patches.

Enabling distributed control of violence.

Other than a leader's control of the distribution of food based on his altruism level, there is only one institutional element that prevents agents from stealing and violence: this is provided by the authorized members of the Leader class engaged in distributed violence control. The correlations between enabling this distributed form of criminal control and different causes of death are significant. Correlation with the death rate due to (a) revenge is -0.78, (b) during thefts is (-0.54), and (c) hunger is +0.8. Specifically correlation with the total number of thefts is -0.77. In general, theft and killings are reduced considerably by implementing distributed control of violence, while death due to hunger rises. This could suggest that even in this artificial society, mere prevention of violence is not enough. There should be further institutions beyond stopping crime, such as providing the deprived agents with assistance for food acquisition. Additionally, since agents who are enforcing the rules are not productive anymore, they do not contribute to village central storage sites any longer, and thus the society has fewer contributors and more consumers. This result raises the question concerning to what degree can distributed

law enforcement be tailored to achieve a balance between crime and starvation. Figure 5 shows the average percentages of different causes of death for all scenarios with activated observation of events for configurations with and without distributed enforcement. As shown in Figure 5, death due to old age is not much affected by this feature.

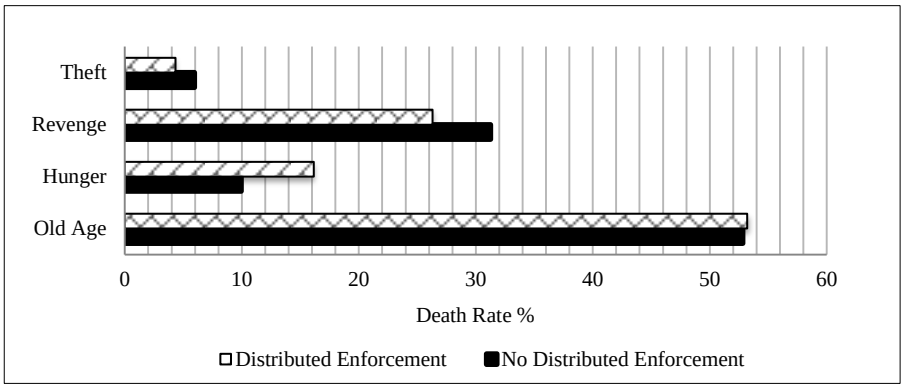


Fig. 5. Effect of distributed enforcement on different causes of death.

Conflict mediation

This feature made much more of a difference in the absence of other types of crime prevention (See Figure 6 and 7). Unsurprisingly, it has a correlation of +0.8 with the *Reputation Gini*, which defines the inequality in agent reputations[1]. The reason behind this effect is due to the role of the mediator who gains in reputation as he resolves the disputes. In addition, those with negative reciprocity toward each other have the chance to remedy their relationship and thus improve it. However, this indicates the emergence of class stratification based on reputation. While we expected that conflict mediation improves the overall welfare of the society, it has the unforeseen effect in population rise this lead to resource scarcity and more conflict over resources. This is schematically illustrated in Figure 8.

The correlations between population increase and different causes of death are significant (see Figures 6 and 7). The correlation between the number of agents and: (a) death due to hunger is +0.51, (b) death due to thefts is +0.63, and (c) death due to old age -0.8. Moreover, it decreases the life expectancy of agents in such a way that the average age at death decreases considerably when population size increases (correlation is -0.8). Figure 6 shows the average population change for scenarios with and without dispute resolution, and Figure 7 shows the average rates of different causes of death in scenarios with and without dispute resolution.

[1] The reputation Gini index shows the relative reputation inequality in a group. In particular, it reveals the gap between agents with very high reputation and agents with low reputation. In order to calculate the Gini index, we implement formula used by [35]. $Agent_i$'s reputation represented by y_i. Then we sort $y_i, i = 1\ to\ n$ in ascending order ($y_i <= y_{i+1}$). Finally, Gini can be calculated using following formula: $G = \frac{1}{n}\left(n + 1 - 2\left(\frac{\sum_{i=1}^{n}(\mathrm{n+1-i})y^1}{\sum_{i=1}^{n} y_i}\right)\right)$

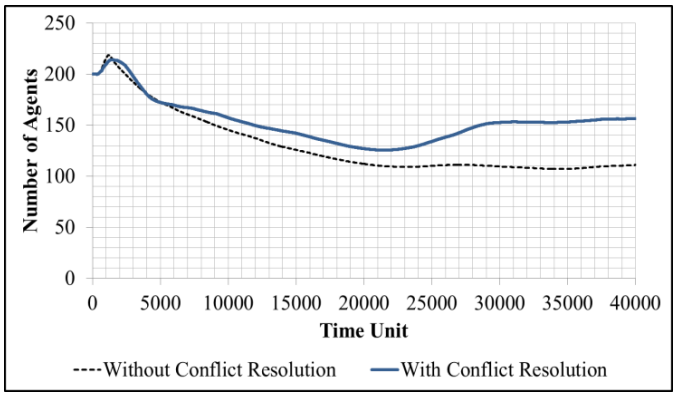


Fig. 6. Population change over 10 generations for scenarios with and without conflict resolution.

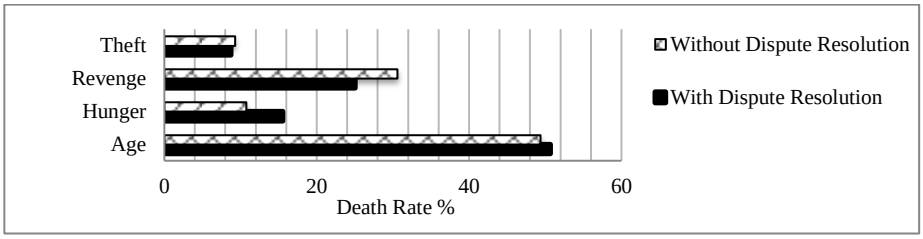


Fig. 7. Average rates of different causes of death with and without dispute resolution.

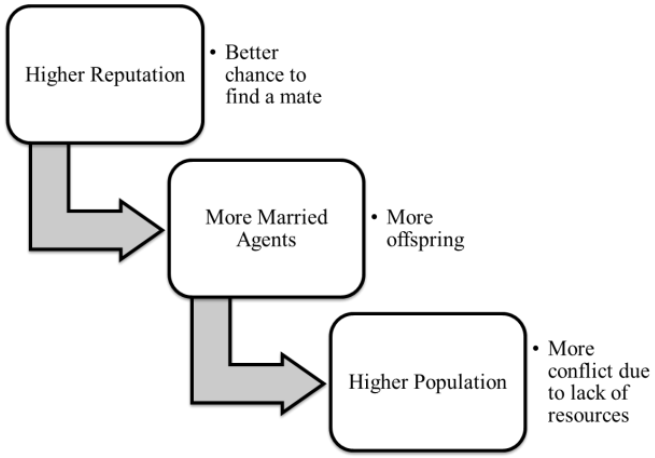


Fig. 8. Effect of higher reputation results of conflict resolution.

In addition to calculating the correlations between each feature and different outputs, we have used regression analysis to confirm the results. Table 1 summarizes the regression analysis of 180 experiments which shows the p-value and coefficients of regression test with a confidence level of 95%.

Table 1. Regression results.

	Revenge		Hunger		Thefts	
	P-value	Coefficients	P-value	Coefficients	P-value	Coefficients
Intercept	0.79	661.29	0.46	-2196.00	0.69	317.62
Random Seed	0.79	-4.7538E-06	0.45	1.65E-05	0.70	-2.338E-06
Distributed Enforce-	4E-40	-8.53	5E-50	12.47	5E-20	-1.70
Event Observation	0.020	1.27	0.00	-2.14	0.11	0.28
Conflict Resolution	0.000	-1.59	8E-07	2.52	7E-13	1.03
Adjusted R Square	**0.685**		**0.73**		**0.48**	

Distributed enforcement has significant p-values for all three output variables. It is worth mentioning that the reason behind less significant p-values for event observation compared to distributed enforcement lies in its comparison with scenarios in which enforcers had actual knowledge of crime areas. As shown in the results, distributed enforcement comes with the cost of higher death due to hunger. As mentioned earlier, this can be due to more consumers and less contributors. In the same way, in the real world, enforcement comes at a cost too, and this brings up the challenge of balancing enforcement and the cost of enforcement.

Figure 9 compares the average values for three main scenarios at once. It can be seen that as crime prevention features are added, deaths due to revenge decrease, but deaths due to starvation increase which shows the cost of resolving conflicts or its prevention.

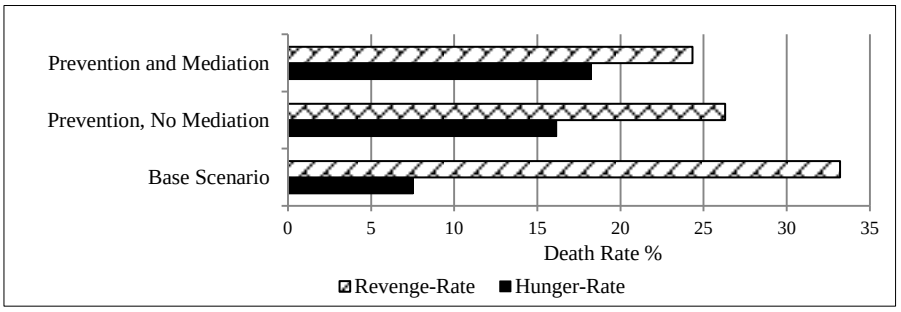


Fig. 9. Death rates due to revenge and hunger for different scenarios.

7 Conclusion And Future Work

As evidenced by the claim made in [36] that agent-based modeling is "a new standard of explanation", there has been a growing interest in agent-based modeling of complex social phenomena. However, perhaps partly due to computational limitations, the complexity and interactive scope of the modeled agents has been limited. In the work presented in this paper and by expanding the model developed by [33], we have aimed to include more aspects of a real society in the model and study the interaction of these different aspects under different simulation settings. In this work we have explored the impact of compliance and dispute resolution mechanisms on the well-being of a society, along with the structural change of the society's configuration based on the different social roles.

However our path toward building more realistic artificial human societies has much ahead of it. We believe that continued development of CKSW-based meta-role social modelling can offer new opportunities in the area of social modelling. The CKSW perspective takes into account social ordering activities that have been observed across the history of human societies. Building models using agents with these meta-role capabilities will enable us to reproduce some of the observed higher-level social structures in an organic fashion. These general role scenarios offer a more realistic representation of how primitive societies of autonomous agents achieve a measure of societal coordination.

Considerably more work will need to be done to achieve our main objective of modelling a human society with the internal ability to construct essential institutions to sustain and enhance the overall social prosperity. A number of important limitations need to be considered in order to refine and improve the current model. Some immediate extensions we will be pursuing include improving the current simplistic view of mate selection (the selected mate cannot reject the proposal) by considering real mate selection criteria in different cultures. Furthermore, we will be introducing more variation in food resource fertility rates and transportation channels. The next major extension of the model will be implementing the skill (S) class and introducing concepts such as agricultural technology for different societies.

References

1. Huang, H.Q., Macmillan, W.: A generative bottom-up approach to the understanding of the development of rural societies. Agrifood Res. Reports. 68, 296–312 (2005).
2. Macmillan, W., Huang, H.Q.: An agent-based simulation model of a primitive agricultural society. Geoforum. 39, 643–658 (2008).
3. Makowsky, M.D., Smaldino, P.E.: The Evolution of Power and the Divergence of Cooper-ative Norms, Available at SSRN 2407245(2014).
4. Gilbert, N., den Besten, M., Bontovics, A., Craenen, B.G.W., Divina, F., Eiben, A.E., Griffioen, R., Hévízi, G., Lõrincz, A., Paechter, B., Schuster, S., C. Schut, M., Tzolov, C., Vogt, P., Yang, L.: Emerging Artificial Societies Through Learning. J. Artif. Soc. Soc. Simul. 9, (2006).
5. Tesfatsion, L.: Agent-Based Computational Economics: Growing Economies from the Bottom Up. Artif. life. 8, 55–82 (2002).
6. Younger, S.: Leadership, Violence, and Warfare in Small Societies. J. Artif. Soc. Soc. Simul. 8, (2011).
7. Younger, S.: Leadership in small societies. J. Artif. Soc. Soc. Simul. 13, 5 (2010).
8. Younger, S.: Reciprocity, sanctions, and the development of mutual obligation in egalitarian societies. J. Artif. Soc. Soc. Simul. 8, (2005).
9. Younger, S.M.: Discrete agent simulations of the effect of simple social structures on the benefits of resource sharing. J. Artif. Soc. Soc. Simul. 6, (2003).
10. Younger, S.: Reciprocity, normative reputation, and the development of mutual obligation in gift-giving societies. JASSS-THE J. Artif. Soc. Soc. Simul. 7, (2004).
11. Purvis, M.K., Purvis, M.A.: Institutional expertise in the Service-Dominant Logic: Knowing how and knowing what. J. Mark. Manag. 28, 1626–1641 (2012).
12. Purvis, M., Purvis, M., Frantz, C.: CKSW: A Folk-Sociological Meta-Model for Agent-Based Modelling. Social.Path Workshop (2014).
13. Jahanbazi, M., Frantz, C., Purvis, M., Purvis, M.: Building an Artificial Primitive Human Society: an Agent-Based Approach. Co-ordination, Organizations, Institutions and Norms in Agent. Princeton University Press (2014).
14. Makowsky, M.D., Rubin, J.: An Agent-Based Model of Centralized Institutions, Social Network Technology, and Revolution, PLoS ONE 8(11) (2013).
15. Cervellati, M., Sunde, U.: Life expectancy and economic growth: the role of the

demo-graphic transition. J. Econ. Growth. 16, 99–133 (2011).
16. Read, D.W.: Emergent properties in small-scale societies. Artif. life. 9, 419–434 (2003).
17. Axtell, R.L., Epstein, J.M., Dean, J.S., Gumerman, G.J., Swedlund, A.C., Harburger, J., Chakravarty, S., Hammond, R., Parker, J., Parker, M.: Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. Proc. Natl. Acad. Sci. United States Am. 99 Suppl 3, 7275–7279 (2002).
18. Diaz, B.A., Fent, T.: An agent-based simulation model of age-at-marriage norms, Agent-Based Computational Modelling (2006).
19. Billari, F.C., Prskawetz, A., Diaz, B.A., Fent, T.: The "Wedding-Ring": An agent-based marriage model based on social interaction, (2008).
20. Bentley, G.R.: Hunter-gatherer energetics and fertility: A reassessment of the! Kung San. Hum. Ecol. 13, 79–109 (1985).
21. Read, D.: Kinship based demographic simulation of societal processes. J. Artif. Soc. Soc. Simul. 1, (1998).
22. Ewert, U.C., Roehl, M., Uhrmacher, A.M.: Hunger and market dynamics in pre-modern communities: Insights into the effects of market intervention from a multi-agent model. Hist. Soc. Res. Sozialforsch. 122–150 (2007).
23. Lake, M.: Explaining the Past with ABM: On Modelling Philosophy, (2015).
24. Barceló, J., Del Castillo, F., Del Olmo, R., Mameli, L., Quesada, F.M., Poza, D., Vilà, X.: Simulating Patagonian Territoriality in Prehistory: Space, Frontiers and Networks Among Hunter-Gatherers, (2015).
25. Tisue, S., Wilensky, U.: Netlogo: A simple environment for modeling complexity. Interna-tional Conference on Complex Systems. pp. 16–21 (2004).
26. BOEHM, C.: Hierarchy in the Forest: The Evolution of Egalitarian Behavior. Harvard University Press (2009).
27. Villatoro, D., Andrighetto, G., Brandts, J., Nardin, L.G., Sabater-Mir, J., Conte, R.: The Norm-Signaling Effects of Group Punishment Combining Agent-Based Simulation and Laboratory Experiments. Soc. Sci. Comput. Rev. 32, 334–353 (2014).
28. Boyd, R., Gintis, H., Bowles, S.: Coordinated punishment of defectors sustains cooperation and can proliferate when rare. Science. 328, 617–620 (2010).
29. Fry, D.P., Bj"rkqvist, K., Bjorkqvist, K.: Cultural Variation in Conflict Resolution: Alter-natives To Violence. Taylor & Francis (2013).
30. Wagner, G.: The political organization of the Bantu of Kavirondo. Afr. Polit. Syst. 197, (1940).
31. Wolff, P.M., Braman, O.R.: Traditional dispute resolution in Micronesia. South Pac. J. Psychol. 11, 44–53 (1999).
32. Thiele, J.C., Kurth, W., Grimm, V.: Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and R. J. Artif. Soc. Soc. Simul. 11, (2014).
33. Jahanbazi, M., Frantz, C., Purvis, M., Purvis, M., Nowostawski, M.: Agent-Based Model-ling of Primitive Human Communities. Intelligent Agent Technology., Warsaw (2014).
34. Gould, M.: Culture, Personality, and Emotion in George Herbert Mead: A Critique of Em-piricism in Cultural Sociology. Sociol. Theory. 27, 435–448 (2009).
35. Epstein, J.M.: Why Model? J. Artif. Soc. Soc. Simul. 12 (2008).

Communication in Human-Agent Teams for Tasks with Joint Action

Sirui Li, Weixing Sun, and Tim Miller

Department of Computing and Information Systems
University of Melbourne, Australia
{siruil1,w.sun}@student.unimelb.edu.au, tmiller@unimelb.edu.au

Abstract. In many scenarios, humans must team with agents to achieve joint aims. When working collectively in a team of human and artificial agents, communication is important to establish a shared situation of the task at hand. With no human in the loop and little cost for communication, information about the task can be easily exchanged. However, when communication becomes expensive, or when there are humans in the loop, the strategy for sharing information must be carefully designed: too little information leads to lack of shared situation awareness, while too much overloads the human team members, decreasing performance overall. This paper investigates the effects of sharing beliefs and goals in agent teams and in human-agent teams. We performed a set of experiments using the BlocksWorlds for Teams (BW4T) testbed to assess different strategies for information sharing. In previous experimental studies using BW4T, explanations about agent behaviour were shown to have no effect on team performance. One possible reason for this is because the existing scenarios in BW4T contained joint tasks, but not *joint actions*. That is, atomic actions that required interdependent action between more than one agent. We implemented new scenarios in BW4T in which some actions required two agents to complete. Our results showed an improvement in artificial-agent team performance when communicating goals and sharing beliefs, but with goals contributing more to team performance, and that in human-agent teams, communicating only goals was more effective than communicating both goals and beliefs.

Keywords: human-agent collaboration, BlocksWorld for Teams, joint action, interdependence

1 Introduction

Over the past decade or so, there has been a realisation that "autonomous" intelligent agents will offer more value if they work semi-autonomously as part of a team with humans [4]. Semi-autonomous agents must therefore be designed to explicitly consider the human in the loop to work effectively as part of a team.

In a joint task, a team has a joint aim to achieve a goal, and they must work together to do achieve this goal. While in some simple scenarios, team

members may be able to operate individually to achieve the joint goal, in most scenarios, the individual actions within a task are *interdependent* [17]. However, to successfully operate on an interdependent tasks, team members must have a shared situation awareness of at least part of the task, and must coordinate the actions that comprise the task. As such, communication between team members is important to efficiently complete a task.

This is the case in human-human teams, but also in human-agent teams. For example, Stubbs et al. [18] observed over 800 hours of human-robot interaction and noted that as the level of autonomy in the robot increased, the efficiency of the mission decreased as operators started questioning the robot's decision making more and more. They conclude that having an agent explain relevant parts of the behaviour to maintain a *common ground* on the task is important for effective collaboration. The process to achieve common ground requires communication, taking into account what is necessary and important, and what the other team members already know.

The aim of our work is to identify the types of and amount of information that are relevant for interdependent tasks. We use the BlocksWorlds for Teams (BW4T) [11] test bed for this. BW4T is a simulation tool that allows experimentation of scenarios involving humans and agents. The joint goal of the human-agent team is to locate and retrieve a sequence of coloured blocks in a given order. Harbers et al. [6, 5] have experimented with the same concept in BW4T, however, they found that communication did not improve team efficiency in completing the task. They hypothesise that one reason may be the simple nature of the task, and that more complex scenarios show results similar to those seen in field experiments such as the ones by Stubbs discussed above.

In this paper, we developed a new scenario for BW4T that contained *joint actions*, rather than just joint *tasks*. By this, "action", we mean the atomic actions that make up a task. Our simple extension is to introduce a type of heavy block that requires one agent to hold the door to a room for another agent, meaning that moving the block out of the room is a joint action. In terms of the model proposed by Saavedra [15], this extension moves the task from one of a team merely working in parallel towards a common goal, called *pooled interdependence*, to one of *team task interdependence*, where team members must execute actions jointly.

We performed initial experiments to assess different communication strategies teams of artificial agents, demonstrating that sharing goals improves task efficiency better than sharing beliefs. Then, we used this to determine experimental parameters for human-agent experiments on similar scenarios, and showed sharing goals in the scenarios does indeed increase the efficiency of the team in completing the task.

This paper is outlined as follows. Section 2 presents the most closely related work, and Section 3 presents relevant background on the BW4T simulator. Section 4 outlines the agent communication models used in our experiments, including how agents handle communication for joint tasks. Section 5 presents the

experimental evaluation, including results, for both sets of experiments (agent teams and human-agent teams), while Section 6 concludes.

2 Related work

In this section, we discuss the most closely related work to the work in this paper. While there is a large body of work investigating how human teams work together on interdependent tasks [15, 14] and how the process of *grounding* a common ground [12, 3], and their relation to shared cognition of a team [2, 16], this section will focus on related work on interdependence in human-agent teams.

The primary questions of work in this domain are: (1) "how much" autonomy should we grant a semi-autonomous agent, and; (2) given this, what information needs to be communicated between the agent and the human for efficient task completion. In this paper, we look mostly at the second question.

In recent years, the realisation that human-agent teams offer more than agent-only teams was lead to many empirical studies of human-agent teams [1, 13, 4] that address the issue of what and when to communicate to team members. For example, Stubbs et al. [18] discuss their experience observing over 800 hours of human-agent teamwork in a scientific setting. Their team remotely deployed a robot in the in Chile's Atacama Desert to investigate microorganisms that grow there, with the view that such a deployment would be similar to deploying a semi-autonomous robot on other planets. The team changed the level of autonomy of the deployed robot, giving it more responsibility on some tasks in certain cases, and observed the scientific teams' response. Stubbs et al. found that as the level of autonomy increased, the effectiveness of the team reduced. This was mostly caused by a lack of transparency in the robot's decision-making, resulting in cases where the scientific team spent more time discussing and trying to understand why the robot had made certain decisions, rather than on the scientific aims related to microorganisms. Stubbs et al. hypothesis that establishing a *common ground* between the relevant parties on tasks is essential.

Bradshaw et al. [1] hypothesise that human-agent teams will become more effective if agents are considered peers and team members, rather than just tools to which to delegate tasks. They later discuss the concept of *coactive design* [9], and argue that the consideration of interdependence between agents in performing joint tasks is key to effective human-agent teams. They define interdependence as the relationships between members of a team, and argue that these relationship determine what information is relevant for the team to complete a task, and in that sense, the interdependent relationships define the common ground that is necessary. In more recent work [10], they present the Coactive Design Method for designing intelligent agents that must interact with humans. In this model, interdependence is the organising principle. Human and artificial agents worked together through an interface that is designed around the concepts of Observability, Predictability and Directability (OPD). The model was applied to the design of a simulated teleoperated robot for the DARPA Virtual Robotics Challenge, and gained excellent score due to the advantages the coactive system

model. They describe scenarios in which the identification of interdependent tasks improved their agent design, such as the robot having to attach a hose to a spigot. The robot is unable to identify the hose — a task done by the human —, but attaching the hose itself was a joint task, in which the robot positioned the hose and the human directed the arm to the spigot.

Other recent work looks at how to simulate such scenarios in a laboratory setting to allow for more controlled experimentation. In particular, the BlocksWorld for Teams (BW4T) testbed [11], used in our work, was developed to support experimentation of human-agent teaming in joint activities.

Harbers et al. [5, 6] use the BW4T testbed to experiment with explanation in human-agent teams. In particular, they looked at the effect of sharing beliefs and intention within teams, providing the humans with the ability to exploit information about intentions to improve their understanding of the situation. Their results showed that, while participants reported increased awareness of what the agents were doing, there was no improvement in team effectiveness measured by completion time. Thus, their explanation model did indeed explain the situation, but this information was not useful for the human players to coordinate their actions. Harbers et al. hypothesise that this may be because the team tasks are so straightforward that the human player can easily predict what behaviour it requires, and thus processing the explanations has a cost that is similar to what the explanation is worth. We agree with this analysis. Our experiments are similar in spirit to these experiments, however, the introduction of joint action helps to provide a more complex scenario without increasing the complexity to a point that confuses the human players or requires training.

In other work, Harbers, Jonker, and Van Riemsdijk [7] used BW4T to investigate communication in agent-only teams, and found that sharing intentions and taking advantage of this knowledge increased the team efficiency, while sharing beliefs had minimal impact — a finding consistent with the work in this paper.

Wei, Hindriks, and Jonker [19] study the construction and effectiveness of shared mental models between artificial agents using BW4T. They designed four scenarios with different numbers of artificial agents and environment sizes, and measured completion time as a proxy for the effectiveness of different communication strategies. Their results showed that communicating between team members improved efficiency, especially in the case in which there were sequential interdependencies between tasks; that is, the tasks had an explicit order in which they must have been completed. Further, they also found that communicating more information lead to more interference between agents, indicating that even in agent teams where processing is not a large issue, it is important to communicate only the most relevant and important information. Our work goes further than the experiments by Wei, Hindriks, and Jonker by looking at joint actions and including humans in the loop.

3 BlocksWorld For Team

BlocksWorld For Team (BW4T) is a simulator that extends the classic blocks world domain, written specifically for experimenting with human-agent teams. The overall goal of the agent team is to search for the required blocks in a given set of rooms. The task can be performed by a single agent or a group of agents. Agents can be either artificial or human. The role of the agents can be distinguished based on how it is programmed.

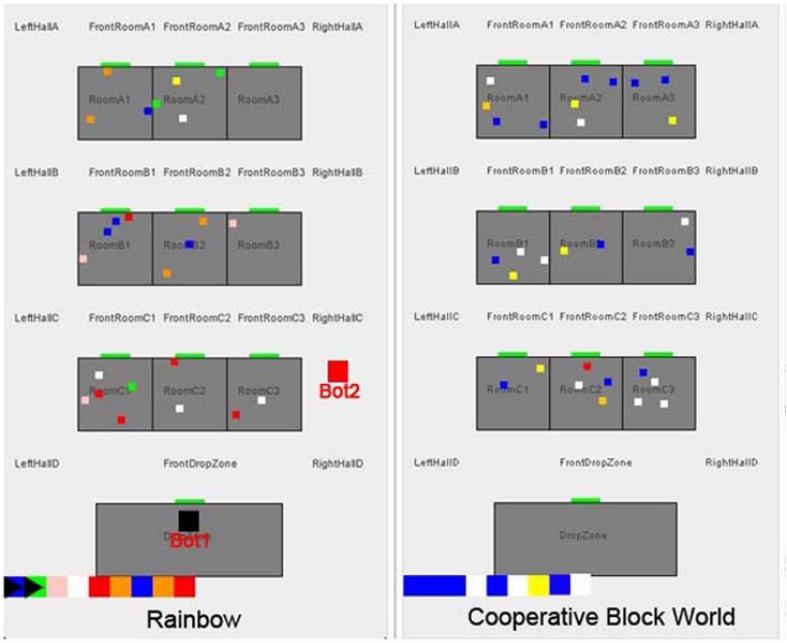


Fig. 1: The BW4T environment

Fig 1 displays the three different BW4T maps we used in our experiments. The environment of BW4T consists of rooms and coloured blocks scattered in different rooms. Each room has one door, which is represented by the small green bold line. The dark area on the bottom is the drop zone, where blocks are dropped once collected. The small black squares with red labels represent agents. At the bottom, the sequence of colours specifies the blocks that the team is tasked with collecting. The team must put down the block with the right colour into the drop zone, otherwise, the block will disappear. The sequence is represented by the colourful bar on the bottom of the environment. The small triangles on the colourful bar means the completed tasks.

The agents within BW4T are programmed using the GOAL programming language [8], and the BW4T simulator provides specific constructs for interacting

with GOAL agents. Agents can perceive the environment using an environmental sensor, including information such as the next target block, or the blocks in the room they are in.

Agents communicate to each other using messaging, and the contents can be arbitrary. On receiving a message, it is stored in a "mailbox" for reading. When an agent representing a human (which we call the *supervisor* agent) receives a message, it translates the message into a human-readable format, and displays this on the GUI that is viewable by the human. The human player can inform and direct the supervisor agent using a drop-down menu of commands; e.g. telling the agent which room a particular-coloured block is in.

4 Agents and Joint Activities in BW4T

In this section, we present the scenario and models of agents that we used to experiment with human-agent teams in joint activity. We model how an artificial agent communicates with artificial team members, and then with humans.

4.1 The scenario

From the perspective of the rules of the BW4T game, we alter only one aspect: we introduce *types* of block. In the BW4T simulator, blocks have colours, and the sequence of target blocks must be returned according to a specific colour in each slot. In our model, blue blocks are given a special status, in that they are considered *heavier* than other blocks, and they require two agents to get the block from its location to the drop zone. As part of our experiments, we implemented a simple scenario in which, when an agent wanted to take a blue block from a room, a second agent was required to hold the door open for them (because the block is too heavy to hold in one arm, and the carrying agent therefore has no hand to open the door).

This represents an *interdependent action* [17]: an agent can only take a blue block from a room if another agent opens the door, and the agent opening the door receives no value from this unless the block is taken from the room and back to the drop zone. One can imagine different implementations; e.g. two or more agents must carry blocks together, but this simple variation is enough to test out joint actions in BW4T.

4.2 Agent models

In this section, we outline our model for dealing with the joint activity of collecting a blue block. We adopt a basic model of searching and retrieving blocks, and extend this with the ability for agents to *request* and *offer* assistance for heavy blocks. In our basic model, all agents know the sequence of blocks to be found. They search rooms in a random fashion, not revisiting previous rooms, and maintain a belief set of the locations of blocks (which colours and in which rooms) that they have perceived. An agent's "default" goal is to find and retrieve

the the next block in the sequence, until they receive a request for assistance, or until another agent finds the block and broadcasts this fact, in which case they adopt the goal of finding the next block.

4.2.1 Requesting and offering assistance

Requesting help and offering assistance are required for the particular joint activity of heavy blocks. As mentioned before, blue blocks represent heavy blocks. This process for request and offering assistance for a blue block is shown in Fig 2.

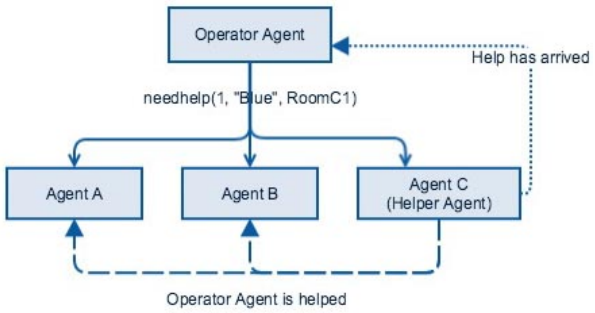


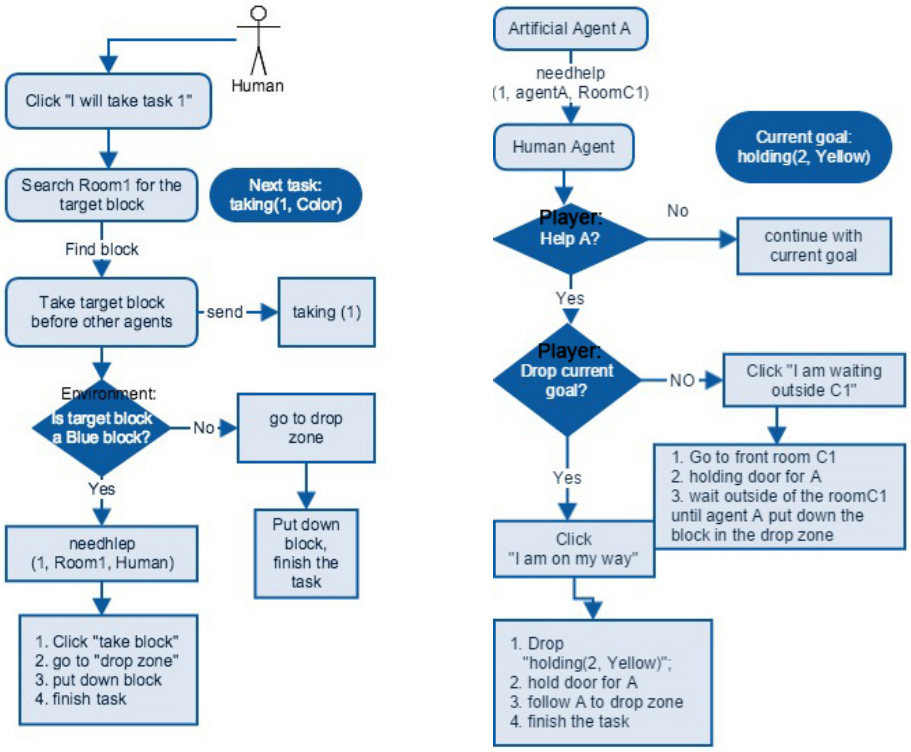
Fig. 2: Request help and offer assistance

All available agents will search for the blue block. The first agent to find one, who we call the *operator agent* will broadcast the `needhelp` message to all other agents. Any artificial agent ready to assist will move towards the room, and send a "ready" message ("Help has arrived") indicating they are at the help position (e.g. holding the door at `Room1`). The first agent to arrive will inform all others, who adopt their default goal of searching for the next block in the sequence. All agents attempting to help may not be an efficient use of their time, but we opt for a simple policy here to avoid any possibility of this policy influencing results about communication.

4.2.2 Supervisor agent

Recall that humans are represented by a *supervisor* agent, who can direct other agents to perform tasks. This agent acts as an interface between the human and artificial agents, but is also a player capable of finding and retrieving blocks. Human players direct their representative agents using high-level commands, such as which block to search for; in sense, simulating a basic remote teleoperation of a robot. Human players can request and offer assistance like artificial agents, however, the decision making about whether to offer assistance is left up to the human, rather than coded in GOAL.

Fig 3 shows the models used for a supervisor agent. The first model is used when taking on a new task (Fig 3a), and the second is used when the player intends to provide assistance to another agent trying to retrieve a blue block (Fig 3b).



(a) Completing a New Task (b) Helping the Artificial Agent

Fig. 3: Supervisor Agent

From Fig 3a, one can see that a supervisor agent is idle unless directed by the human player to take a task; that is, to starting searching for a particular colour block. The supervisor agent then searches autonomously for the block. If it finds the block and the block is non-blue, it will update the other agents to inform them that the block has been located and is being taken back to the drop zone, allowing other agents to drop this task[1]. If the block is blue, it will request help and wait. After getting help from other agent, the "*take block*" option is made available on the human player's GUI, and clicking this directs the supervisor agent to take the block to the drop zone autonomously.

[1] Artificial agents are also programmed with this capability in our model.

The other artificial agents adapt the player's changing actions. For example, if the supervisor agent drops its goal while carrying a block (e.g. yellow) to the drop zone, the other agents will drop their current task of searching for the next block in the sequence, and will adopt the goal of finding a yellow block.

The model outlined in this section is put in place to provide human decision support into the system. In a team with only artificial agents, if all agents have a current goal and one agent finds a blue block, it will be required to wait for one of its team members to complete its tasks.

However, in our model, we offer the human player the possibility to drop its own goal to help complete the tasks. We opted not to have the human player directing other artificial agents to drop goals when other agents find blue blocks, as we believed that the extra decision of *which* agent to direct could increase the cognitive load of the human player to the point where decisions became arbitrary. By allowing the human player to direct only their own agent, this model provides a complex-enough scenario to introduce an interdependent action into BW4T, without the complexity of the scenario overwhelming participants.

Ultimately, we believe that the results from our experiments (see Section 5) demonstrate that our decision is justified.

4.3 Information exchange between agents

It is clear that sharing information can improve team efficiency. However, the information shared, and how much of it, is crucial, especially in human-agent teams, where the humans' capacities to process information is reduced compared to its artificial team members.

4.3.1 Information messaging

In this section, we present the communication protocols between agents, which consist of individual messages. Several types of message can be sent, enabling agents to inform others about part of the environment, or its own goals.

Beliefs are the information about the environment, which are perceived via agents, such as the location of different coloured blocks. *Goals* are mental states that motivate action. To complete a single task, an agent must complete a sequence of goals. We enabled agents to share their beliefs and goals.

Five messages can be transferred amongst agents:

1. `block(BlockID, ColourID, PlaceID)`: block `BlockID` with colour `ColourID` has been found by the message sender at room `PlaceID`. When in a room, an agent broadcasts information about any block colours that are in the goal sequence.
2. `visited(PlaceID)`: room `PlaceID` has been visited by the message sender.
3. `hold(BlockID, ColourID)`: block `BlockID` with `ColourID` is held by the message sender.
4. `will_deliver(Index, ColourID)`: a block of colour `ColourID`, which is also the `Index`th block in the sequence of the main goal, is being delivered to the drop zone.

5. `dropped(Index, ColourID)`: the `Index`th block in the task sequence, with colour `ColourID`, previously held by the message sender, has been dropped.

While all messages are sharing information about the task, the intention of the first three is to share belief about the environment, while the intention of the last two is to share goals; e.g. when an agent delivered the task, it will drop this goal.

Agents use the information about where they have visited and what colour blocks are in the rooms to inform their search strategy. We model the artificial agents to used the shared information about block locations and room searching to improve the completion of the task. For example, when the agents share their belief about the location of blocks, others can update their own beliefs with this information, preventing unnecessary searching of rooms.

Our models use shared information about blocks being retrieved and dropped to further improve this. That is, when an agent broadcasts that they have located the next block, others will stop searching for that colour, and when a block in the main sequence is dropped, others will starting searching for this again.

4.4 Filtering for human players

The hypothesis in human-agent collaboration research is that explanation from the later can improve team performance in the joint activities. However, it is clear that humans do not have sufficient processing capabilities to use all information shared in the previous section. Despite this, the human player also needs to know some of the critical information such as the environment states and other artificial agents' message.

In our model, the supervisor agent takes on the role of an information broker who is responsible to deliver and translate information for the human player, and to filter the "explanation" from artificial agents. From the artificial agents' perspective, a supervisor agent is another artificial agent that receives and sends messages, and supervisor agents are the bridge between the environment, artificial agents, and human players.

The key part of any design is what information should be filtered out, and what should be filtered in and explained. In the next section, we describe an experiment design that looks at three levels of filtering, and their effect on the performance of the overall system.

5 Experimental evaluation

In this section, we outline two sets of experiments to provide evidence towards our hypothesis that communication can improve the team performance in joint activities, and report the results. The first set of experiments runs three BW4T scenarios using a team made entirely of artificial agents, while the second set includes a human player in the loop, along with its supervisor agent. Within each experiment, the information that is shared between team members is changed to measure the effect of information exchange.

5.1 Artificial agent team experiment

5.1.1 Experiment design

The aim of this experiment is to study which type of information sharing between artificial agents effects the team performance: sharing beliefs, goals, or both.

Independent variable We modify two independent variables: (1) communication strategy; and (2) the environment type. For the communication strategy, we use four values: (1) minimal information shared: the only communication is to ask for help moving a blue block; (2) belief only: minimal plus belief about the environment (items 1-3 in Section 4.3.1); (3) goals only: minimal plus agent goals (items 4-5 in Section 4.3.1); and (4) belief and goals. For the environment, we use three different maps: (1) cooperative block world; (2) rainbow; and (3) simple. The first two are shown in Fig 1 (page 5). Cooperative block world contains seven block colours, but only three occur in the main goal, and these are randomly allocated to the main goal in a uniform manner. Rainbow contains seven coloured blocks, and all seven colours can appear in the main goal. Simple contains randomly allocated blocks, but with no blue blocks; and therefore, no joint action.

Measures We measure completion time of the entire scenario as a proxy for the effectiveness of each communication strategy.

Setup For each map, we run all four communication strategies giving us 12 combinations. Each combination is run 30 times, with different random seeds to generate different block locations, resulting in 120 experiments run in total. All experiments were run with two artificial agents, nine rooms, and nine blocks in the main goal.

5.1.2 Results

Fig 4 shows the average completion time for all combinations of scenarios and communication strategies. This figure demonstrates several interesting findings from our experiments.

With regards to the three scenarios maps, cooperative blocks world consumes more time than other two, and the simple map, with no strictly joint action, took the least time to finish on average. This supports our hypothesis that having joint actions in a scenario increasing the complexity more than simply joint tasks. The largest gap (40%) between the cooperative block world and simple world results is in the scenario where "nothing" is shared (recall that agents still request help once they pick up a heavy blue block), indicating that sharing beliefs and goals is useful in this environment. Further, for the cooperative blocks world scenario, there is a large step between sharing belief and sharing goals, indicating that sharing goals is far more valuable that sharing just belief. This is further backed up by the small decrement from sharing goals to sharing both belief and goals. In all three maps, sharing belief had only a small impact. This finding is interesting, because while agents share their knowledge of the environment, meaning that searching for the right coloured block can be reduced, it is in fact coordinating the joint action early that increases efficiency the most in this scenario.

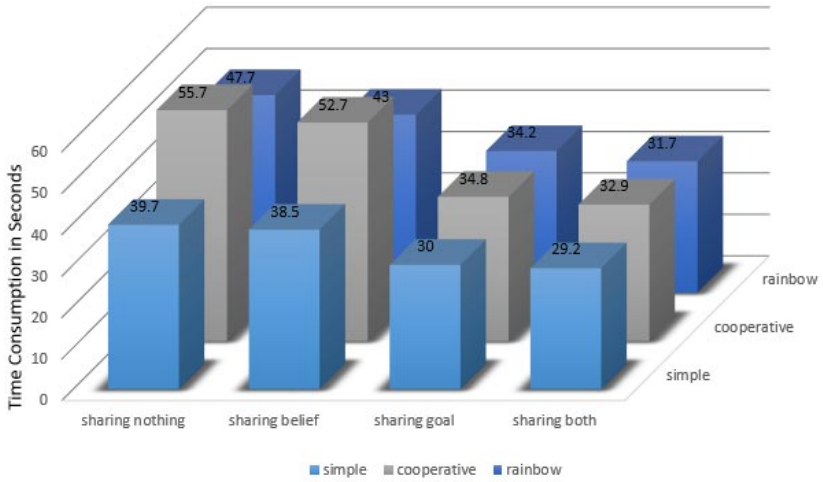


Fig. 4: Average task completion time for the artificial agents team

Source of Variance	SS	F	P-value
Communication strategy	6669.31	110.39	9.20E-33
Map	1948.40	48.38	9.97E-16
Interaction	585.17	4.84	2.04E-04

Table 1: The two-way factorial ANOVA results for the artificial agents team

Table 1 shows the outcomes of a two-way factorial ANOVA to examine the influence of the two different independent variables. The p-values for the rows (maps), columns (communication strategy), and the interaction, are all < 0.001, indicating that the results are statistically significant to this level. Comparing the sum of square errors (SS), we see that communication has more impact than the scenarios, but both factors have a significant influence on the results.

The results show that communication is beneficial for improving cooperative team work, and sharing goals has the largest impact. We drilled down into the experiment data and found that the primary reason for this was labour redundancy. An agent will update its team members once a block is placed in the drop zone, limiting the team members' knowledge of task progress. By sharing the goal that they have collected a block suitable to fulfil the current team sub-goal, the other team members can start on a new task.

5.2 Human agent teams

The results from the artificial agent teams helped to inform the design of the human-agent team experiments. In this section, we outline the experimental design and results for the human-agent team scenarios.

Information	Full info	Partial info	Silence
Next target	✔	✔	✔
Other agent's current task	✔	✔	
Request assistance	✔	✔	
Offer assistance	✔	✔	
Task completion	✔	✔	
Block location	✔		
Room occupancy	✔		
Other agents' state	✔		

Table 2: The information shared in the three scenarios

5.2.1 Experiment design

The aim of this experiment is to study how the type of information shared between the human player and other agents effects the team performance. Due to the introduction of a human into the loop, the experiment is much simplified compared to the experiments in the previous section, as we aimed to keep total completion time to under 30 minutes for each participant.

Independent variable The independent variable in the experiments is filtering strategy used by the supervisor agent to exchange information with the human player: (1) *full info*: everything is shared as in the artificial team; (2) *partial info*: only information that will change the goals of the human player are shared; and (3) *silence*: only information that a block has been delivered to the drop zone. Table 2 outlines what information is shared in each of the three cases.

Measures As in the artificial team experiment, we use completion time of the entire scenario as a proxy for the effectiveness of each communication strategy.

Setup We recruited 12 participants to perform three runs of the experiment — one with each communication strategy. No participant had used or heard of the BW4T simulator previously. To avoid bias, the order in which the participants used the various communication strategies were systematically varied.

Due to the relative difficult of recruiting participants and running the experiments, we used only one map in all three scenarios: the cooperative block world map (Fig 1). We chose this map because the results of the agent-team experiments demonstrate that this best simulates a reasonably complex scenario with joint action. The speed of the BW4T simulation is adjusted to be slow to provide the human player with sufficient time to make decisions. Each experiment consisted of two artificial agents, one supervisor agent, and one human players. There was no time out for completion of the experiment, and none of the participants failed to complete the scenario.

5.2.2 Results

Fig 5 shows the results for the human-agent team experiments. Due to the relatively smaller number of data points, results for each participant is shown. The x-axis is the communication strategy, the y-axis are the individual participants, and the z-axis is completion time. Results are sorted roughly by completion time. The overall average completion time for the three scenarios are: full info = 4.92 minutes, partial info = 4.36 minutes, and silence = 4.72 minutes.

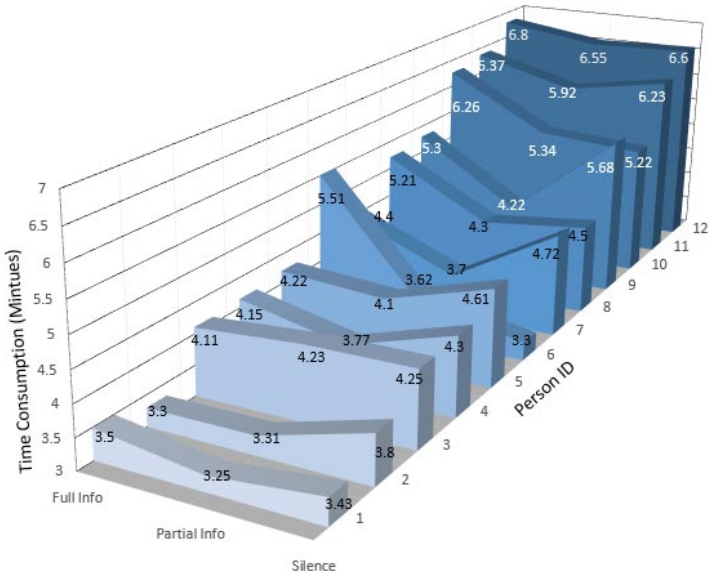


Fig. 5: The results of human-artificial agents' team

From the figure, it is clear that results differ among people, but that the difference between the strategies per person establishes a trend. From the average scores, having full information took the longest time, followed by silence, and finally, the partial information. We discuss this more below.

To test the effect of different communication scenarios, we performed a two-way ANOVA between groups, and a pairwise Tukey HSD comparison between all pairs of groups. Relevant values for the ANOVA are shown in Fig 3. These demonstrate that the results between groups is statistically significant, supporting the hypothesis that explanation can improve the team performance in scenarios with joint action, and further, that too much explanation can hinder a human players ability for decision making. For the pairwise Tukey HSD test, the full information vs. partial information results are significant at the 0.05 level, while the other two pairs are not.

Source	SS	df	MS	F	P-value
Treatment (between groups)	1.985	2	0.9925	5.07	0.0154
Error (within groups)	4.305	22	0.1957		

Table 3: ANOVA analysis of human-artificial agents' team results

6 Conclusions and future work

In this paper, we studied the effectiveness of communication in artificial agent teams and human-agent teams using the BW4T testbed. Extending previous studies using BW4T, we added the concept of a *joint action* — a single atomic action that requires more than one agent to complete.

For the artificial agents team, we performed extensive simulation experiments to assess the value of sharing beliefs, sharing goals, and sharing both belief and goals. The results showed that sharing goals, namely, agents exchanging their immediate goals, increase team efficiency significantly more than sharing beliefs.

Using these results, we designed an experiment using the same joint action scenario, but with a human player in the loop. We recruited 12 people to each play in three scenarios using three different communication strategies: (1) update only when a block sub-task has been completed; (2) share goals; and (3) share goals and beliefs. We observed that sharing goals and beliefs lead to information overload of the human, resulting in a less efficient team than just sharing goals, and that sharing almost nothing is more efficient than sharing all goals and beliefs, most likely because the scenario is still straightforward enough to guess the optimal next movement.

We identify two areas of future work: (1) a more fine-grained study on the types of goals that are shared; and (2) study of tasks in which communication is necessary to complete a task.

Acknowledgements The authors thank Christian Muise for his valuable feedback on drafts of this paper. This research is partially funded by Australian Research Council Discovery Grant DP130102825: *Foundations of Human-Agent Collaboration: Situation-Relevant Information Sharing.*

References

1. Jeffrey M Bradshaw, Paul Feltovich, Matthew Johnson, Maggie Breedy, Larry Bunch, Tom Eskridge, Hyuckchul Jung, James Lott, Andrzej Uszok, and Jurriaan van Diggelen. From tools to teammates: Joint activity in human-agent-robot teams. In *Human Centered Design*, pages 935–944. Springer, 2009.
2. Janis A Cannon-Bowers and Eduardo Salas. Reflections on shared cognition. *Journal of Organizational Behavior*, 22(2):195–202, 2001.
3. H. Clark. *Using language*. Cambridge University Press, 1996.
4. Mary Missy Cummings. Man versus machine or man+ machine? *IEEE Intelligent Systems*, 29(5):62–69, 2014.

5. Maaike Harbers, Jeffrey M Bradshaw, Matthew Johnson, Paul Feltovich, Karel van den Bosch, and John-Jules Meyer. Explanation and coordination in human-agent teams: a study in the BW4T testbed. In *Proceedings of the 2011 IEEE International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 17–20. IEEE Computer Society, 2011.
6. Maaike Harbers, Jeffrey M Bradshaw, Matthew Johnson, Paul Feltovich, Karel van den Bosch, and John-Jules Meyer. Explanation in human-agent teamwork. In *Coordination, Organizations, Institutions, and Norms in Agent System VII*, pages 21–37. Springer, 2012.
7. Maaike Harbers, Catholijn Jonker, and Birna Van Riemsdijk. Enhancing team performance through effective communication. In *Proceedings of the 4th Annual Human-Agent-Robot Teamwork Workshop*, pages 1–2, 2012.
8. Koen V Hindriks. Programming rational agents in GOAL. In *Multi-Agent Programming:*, pages 119–157. Springer, 2009.
9. Matthew Johnson, Jeffrey M Bradshaw, Paul J Feltovich, Catholijn M Jonker, Birna van Riemsdijk, and Maarten Sierhuis. The fundamental principle of coactive design: interdependence must shape autonomy. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, pages 172–191. Springer, 2011.
10. Matthew Johnson, Jeffrey M Bradshaw, Paul J Feltovich, Catholijn M Jonker, M Birna Van Riemsdijk, and Maarten Sierhuis. Coactive design: Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction*, 3, 2014.
11. Matthew Johnson, Catholijn Jonker, Birna Van Riemsdijk, Paul J Feltovich, and Jeffrey M Bradshaw. Joint activity testbed: Blocks world for teams (BW4T). In *Engineering Societies in the Agents World X*, pages 254–256. Springer, 2009.
12. Y. Kashima, O. Klein, and A.E. Clark. *Grounding: Sharing information in social interaction*, pages 27–77. Psychology Press, 2007.
13. Janice Langan-Fox, James M Canty, and Michael J Sankey. Human–automation teams and adaptable control for future air traffic management. *International Journal of Industrial Ergonomics*, 39(5):894–903, 2009.
14. Phanish Puranam, Marlo Raveendran, and Thorbjørn Knudsen. Organization design: The epistemic interdependence perspective. *Academy of Management Review*, 37(3):419–440, 2012.
15. Richard Saavedra, P Christopher Earley, and Linn Van Dyne. Complex interdependence in task-performing groups. *Journal of applied psychology*, 78(1):61–72, 1993.
16. Eduardo Salas, Carolyn Prince, David P Baker, and Lisa Shrestha. Situation awareness in team performance: Implications for measurement and training. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):123–136, 1995.
17. Ronal Singh, Tim Miller, and Liz Sonenberg. A preliminary analysis of interdependence in multiagent systems. In *PRIMA 2014: Principles and Practice of Multi-Agent Systems*, pages 381–389. Springer, 2014.
18. Kristen Stubbs, David Wettergreen, and Pamela J Hinds. Autonomy and common ground in human-robot interaction: A field study. *Intelligent Systems, IEEE*, 22(2):42–50, 2007.
19. Changyun Wei, Koen Hindriks, and Catholijn M Jonker. The role of communication in coordination protocols for cooperative robot teams. *International Conference on Agents and Artificial Intelligence*, pages 28–39, 2014.

Manipulating Conventions in a Particle-based Topology

James Marchant, Nathan Griffiths, and Matthew Leeke

Department of Computer Science, University of Warwick,
Coventry, CV4 7AL, United Kingdom
{james, nathan, matt}@dcs.warwick.ac.uk

Abstract. Coordination is essential to the effective operation of multi-agent systems. Convention emergence offers a low-cost and decentralised method of ensuring compatible actions and behaviour, without requiring the imposition of global rules. This is of particular importance in environments with no centralised control or where agents belong to different, possibly conflicting, parties. The timely emergence of robust conventions can be facilitated and manipulated via the use of fixed strategy agents, who attempt to influence others into adopting a particular strategy. Although fixed strategy agents have previously been investigated, they have not been considered in dynamic networks. In this paper, we explore the emergence of conventions within a dynamic network, and examine the effectiveness of fixed strategy agents in this context. Using established placement heuristics we show how such agents can encourage convention emergence, and we examine the impact of the dynamic nature of the network. We introduce a new heuristic, Life-Degree, to enable this investigation. Finally, we consider the ability of fixed strategy agents to manipulate already established conventions, and investigate the effectiveness of placement heuristics in this domain.

Keywords: Dynamic networks, Conventions, Social Norms, Influence

1 Introduction

Within multi-agent systems (MAS) cooperation and coordination of individuals' actions and goals are required for efficient interaction. Incompatible actions result in clashes that often incur a resource cost, such as time, to the participating agents. The pre-determination of which actions clash is not always possible, particularly for large action spaces and dynamic populations.

The emergence of *conventions* is often used to solve these problems. Conventions represent socially-adopted expected behaviour amongst agents and thus facilitate coordinated action choice without the dictation of rules. Convention emergence has been shown to be possible in static networks with minimal requirements, namely agent rationality and the ability to learn from previous interactions [5, 25]. This adds little design overhead, and is of particular importance in open MAS where agent modification is likely to be impractical or impossible.

Fixed strategy agents continue to choose the same action regardless of its efficacy or the choices of others in the system. Their presence has been shown to affect the direction and speed of convention emergence in static networks. Small numbers of these agents are able to influence much larger populations when placed within such networks [21], especially when placed using appropriate heuristics [7, 10]. Fixed strategy agents can also be used to cause a system to abandon an already established convention in favour of an alternative [13, 15].

In many domains, the nature of the relationships between agents is not static. Agents may leave the system, new agents can enter, and the links between agents may change over time. These dynamic interaction topologies induce different system characteristics than those found in static networks. Relatively little work has studied the nature of convention emergence in these types of networks.

This paper considers the emergence and manipulation of conventions within dynamic topologies. We introduce a new heuristic, Life-Degree, to support this investigation which considers aspects of the dynamic nature of the system when placing fixed strategy agents. We examine the importance of dynamic topology characteristics by comparing the performance of Life-Degree against previously used heuristics based on network metrics. We then consider the efficacy of the various heuristics when fixed strategy agents are used to destabilise or remove an established convention.

The remainder of this paper is organised as follows: Section 2 discusses the related work on convention emergence, fixed strategy agents and dynamic topologies. Section 3 describes the model of convention emergence being used, as well as the simulation model used to generate the topologies. Additionally this section introduces the heuristics used to place fixed strategy agents. Our results are shown in Section 4 and, finally, we present our conclusions in Section 5.

2 Related Work

A *convention* is a form of socially-accepted rule regarding agent behaviour and choices. Conventions can be viewed as "an equilibrium everyone expects in interactions that have more than one equilibrium" [26]. No explicit punishment exists for going against a convention nor is there any implicit benefit in the action represented by the convention over other possible actions. Members of a convention expect others to behave a certain way, and acting against the convention increases the likelihood of incompatible action choices and the costs associated with these. Conventions have been shown to emerge naturally from local agent interactions [5, 12, 23, 25] and enhance agent coordination by placing *social constraints* on agents' action choices [22].

Although the terms are often used interchangeably in the literature [17, 21], in this paper we differentiate between conventions and *norms*. Norms typically imply an obligation or prohibition on agents with regards to a specific action. Failure to adhere to norms and exhibit the expected behaviour is often associated with punishments or sanctions [1, 3, 11, 19]. Alternatively, agents may be explicitly rewarded for adherence to norms. Thus, norms generally require addi-

tional system or agent capabilities as well as incurring a system-level overhead for punishment/reward. In this paper, we assume that agents do not have the capability to punish one another, nor can they observe defection in others. Instead, we use conventions as a lightweight method of increasing coordination.

We make only minimal assumptions about agent architecture and behaviour; we assume that agents are rational and that they have access to a (limited) memory of previous interactions. Numerous studies have focussed on convention emergence with these assumptions [5, 10, 21, 25] and have shown they allow rapid and robust convention emergence. Walker and Wooldridge [25] investigated convention emergence whilst making few assumptions about the capabilities of the agents involved. In their model, agents select actions based on the observed choices of others, and global convention emergence is shown to be possible.

Expanding on this, Sen and Airiau [21] investigated social learning for convention emergence, where agents receive a payoff from their interactions which informs their learning (via Q-Learning). They showed that convention emergence can occur when agents have no memory of interactions and are only able to observe their own rewards. However, their model is limited in that agents are able to interact with any other member of the population rather than being situated in a network topology. Additionally, the convention space considered is restricted to only two possible actions. In more realistic settings larger convention spaces and more restrictive connecting network topologies are likely. The network topology agents are situated in has been shown to have a significant effect on convention emergence [4, 5, 12, 24], affecting the speed with which emergence occurs. Recent work has shown that a larger number of actions typically slows convergence [7, 10, 18].

The use of fixed strategy agents, who always choose the same action regardless of others' choices, to influence convention emergence has also been explored. Sen and Airiau [21] show that a small number of such agents can cause a population to adopt the fixed strategy as a convention over other equally valid choices. This indicates that small numbers of agents are able to affect much larger populations.

In Sen and Airiau's model, due to the lack of connecting topology, all agents are identical in terms of their ability to interact with others. However, in many domains, agent interactions may be limited to neighbours in the network. As such, some agents will have larger sets of potential interactions than others. In the context of static topologies, Griffiths and Anand [10] establish that which agents are selected and *where* they are in the topology is a key factor in their effectiveness as fixed strategy agents. They show that placement by simple metrics such as degree offers better performance than random placement.

Franks *et al.* [6, 7] investigated fixed strategy agents where interactions are constrained by a static network topology and agents are exposed to a large convention space. They found that topology affects the number of fixed strategy agents required to increase convergence speed. This also expanded on the work of Griffiths and Anand [10] by investigating the effectiveness of placing by more advanced metrics such as eigenvector centrality.

Few studies have focussed on convention emergence in dynamic topologies, with most work focussing on static networks. Savarimuthu et al. [20] consider the related emergence of *norms* in a dynamic topology. They show that norms are able to emerge under a number of conditions, but their work differs from ours due to the requirements placed on agents. The interaction model used requires agents to maintain an internal norm as well as being able to query other agents. We make minimal assumptions about agent internals or the information available. Additionally, our work investigates the manipulation of convention emergence, something not considered by Savarimuthu et al.

Mihaylov et al. [16] briefly consider convention emergence in dynamic topologies using the coordination game. Their work focusses on a new proposed method of learning, rather than on the emergence itself and how it may be influenced. In particular, they do not consider fixed strategy agents, or the action that emerges as a convention.

In this paper, we consider both convention emergence in dynamic topologies and the use of fixed strategy agents to understand the impact of network dynamics.

Relatively little work has considered destabilising established conventions, with previous investigations of fixed strategy agents typically inserting them at the beginning of interactions. We have previously [13,15] investigated using fixed strategy agents in static topologies to cause members of the dominant convention to change their adopted convention and hence *destabilise* it. We found that this required substantially more fixed strategy agents than is needed to influence conventions before emergence. We also expand on this work to examine aspects of dynamic networks when selecting fixed strategy agents for destabilisation.

This paper expands on [14] and considers the general nature of convention emergence in dynamic topologies, particularly without the use of fixed strategy agents. We also consider the effect of topology features on convention emergence time. Finally, we explore the relationship between placement heuristics, number of fixed strategy agents and the speed of convention emergence.

3 Convention Emergence Model

Our experimental setup consists of three main components, introduced below: the network topology, the interaction regime used by agents and the heuristics used for placing fixed strategy agents.

3.1 Dynamic Topology Generator

Similarly to Savarimuthu et al. [20] we utilise a particle-based simulation, developed by González et al. [8, 9], to model dynamic network topologies with characteristics comparable to those observed in real-world networks. Agents are represented as colliding particles and the topology is modified by collisions creating links between the agents. A population of N agents, represented as a set of particles with radius r, is placed within a 2D box with sides of length L. Initially,

all agents are distributed uniformly at random within the space and are assigned a velocity of constant magnitude v_0 and random direction.

Each timestep, agents move according to their velocity and detect collisions with other agents. When two agents collide, an edge is added between them in the topology if one does not already exist. Both agents then move away in a random direction with a speed proportional to their degree. Thus, higher degree nodes have an increased probability of further collisions, which in turn further increases their degree. In this way, the model exhibits preferential attachment, a characteristic found in static scale-free networks [2]. Such networks are often studied in the field of convention emergence [5, 7, 10, 18] due to characteristics that are representative of real-world networks.

Additionally, all agents are assigned a Time-To-Live (TTL) when created. This is drawn uniformly at random between zero and the maximum TTL, T_l. After each timestep agents' TTLs are decremented by one. When an agent's TTL $= 0$ the agent and all its edges are removed. A new agent is placed at the same location within the simulation with the randomised initial properties discussed above. In this manner, the topology is constantly changing.

Different topologies can be characterised by the value of T_l/T_0 where T_0 is the characteristic time between collisions. This can be expressed as:

$$\frac{T_l}{T_0} = \frac{2\sqrt{2\pi}rNv_0T_l}{L^2}$$

González et al. show that this value dictates key characteristics of the generated topology, primarily the average degree and degree distribution.

The concept of a quasi-stationary state (QSS) is discussed by González et al., such that a QSS emerges after a number of timesteps and is characterised by macro-scale stability of network characteristics. Micro-scale characteristics, for individual agents, remain in flux. In [8] it is shown that the QSS can be described as any timestep, t, where $t \gtrsim 2T_l$. Our approach here differs from Savarimuthu et al. [20] as we consider agent interactions starting from $t = 0$ rather than waiting for the QSS. This allows us to mimic scenarios where agents have been placed in a new environment rather than only considering already established networks.

3.2 Interaction Regime

Agents within the system interact with one another and, learning from these interactions, converge to a shared behaviour in the form of a convention. Agent interactions occur during each timestep of the regime. In each timestep, every agent chooses one of its neighbours in the network at random. These agents play a round of the n-action pure coordination game. In this game, both agents are given a choice from a set of n-actions, A. Agents do not know what their opponent has chosen. The payoff that each agent receives depends on the combination of chosen actions: if both chose the same action, they receive a positive payoff (+4); if the actions differ they receive a negative payoff (-1).

Each agent monitors their expected payoff for each action, based on the previous payoffs they have received when choosing that action. We adopt the

approach of Villatoro et al. [24] in this regard by using a simplified form of the Q-Learning algorithm. For each action, $a \in A$, the agent maintains a Q-Value which is updated by $Q^i(a) = (1 - \alpha) \times Q^{i-1}(a) + \alpha \times \textit{payoff}$ where α is a parameter known as the learning rate and i represents the number of times a has been chosen. All agents start with $Q^0(a) = 0, \forall a \in A$. To combat the issue of local optima, we allow each agent, with probability $p_{explore}$ to randomly select an action. Otherwise, as each agent is rational, they will always select the action with the highest Q-Value, selecting randomly between ties.

In the formulation proposed by Kittock [12], a convention has emerged when 90% of (non-fixed strategy) agents, when not exploring, would choose the same action. We adopt this definition of a convention but modify it to better fit the dynamic nature of the network topology. Instead of considering the entire population, we monitor adoption within the largest connected component. This follows from the findings of Gonzalez et al. [8] that in most simulations a giant cluster consisting of nearly all agents will emerge having the properties discussed above. Agents not within this cluster are likely to be recently created agents and, as such, should not be included in the adoption rate calculation as they have not interacted. This is reinforced by our simulations which showed that most agents not within the largest connected component had degree zero. Similarly, 100% adoption is unlikely due to new agents joining.

Fixed strategy agents will be placed within the network to study the effect they have on convention emergence. These agents will replace selected agents upon insertion, keeping all edges of that agent. This can be justified in real-world scenarios as persuading selected agents to act in a desired manner through some reward mechanism. All such agents will be assigned the same fixed strategy and their placement will be determined heuristically as discussed below. If a fixed strategy agent's TTL should reach zero, a new agent will be selected using the same placement heuristic.

We consider two different scenarios: placing fixed strategy agents at the beginning of a system's life, to encourage and direct initial convention emergence in a population, and inserting fixed strategy agents once a convention has emerged to attempt to change it. In the former case, the fixed strategy will be randomly chosen from the available actions. In the latter, it will be randomly chosen from the available actions excluding the already established convention. Initial insertion will occur once a connected component of size greater than $N/2$ has emerged. This prevents convention emergence being declared prematurely for a non-giant cluster. Additionally, placement heuristics which rely on network metrics (such as degree) may select sub-optimal agents if used before a main cluster has emerged.

3.3 Placement Heuristics

Previous work has utilised placement heuristics to enhance the effect of fixed strategy agents. Metrics such as degree, eigenvector centrality and betweenness centrality have been used with greater efficacy than random placement [6,10]. In this paper, we focus on degree-based placement. However, the dynamic nature of

the topology introduces a number of ways to apply it. All heuristics are calculated with respect to the largest connected component.

Our initial heuristic, Static Degree, corresponds to the equivalent heuristic for static networks. At the time of insertion, agents are chosen to be fixed strategy agents in descending order of degree. This selection is static once chosen, only being modified upon agent expiration as detailed above. This simplistic approach is computationally cheap, a factor of importance in settings where gathering or computing this information is expensive. However this risks selected agents potentially becoming sub-optimal choices as the simulation progresses. The static nature of this heuristic means that if another agent acquires a larger degree it will not be selected until one of the current agents expires. Depending on the TTL of the current fixed strategy agents, this could be a substantial period.

To address this issue we propose another degree-based heuristic: Updating Degree. This approach is sensitive to the dynamic nature of the topology and reselects the fixed strategy agents each timestep, based on highest current degree. Whilst this offers a solution to the potential sub-optimality of Static Degree it suffers from two problems. Firstly, the ability to acquire this information each timestep in a timely manner may be infeasible in many domains. Secondly, there is the potential that the fixed strategy agents will not remain in a given location long enough to influence the local area before being replaced.

The Static and Updating Degree heuristics do not fully consider the dynamic network context. Whilst high degree agents are likely to be influential due to their ability to interact with many others, additional dimensions may affect their applicability. Agents that are close to expiring may be less desirable than younger nodes as their expected number of interactions before replacement is much lower. However, the youngest nodes, those newly created, cannot be guaranteed to become influential later on. Hence, the age of an agent adds an additional consideration. We propose a new heuristic, Life-Degree, that allows exploration of the effect of age in addition to degree on a fixed strategy agent's efficacy.

In many settings it may be impossible to *know* an agent's TTL. However, we can estimate an agent's remaining life. Given the upper bound, T_l, and the uniformly distributed nature of TTL, the normalised expected remaining TTL, E_{rTTL}, for an agent $n \in N$ is:

$$E_{rTTL}(n) = 1 - \frac{age(n) \times 2}{T_l}$$

We can also calculate the normalised degree of a node within the largest connected component as:

$$deg_{norm}(n) = \frac{deg(n)}{\max_{n' \in LCC} deg(n')}$$

The Life-Degree heuristic is then defined as:

$$\text{Life-Degree}(n) = \omega \times deg_{norm}(n) + (1 - \omega) \times E_{rTTL}(n)$$

In this, $0 \leq \omega \leq 1$ is a weight, determining the relative contributions of degree and expected TTL.

Life-Degree allows combination of the relevant information, normalised against theoretical maximums, in a manner that allows exploration of the importance of both. Two variations of Life-Degree will be used, Static and Updating, to compare against the heuristics discussed above.

4 Results and Discussion

In this section we present our findings on convention emergence in dynamic topologies and consider the effect of agent age via our proposed heuristic, Life-Degree. Unless otherwise mentioned, all experiments used 1000 agents, the 10-action coordination game and an exploration and Q-Learning rate of 0.25. Results were averaged over 100 runs.

4.1 Characterising Topology

We initially consider convention emergence without external manipulation in dynamic topologies. This gives insight into the impact of network dynamics on convention emergence and provides a baseline. Additionally, it allows us to quantify the point at which a stable convention will have emerged for later experiments that focus on destabilisation.

The features of the dynamic topology can be manipulated by varying the parameters of the network model, and are encapsulated in different values of T_l/T_0. González et al. [9] show that the features of the topology thus only depend on the ratio T_l/T_0 and the density, $\rho \equiv N/L^2$. Additionally, they show that the average degree is a non-linear function of T_l/T_0 that depends on the chosen ρ. As such, for all experiments we use a constant $\rho = 0.625$ (i.e. $N = 1000$, $L = 40$) to allow meaningful comparisons of the T_l/T_0 values.

Parameter settings were chosen that generated values of T_l/T_0 between 0 and 20. These were rounded to the nearest integer to combine similar T_l/T_0 values, with each bucket containing 10 values. The average time taken, over 30 rounds, for convention emergence to occur was measured on the generated topologies and the average time over the bucketed values was then calculated. Values which did not result in convention emergence after 20,000 timesteps were discounted from the second average as they were unlikely to result in conventions emerging. Only runs with $T_l/T_0 \lesssim 4$ are affected by this. Simulations with a higher T_l/T_0 exhibited convention emergence for all runs. With $T_l/T_0 \lesssim 4$ as much as 80% of the runs for a given simulation did not result in convergence. The transition is notable and is discussed below.

It is clear that convention emergence is successful in the dynamic topology, and for most values of T_l/T_0 there is little variation in the average time for convention emergence as shown in Figure 1. Values of $T_l/T_0 \gtrsim 5$ all have a convention emergence time of around $t = 500$ with little variation between runs. However, values of $T_l/T_0 \lesssim 4$ displayed significant variation and, in general,

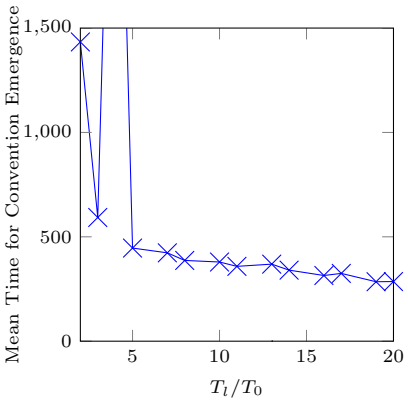


Fig. 1. Average convention emergence time for different values of T_l/T_0 with no fixed strategy agents.

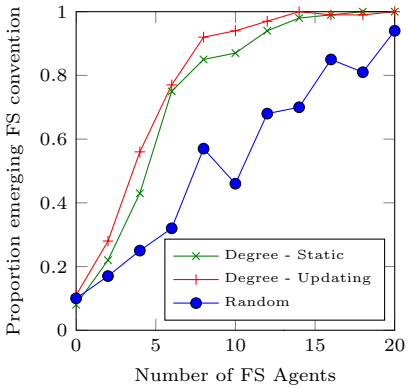


Fig. 2. Proportion of runs in which the fixed strategy emerged as a convention after initial intervention using standard heuristics

much more time was required for convention emergence to occur if it occurred at all. Higher values of T_l/T_0 did not exhibit this.

At low values of T_l/T_0 the topology was found to either not generate a giant cluster or agents were found to expire before meaningful convention emergence could occur. This follows from the parameter settings required to give a small T_l/T_0 and means that there is a lower threshold for the topology to experience convention emergence. In particular there is a minimum level of connectedness and lifespan that must be present. Below this threshold the network will be partially disconnected and not representative of real-world topologies. However, once this is achieved the time required for convention emergence is mostly independent of T_l/T_0. As such, we select parameter settings that are used for all following simulations that give $T_l/T_0 = 4.7$ which was found to provide stable convention emergence times. For completeness, additional T_l/T_0 values in the range 20 to 200 were also examined. There was a slight decrease in the average time at higher values, although the low variation remained.

As the real-world networks examined by González et al. had equivalent T_l/T_0 values around 5-6 these results were purely to determine the impact of high T_l/T_0 values, and hence have not been included.

4.2 Initial Intervention

Having established that convention emergence occurs in dynamic topologies, we now examine the effect of fixed strategy agents. We start by considering the scenario where fixed strategy agents are introduced early in a system's lifespan to manipulate convention emergence. As discussed in Section 3, this initial insertion is delayed until a cluster of size greater than $N/2$ has emerged. This was

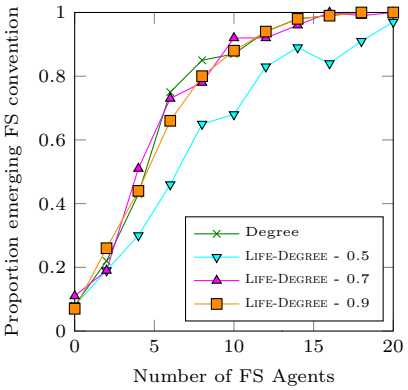


Fig. 3. Proportion of runs in which the fixed strategy emerged as a convention after initial intervention using Static Degree and Life-Degree

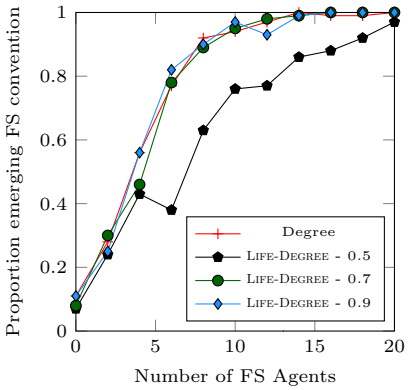


Fig. 4. Proportion of runs in which the fixed strategy emerged as a convention after initial intervention using Updating Degree and Life-Degree

found empirically to always have occurred by $t = 200$. Fixed strategy agents are inserted after this "burn-in" period has elapsed.

We begin by considering the initial heuristics discussed in Section 3: Static Degree and Updating Degree. We also consider random placement of the fixed strategy agents as a baseline. The fixed strategy agents were inserted into the system at $t = 200$ and the simulation allowed to run for 5000 timesteps. Prior simulations showed that conventions always emerged well before this time even without the presence of fixed strategy agents. The number of fixed strategy agents inserted into the system was varied from zero to twenty and the proportion of simulations in which the fixed strategy emerged as the convention was monitored. The results of this setting are shown in Figure 2.

As expected, given the size of the action space (10), when no fixed strategy agents were inserted, the proportion of times the fixed strategy emerged as the convention is approximately 0.1. With the introduction of only a few fixed strategy agents placed at targeted locations we are able to readily manipulate the emerged convention more than 50% of the time. The results also show that even randomly placed fixed strategy agents are able make a large difference in convention emergence. This corroborates the findings in previous work on static networks [10, 21], although larger numbers of fixed strategy agents are needed comparatively. As the number of inserted agents increases, the difference between the targeted heuristics and random placement becomes more pronounced. The targeted heuristics are able to cause convention emergence in nearly 100% of cases with only 12 agents whilst random placement requires 20.

Most importantly, there is very little difference between the two targeted heuristics. Updating Degree slightly outperforms Static Degree but, given the additional complexity and resource requirements needed for calculating the Updating Degree heuristic, Static Degree would likely be sufficient in most cases.

Having established the efficacy of the traditional heuristics, we now examine the effect of considering agent age using our new heuristic, Life-Degree. We begin by examining Static Life-Degree, contrasting this to Static Degree. Various weightings of Life-Degree were considered and the results are presented in Figure 3. The results of Static Degree have also been included for comparison.

When given equal weighting between expected life and degree ($\omega = 0.5$), Life-Degree performs markedly worse than Static Degree for nearly all numbers of fixed strategy agents. This is due to the fact that such a weighting is heavily biased to much younger agents. The range of possible ages is larger than that of degree and as such, even when normalised, age was found to be the primary selector. As can be seen, this has similar performance to random placement and should be avoided. A weighting of 0.7 in favour of degree exhibits similar performance to Static Degree. Further increasing the weighting offers no further improvement in performance with $\omega = 0.9$ also performing the same as Static Degree. Additional weightings of 0.95 and 0.99 (asymptotically approaching pure degree) were also considered and similarly offered no improvements.

These results show that an agent's connectivity, indicated by its degree, is a much larger contributor to its ability to influence others than how long that agent will remain in the system. The fact that considering age can only decrease the effectiveness of the chosen agents indicates that agents' short-term influence is a larger factor in convention emergence than choosing long-term targets.

Life-Degree was also used in an updating manner, such that the set of fixed strategy agents was re-calculated each iteration. The results from this and, for comparison, Updating Degree are shown in Figure 4. Similarly to the Static Life-Degree experiments, the performance of Updating Life-Degree depends heavily on the value of ω being used. As before, giving equal weighting to each factor results in poor performance, far below that of pure degree. Increasing the weighting again enhances performance but only to that of Updating Degree. This mirrors the results of Static Life-Degree and shows that, regardless of the ability to continuously assess an agent's remaining lifespan, choosing agents with numerous connections is the most important factor. This indicates that, even in the extreme case where an agent is expected to expire in a few timesteps, on average equal performance can be achieved when selecting them compared to selecting an equivalent agent who remains in the system much longer.

Static Life-Degree and Updating Life-Degree, like their pure degree counterparts, have only slight differences in performance, with Updating Life-Degree performing slightly better. However, the constant information updates may make Updating Life-Degree untenable in many domains. In domains where this information is readily available, we have shown that using up-to-date estimates of degree is sufficient to offer improved outcomes from fixed strategy agent selection.

The results presented above show that it is possible to influence the direction of convention emergence in dynamic topologies. Another commonly used metric of the efficiency of fixed strategy agents is the effect they have on the *speed* of

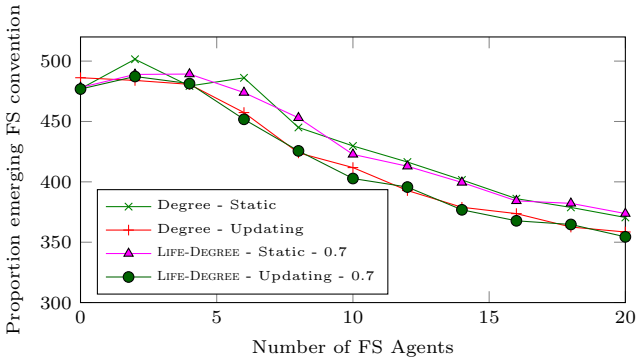


Fig. 5. Effect of fixed strategy agents on convention emergence speed

convention emergence [7,10]. Figure 5 shows how time for convention emergence varies for different numbers of fixed strategy agents using the heuristics. As is to be expected, given the asymptotic behaviour exhibited above, consideration of age, depending on weighting, causes either an increase in the average time required or results in similar times to the equivalent pure degree heuristics. Omitted from the graph for clarity, a value of $\omega = 0.5$ requires more time for convention emergence to occur for any number of fixed strategy agents. Values higher than 0.7 perform similarly to 0.7 and hence have also been omitted.

The standard deviation of the convention emergence time also decreases rapidly as the number of fixed strategy agents rises, from up to 100 with zero agents to around 20 with 20 agents. The standard deviation of the results from the Life-Degree simulations are equivalent to those of the Degree heuristics except for $\omega = 0.5$ which exhibits much larger variance. Thus, consideration of age has a negative effect both in establishing conventions as well as the time it takes to do this. This indicates that, in all aspects, degree is the factor that contributes most to how influential a given agent will be.

4.3 Late Intervention

We now look to the related use of fixed strategy agents in *destabilising* and replacing an already established convention [13, 15]. This requires a convention to already have emerged within the system. So that the results are representative of the general case, we allow a convention to naturally emerge without the use of fixed strategy agents to encourage it. It was found that conventions always emerged before timestep $t = 1500$ and, as such, insertion of fixed strategy agents occurs at this time. This also means that the system will have entered the QSS, and the topology and convention can be considered truly emerged. The action of the fixed strategy is chosen uniformly at random from the actions that exclude the established convention.

In common with the findings of Marchant et al. [13, 15] for static networks, our initial experiments showed a much larger number of fixed strategy agents

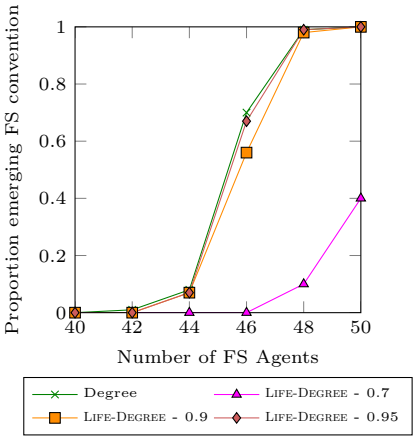


Fig. 6. Proportion of runs in which the fixed strategy emerged as a convention after late intervention using Static Degree and Life-Degree

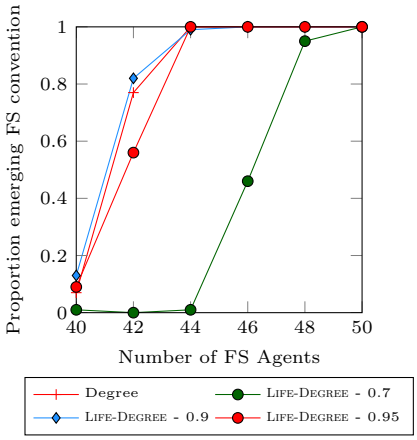


Fig. 7. Proportion of runs in which the fixed strategy emerged as a convention after late intervention using Updating Degree and Life-Degree

was required to affect the established convention compared to the number needed when inserted into a system earlier. However, a relatively small set of fixed strategy agents are still able to effect a change. In contrast to static networks, the transition between no effect and guaranteed change occurs over a much smaller range of fixed strategy agents. For nearly all heuristics (excluding random) there is little or no effect at 40 fixed strategy agents (4% of the population), whilst 50 fixed strategy agents (5% of the population) results in the targeted convention supplanting the established convention in almost 100% of cases. This narrow window indicates that there is a critical number of fixed strategy agents, nearly independent of placement, that is required to guarantee replacement of a convention in dynamic topologies.

Figure 6 shows the proportion of runs in which the convention represented by the fixed strategy became established when using the Static heuristics: Static Life-Degree and Static Degree. Like in initial intervention, consideration of age induces poorer performance here. With $\omega = 0.7$, Life-Degree is substantially outperformed by Static Degree for any non-trivial proportion, in contrast to the case in initial intervention when such a weighting produced similar performances. Even when increasing the weighting to 0.9, previously equivalent to the performance of pure degree, Static Life-Degree is still slightly outperformed by Static Degree though this is within the margin of error. Increasing the weighting further resulted in performance which asymptotically approached that of Static Degree.

Similar results are presented in Figure 7 for updating heuristics. The difference between Updating Life-Degree and Updating Degree in this scenario is even more pronounced. A weighting of 0.7 is again substantially worse than the

pure degree heuristic with the higher weightings, 0.9 and 0.95, being of similar quality to Updating Degree.

Of note, the difference in performance between static heuristics and updating heuristics is more pronounced here than in initial interventions; the updating heuristics consistently require noticeably fewer fixed strategy agents to effect a change. This indicates that the inclusion of up-to-date information regarding agent state is more important when attempting to combat an existing convention and makes a larger contribution compared to when establishing a convention from a state of neutral agents.

These findings indicate that destabilisation of an existing convention is even more sensitive to the consideration of agent longevity than initial convention emergence. Indeed, the age or expected lifespan of an agent can be safely ignored with no detrimental effects to the performance of the fixed strategy agents. This strongly implies that the major factor in destabilising conventions is instead choosing agents with high degree, regardless of how long that agent will last. High degree is more effective at spreading influence than choosing a lower degree agent with longer life. The difference between Static and Updating Degree, not present in initial intervention, also supports this view; the importance of choosing the current highest degree agents is far more pronounced.

5 Discussion and Conclusions

Convention emergence is often used in multi-agent systems to encourage efficient and coordinated action choice. It provides a mechanism though which such behaviour can naturally occur without requiring changes to, or assumptions about, underlying agent capabilities. How best to facilitate robust convention emergence in a timely manner is an area of ongoing research. Fixed strategy agents can be used to speed up and direct emergence. In particular, placing small numbers of fixed strategy agents at targeted locations within the network topology connecting agents has been shown to better facilitate convention emergence than untargeted placement. The heuristics used to choose these locations often make use of metrics derived from an agent's location within the topology.

In this paper, we initially considered organic un-influenced convention emergence in a dynamic network, using the topology model proposed by González et al. [8,9]. We showed that conventions emerge in a dynamic environment and that the average time taken for this is largely independent of the parameter settings used in the network model, provided the value of T_l/T_0, is above a threshold of approximately 4. Below this, the topology or agent lifespans are not conducive to any convention emergence occurring at all. This indicates that there is a minimum level of connectedness required in dynamic topologies for conventions to emerge.

We proposed a new placement heuristic, Life-Degree, that utilises information unique to dynamic topologies in its decision making process, allowing us to test the importance of that information. We contrasted this to the performance of the traditionally used placement heuristics.

We examined the scenario where fixed strategy agents are introduced early in the life of the system to direct and encourage faster convention emergence. We showed that, as in static networks, targeted placement offers better performance than untargeted. A small number of agents are able to influence a population much larger than themselves. We established that, in domains where it is possible to change the fixed strategy agents after selection, doing so offers small improvements in performance. In both settings, the most important aspect of selected agents was found to be their degree, ignoring their longevity. This was found to both increase the probability of a specific convention emerging as well as increasing the speed of that emergence.

Finally, we considered the destabilisation of already established conventions in dynamic networks. We found that destabilisation is more sensitive to the inclusion of agent lifespan than when using fixed strategy agents to establish a convention at the beginning of simulation. Choosing locations that will maximise an agent's influence, regardless of how long they will remain, is the most important aspect to consider when destabilising conventions in dynamic networks. Future work will investigate this further and examine if other features of dynamic networks offer beneficial information when selecting fixed strategy agents. We showed that the updating heuristics cause more destabilisation than the static heuristics and that this effect was much larger than the equivalent difference when encouraging initial convention emergence.

Overall, we have shown that convention emergence is possible in dynamic topologies and that many characteristics have direct parallels in static networks. We have shown that the degree of an agent is a major factor when choosing them and can be used to cause rapid convention emergence and destabilisation.

References

1. R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80:1095–1111, 1986.
2. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
3. C. Bicchieri, R. C. Jeffrey, and B. Skyrms. *The Dynamics of Norms*. Cambridge Studies in American Literature and Culture. Cambridge University Press, 1997.
4. J. Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1-2):171–185, 2002.
5. J. Delgado, J. M. Pujol, and R. Sangüesa. Emergence of coordination in scale-free networks. *Web Intelligence and Agent Systems*, 1(2):131–138, 2003.
6. H. Franks, N. Griffiths, and S. Anand. Learning agent influence in MAS with complex social networks. *Autonomous Agents and Multi-Agent Systems*, 28(5):836–866, 2014.
7. H. Franks, N. Griffiths, and A. Jhumka. Manipulating convention emergence using influencer agents. *Autonomous Agents and Multi-Agent Systems*, 26(3):315–353, 2013.
8. M. C. González, P. G. Lind, and H. J. Herrmann. Networks based on collisions among mobile agents. *Physica D: Nonlinear Phenomena*, 224(1-2):137–148, 2006.

9. M. C. González, P. G. Lind, and H. J. Herrmann. System of mobile agents to model social networks. *Phys. Rev. Lett.*, 96:088702, 2006.
10. N. Griffiths and S. S. Anand. The impact of social placement of non-learning agents on convention emergence. In *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1367–1368, 2012.
11. M. Kandori. Social norms and community enforcement. *The Review of Economic Studies*, 59(1):63–80, 1992.
12. J. Kittock. Emergent conventions and the structure of multi–agent systems. In *Lectures in Complex Systems: the Proceedings of the 1993 Complex Systems Summer School*, pages 507–521. Addison-Wesley, 1995.
13. J. Marchant, N. Griffiths, and M. Leeke. Destabilising conventions: Characterising the cost. In *Proceedings of the 8th IEEE International Conference on Self-Adaptive and Self-Organising Systems*, 2014.
14. J. Marchant, N. Griffiths, and M. Leeke. Convention emergence and influence in dynamic topologies. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (to appear)*, 2015.
15. J. Marchant, N. Griffiths, M. Leeke, and H. Franks. Destabilising conventions using temporary interventions. In *Proceedings of the 17th International Workshop on Coordination, Organisations, Institutions and Norms*, 2014.
16. M. Mihaylov, K. Tuyls, and A. Nowé. A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 28(5):749–778, 2014.
17. P. Mukherjee, S. Sen, and S. Airiau. Norm emergence with biased agents. *International Journal of Agent Technologies and Systems*, 1(2):71–84, 2009.
18. N. Salazar, J. A. Rodriguez-Aguilar, and J. L. Arcos. Robust coordination in large convention spaces. *AI Communications*, 23(4):357–372, 2010.
19. B. T. R. Savarimuthu, R. Arulanandam, and M. Purvis. Aspects of active norm learning and the effect of lying on norm emergence in agent societies. In D. Kinny, J. Y.-J. Hsu, G. Governatori, and A. K. Ghose, editors, *Agents in Principle, Agents in Practice*, pages 36–50. Springer, 2011.
20. B. T. R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis. Norm emergence in agent societies formed by dynamically changing networks. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 464–470, 2007.
21. S. Sen and S. Airiau. Emergence of norms through social learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1507–1512, 2007.
22. Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, 1997.
23. D. Villatoro, J. Sabater-Mir, and S. Sen. Social instruments for robust convention emergence. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 420–425, 2011.
24. D. Villatoro, S. Sen, and J. Sabater-Mir. Topology and memory effect on convention emergence. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 233–240, 2009.
25. A. Walker and M. Wooldridge. Understanding the emergence of conventions in multi-agent systems. In *International Conference on Multi-Agent Systems*, pages 384–389, 1995.
26. H. P. Young. The economics of convention. *The Journal of Economic Perspectives*, 10(2):105–122, 1996.

Association Formation Based on Reciprocity for Conflict Avoidance in Allocation Problems

Yuki Miyashita, Masashi Hayano and Toshiharu Sugawara

Department of Computer Science and Communications Engineering,
Waseda University, Tokyo 1698555, Japan
y.miyashita@isl.cs.waseda.ac.jp, m.hayano@isl.cs.waseda.ac.jp, sugawara@waseda.jp

Abstract. We describe the reciprocal agents that build virtual associations according to past cooperative work in a bottom-up manner and that allocate tasks or resources preferentially to agents in the same associations in busy large-scale distributed environments. The models of multi-agent systems (MASs) are often used to express the tasks that are done by teams of cooperative agents, so how each subtask is allocated to appropriate agents is an important issue. Particularly, in busy environments where multiple tasks are requested simultaneously and continuously, simple allocation methods result in conflicts, meaning that these methods attempt to allocate multiple tasks to one or a few capable agents. Thus, the system's performance degrades. To avoid such conflicts, we introduce reciprocal agents that cooperate with particular agents that have excellent mutual experience of cooperating as team members. They then autonomously build associations in which they try to form teams with their members for new incoming tasks. We introduce the N-agent team formation game, an abstract expression of allocating problems in MASs by eliminating unnecessary and complicated task and agent specifications, thereby identifying the fundamental mechanism to facilitate building and stably maintaining associations. We experimentally show that reciprocal agents can drastically reduce the number of conflicts, enabling tasks to be executed efficiently with teams. Finally, we analyze the reasons for such efficient allocations.

1 Introduction

Many computational tasks are completed by not just a single agent but by teams or groups of cooperative agents. For example, a task in service computing often dynamically composed of a number of service elements and can be achieved by allocating them to appropriate agents. These agents are usually software entities on the Internet created by different developers. We can find such applications not only in computer science such as ad hoc networks, e-commerce, and sensor networks, but also in other domains such as coalitional formation for tackling pollution control problems (economics and social science) [21] and group work in education (e.g., [22, 27]). In these applications, the efficient and effective formation of teams for doing tasks is vital for providing qualitative service in a timely manner. However, in open environments such as the Internet, centralized controls are often impractical because such information is too large and dynamic to process at a single point. Furthermore, numerous agents, which usually work

as delegates of companies, organizations, and individuals, are likely to block access to parts of the information to ensure security and confidentiality. Thus, autonomous distributed methods are required. However, forming a team in a large-scale MASs is costly because agents have to select team members from a large pool of agents. In addition, if tasks are numerous and appear simultaneously in distributed environments, many conflicts in allocations — meaning that multiple tasks will be allocated to a single or to a few capable agents — are likely to occur because centralized managers cannot be assumed to have complete latest information. Hence, they hinder team formation and degrade the entire performance.

A number of literature have discussed approaches to form teams and coalitions for group-based tasks, especially in the multi-agent systems context. For example, in the literature of coalition formation (e.g., [7, 13, 19]), methods have been proposed to find the combinatorial formation that provides the maximum social utility under the assumptions that characteristic functions for all possible groups are given. However, real-world applications often cannot assume characteristic functions in advance. Therefore, a few studies focus on identifying characteristic functions [18, 28]. However, they assume that their environments are static and not busy, so the characteristic functions take into account only a one-shot and static situation [18]. Another approach to team formation is market-based methods, such as the conventional contract net protocol and its extensions. However, they also assume that the system is not large and very busy; when it becomes busier and larger, the efficiency for forming teams, i.e., allocating the elements of the given tasks to agents, is severely degraded. More importantly, most of these studies did not take into account the conflicts in allocations in busy environments.

Often we form groups for doing tasks in the real world, and if conflicts in group formation occur and if no prior communications for pre-negotiation are possible, we first find reliable and dependable persons with whom to work. Such people are usually identified according to past reciprocity and an agreement for benefit distribution within the groups (e.g., [9]). Furthermore, if the opportunities for group work are frequent, we try to form collaborative relationships with these mutually reliable people. In an extreme case when group work with unreliable people is offered, we can refuse the offers for the sake of finding possible future proposals with reliable people and for punishing the proposing agents because of their past unreliable behavior [9, 10]. Although such behavior is irrational because of its self-interest, it can stabilize collaborative relationships and avoid the possibility of conflicts in team formations. Thus, we can expect stable benefits in the future through working with reliable agents.

This paper describes the first attempt to create a computational method for team formations that have fewer conflicts (thereby ensuring stability) and more acceptable benefit distributions among teams. This is done by introducing non-self-interested and irrational behavior — much like that in humans. Initially, agents form teams randomly, so many trials may fail due to conflicts or may result in unacceptable benefit distributions. However, through these trials, the agents mutually memorize the reciprocal and non-reciprocal behaviors by agents encountered in teams and also learn the appropriate behaviors for other reciprocal agents. Then, agents with enough experience in group work will identify which agents are dependable when necessary and virtually build collaboration relationships, called *associations* of agents, in a bottom-up manner. Agents

also learn how the benefits should be shared with collaborators in the associations. Conversely, agents that are found to be dependable by certain agents will try to behave as expected. Agents often behave irrationally like humans in this learning process because they may decline group work offers provided by undependable or first-time agents. However, from a long-term viewpoint, these behaviors and learning will facilitate agents building associations consisting of mutually dependable and trusted agents and will enable teams to be formed stably within the associations to which they belong. Such teams within an association may not be the best from the viewpoint of the optimality, but if they can complete tasks with the required quality, the effectiveness and less conflicts are more important in a large-scale and busy MAS. team

This paper is organized as follows: Section 2 discusses related work, and Section 3 describes the model of agents and the game of team formation by detailing an abstract resource allocation problem in a multi-agent system context. Then, in Section 4, we describe several types of agents, including reciprocal agents, and explain how they perform the games with building associations consisting of dependable agents. Section 5 shows how the performance of the game improved by learning of payoff distributions and by building associations based on the behavior in games. Finally, the paper is concluded in Section 7.

2 Related Work

Many studies on resource or task allocation have been conducted in multi-agent systems contexts. Resource and task allocation problems are usually formulated by using integer or linear programming techniques (e.g., [23]). These techniques are the centralized methods and are thus applicable only when all information is available at a single point. However, this assumption is often impractical in distributed environments. An important approach in such environments is coalitional formation based on cooperative game theory and teamwork [7, 19, 24, 25]. Although this work has numerous applications such as disaster control [2], sensor networks[13] and unmanned air vehicles [1], they assume coalitions for one-shot situations, so these are applicable only to static and unbusy environments. Furthermore, they assume that characteristic functions for coalitions are shared among agents in advance. However, this assumption is often implausible in real-world applications.

Research more related to ours in this approach is coalitional formation in dynamic environments [6, 16, 18, 28]. For example, Chalkiadakis and Boutilier [6] proposed stable coalitional formation in the framework of cooperative game theory using Bayesian reinforcement learning, but this method is not scalable enough. Kluschand and Gerber [18] proposed a dynamic coalition formation mechanism using rational agents, called DCF-A, in which fixed leader agents learn how coalitions should be formed. The leaders are the central points and DCF-A assumes that all agents are constantly available. Ye et al. [28] proposed a dynamic method in environments where agents are connected with a certain network structure. However, we focus on autonomous and bottom-up generation of a stable coalitional (or association) structure formation in busy environments where tasks continuously arrive at the MAS. Jones and Barber [16] proposed a bottom-up method that uses heuristics combining team formation strategies and task selection

strategies to adapt to dynamic environments. Faye et al. [8] proposed coalitional formation in environments where the availability of agents is unpredictable, although it was concerned with network applications. However, our study addresses how dependable agents are mutually recognized and make associations for stable task executions.

Of course, a lot of research on groups and reciprocity in human societies have been conducted in sociobiology and economics (e.g., [26]) and we wanted to utilize the findings in these research areas. Numerous studies attempted to explain non-self-interested behavior using reciprocity. The simplified meaning for this is that people do not engage in selfish actions towards and do not betray others who are reciprocal and cooperative, even if such selfish actions could result in higher payoffs for themselves [12, 11, 20]. Panchanathan and Boyd [20] stated that cooperation can be established from indirect reciprocity, meaning that people work together with certain persons and expect future rewards through cooperating with others [9]. The authors of [11, 12] insisted that fairness in cooperation produces non-self-interested behavior; agents do not betray relevant reciprocal agents because such a betray would be unfair. One important study related to our work is the results of a repeated ultimatum game [14] done by Fehr and Fischbacher [9] that showed how payoffs shared among collaborators affected the strategies in subsequent games. The same authors also found that punishment towards those who distribute unfair payoffs is frequently observed, although the punishment can be costly [10]; fairness and punishment are key points in continuing cooperation in an ultimatum game.

Group formation and selection are also related work. For example, Bowels et al. [5] insisted that people form groups because those belonging to groups have high probabilities to win races occurring in their societies (we believe the notion of *race* in [5] corresponds to *conflict* in our work). Bowles et al. [4] also investigated using agent-based simulations, and they found that groups and group-adapted behavior that may be individually costly evolved because group institutions can limit the fitness cost of the behavior. Bornstein and Yaniv [3] experimentally indicated that in the ultimatum game, people in a group can receive lower payoffs than in individual-based games but are nonetheless likely to accept the proposals in the group. The situations we addressed in this paper are quite similar to those of the repeated ultimatum game — more specifically, the dictator games [17], which is a variant of the ultimatum game — but we focus on algorithmic methods to understand how agents can autonomously form groups and how they become likely to accept group proposals; in our context, this means that conflict situations can be avoided by choosing group-based behavior.

3 Model and Problem

3.1 Overview of Allocation Problem

Our source of motivation in this paper is a continuous task or resource allocation problem in which a task consisting of a number of subtasks is executed by a number of agents that have sufficient resources to process the allocated subtasks [15]. Briefly, the problem is formulated as follows: Let $A = \{1, \ldots, n\}$ be the set of agents, and agent $i \in A$ has its resources expressed by $R_i = \{r_i^1, \ldots, r_i^p\}$, where p is the number of types of resources. Task $T = \{s_1, \ldots, s_K\}$ consists of a number of subtasks s_k. Some amounts

of resources are required to execute subtask s, so we identify it as $s = \{r_s^1, \ldots, r_s^p\}$, where r_s^k is the k-th resource required for s. Agent i can process s when its resources satisfy

$$r^k \geq r_s^k \quad (1 \leq \forall k \leq p), \tag{1}$$

and T is executed by a team of agents, but any agent can belong to only one team at a time. When the agents in the team satisfy Condition (1) for the given subtasks, the team can successfully execute T.

For v_o, a positive integer, tasks on average are given to the systems every tick, which is the time unit in our model. Let $Q = \{T_1, \ldots, T_l\}$ be the set of the given tasks. For task $T \in Q$, one agent works as a leader that is an initiator to form the team. The leader selects one agent (or a few) for each subtask in T, then sends it a solicitation message with a subtask to join the team. The agents that receive the messages select one of them and send back an acceptance message. If agents accepting the solicitation message satisfy Condition (1) for all subtasks, the team can execute T with a certain game duration. Then, the leader receives the payoffs for T and distributes them to members with a certain policy. In this process, a number of (capable) agents may receive multiple solicitation messages simultaneously or during execution. Because agents can belong to only one team simultaneously, such agents have to decline the rest of the solicitations. Thus, team formation may fail. This sort of conflict is common and frequent when the system is busy, so the performance is degraded.

We proposed the aforementioned task allocation method based on past successful cooperation in forming teams and achieved efficient team formation [15]. However, the success rate was insufficient to use in actual applications. One major reason for the low success rate was the conflicts in allocations. This kind of request for group work is often observed in human society, but we attempted to improve the success rate. For example, we sent join solicitations only to dependable people who we believe will probably accept them if they are inactive. Conversely, when we receive multiple solicitations, we tend to select the solicitation from the most reliable leader, meaning that that leader selects us over others. To stabilize such team formations, we often build a group, which is called an *association*, whose members consider each other to be reliable and dependable. Our purpose was to reduce these conflicts in task allocation by using the associations of computer systems from which leaders select the candidates for team members.

3.2 Abstract Model of Allocation Problem

We attempted to identify what information impacts on building virtual associations in a society of agents and how that information should be used, with the help of findings in other disciplines. For this purpose, we created an abstract of a model of an allocation problem with team formation in Section 3.1 by eliminating unnecessary specifications of tasks and agents, and we identified what the fundamental mechanisms were in building associations in a bottom-up manner.

The abstract version of the allocation problem with team formation is called the *team formation game* (henceforth referred to as TF game). It is similar to the repeated N-person ultimatum game ($N \geq 2$ is an integer) because we focus more on how teams

should be formed by distributing the received payoffs. More precisely, this is more similar to the N-person dictator game because member agents cannot refuse the payoffs proposed by the leader but can refuse the solicitation to join the TF game next time.

We introduce *N-agent Team Formation Game* as follows: Leader $l \in A$ selects $N-1$ agents from $A \setminus \{l\}$ and solicits them to form a team. Then, the solicited agents select zero or one solicitation (based on their own policy). If all solicited agents do not accept them, the game is deemed a failure and ends. Otherwise (if all agents accept), the game succeeds, and the formed team is retained for d ticks. After that, l receives the pre-defined payoffs $P > 0$. l picks up some payoffs from P in return for playing as the leader, and the remaining payoffs are distributed to all other members equally. Then, the game ends. We assume that agents cannot accept and attend multiple games simultaneously. The agent currently being engaged on TF games is called *active*; otherwise, it is called *inactive*. Every tick, v_o inactive agents are randomly selected and initiate the TF games as leaders. Then, this process is iterated. We propose a method to increase the success rates so that the number of TF games succeeding during a certain period is called the *game performance*, after this. Note that d corresponds to the processing time for the allocated task.

The findings in (socio-)biology and experimental economy discussed in Section 2 suggest that although humans are usually motivated by self-interest, fairness is a key feature for group-based activities; that is, people tend to behave fairly within a group. Often, they give punishments for unfair behaviors, though punishments incur some costs to themselves and, in this sense, do not represent rational behavior [9]. Therefore, we attempted to find a control method for agents to build associations. We herein show that these association can improve game performance by reducing conflicts.

4 Proposed Method: Reciprocal Agents and Associations

4.1 Reciprocal Agents

We introduce a *reciprocal* agent that is concerned with who are dependable, i.e., who are likely to accept forming teams in TF games and to distribute payoffs fairly based on the past reciprocal activities of other agents. Then, the agent tries to build associations of mutually dependable agents. A reciprocal agent is different from a cooperative agent in the sense that a reciprocal agent demonstrates cooperative attitudes to those that were cooperative in the past and may ignore or understate messages from non-reciprocal and unfair agents as punishment.

We introduce three learning parameters in reciprocal agents for N-agent TF games, *greediness*, the *threshold rate for dissatisfaction* (TRD), and *confidence degree*. The definition of confidence degree is described in the next section. The parameter of greediness of i, $0 \leq g_i \leq 2 \cdot 1/N$, determines that when i has worked as a leader of a successful team, i picks up $P \cdot g_i$, and so $P \cdot (1 - g_i)/(N-1)$ is distributed equally to other members. Agents want to earn more payoffs, so the higher g_i is, the better. However, other members may become dissatisfied. Note that the rate of even proportion is $1/N$, so we set the maximum value of greediness to double that.

Parameter TRD, $0 \leq Trd_i \leq 2 \cdot 1/N$, denotes the threshold for i's (dis)satisfaction of the received payoffs from leader j if

$$P \cdot (1 - g_j)/(N - 1) < Trd_i, \tag{2}$$

where i expresses the dissatisfaction to leader j. How the parameters of greediness and TRD are learned based on the game results and the received payoffs is described in Section 4.4.

4.2 Association and its formation

Agent i can belong to a number of *associations*, L, which are the sets of agents including i. Agent i knows $\boldsymbol{L}_i$ the collection of the associations it belongs to. We also assume that agents know the current state, which is active (attending another TF game) or inactive, for those in the same associations. We think that this assumption is reasonable if the number of agents in each association is low; actually, we experimentally show that it is quite low. Initially, i has a singleton association, so $\boldsymbol{L}_i = \{\{i\}\}$. We also define

$$\mathcal{L} = \cup_{i \in A} \boldsymbol{L}_i, \tag{3}$$

which is the collection of all associations. Note that if $L_2 \subset L_1$, L_2 is eliminated from $\boldsymbol{L}_i$ since L_2 is redundant. Agents working as a leader first select one of their associations and try to find the candidates of team members in it.

Reciprocal agent i has a set of parameters called a *confidence degree* (CD), $\{c_{ij} \mid j \in A \setminus \{i\}, 0 \leq c_{ij} \leq 1\}$, to extend or reduce the member of associations. Intuitively, the CD denotes how much agent i wants to form teams with $j \in A \setminus \{i\}$ again, and it is learned through j's past behavior to i using

$$c_{ij} = (1 - \alpha_c)c_{ij} + \alpha_c \cdot \lambda_{ij}, \tag{4}$$

where $0 < \alpha_c \ll 1$ is the learning rate, and λ_{ij} is defined according to the process of TF games as follows:
Case 1: If i worked as a leader and j accepted the solicitation from i, then $\lambda_{ij} = 1$, and if it refused the solicitation, $\lambda_{ij} = 0$.
Next, suppose that i worked as a member of a team whose leader is j.
Case 2: If the TF game succeeded, $\lambda_{ij} = 1$; otherwise, $\lambda_{ij} = 0$.
Case 3: Furthermore, if the game succeeded, i raises the CD values to other members by $\lambda_{ik} = 1$ for any $k(\neq i, j)$ in the team. Conversely, i lowers the CD for agent k by $\lambda_{ik} = 0$ if k refused the solicitation from j because this is a reason for the failure of the TF game. However, for agent k' who accepted the invitation, $c_{ik'}$ remains unchanged.

The association is extended or reduced as follows according to the CD values. Agent i started the process to invite non-associating agent j when $c_{ij} > F_c$, where F_c is the threshold value for invitation to the association. Such j is called i's *dependable* agent. For $\forall L \in \boldsymbol{L}_i$ s.t. $j \notin L$, if $c_{i'j} > F_c$ for more than half agent $i' \in L$, j is accepted to join L so $L = L \cup \{j\}$. Then, redundant associations are eliminated. Conversely, if $\exists j \in L$ s.t. the number of agents $i \in L$ whose confidence satisfies $c_{ij} < F_{exp}$, then j is expelled from L. Note that i may have low confidence for agent $j \in L_i$ ($c_{ij} < F_{exp}$)

in the shared association if other member in L_i have high CD values for j. Agent j is called an *undependable* agent for i when $c_{ij} < F_{exp}$. Agent i can also withdraw from association $L_i \in \boldsymbol{L}_i$ when the average of the CDs of other members is lower than F_{exp}, i.e.,

$$\sum_{j \in L_i \setminus \{i\}} c_{ij}/(|L_i| - 1) < F_{exp} \tag{5}$$

is held.

4.3 Forming Teams Based on Associations and Confidence Degree

A reciprocal agent plays TF games using an ε-greedy strategy as follows. Agent i working as a leader first selects one association, L_i, that has the most dependable inactive agents; this is possible because agents know the states of other agents in the common associations. If the number of dependable inactive agents in L_i is greater than or equal to $N-1$, i selects the $N-1$ agents from it according to a descending order of the CD values of i. If the number is smaller than $N-1$, the rest of the members are selected according to i's CD values. However, with probability ε, one selected member is replaced by another agent that is selected randomly.

Next, suppose that i is currently not a leader and has received a number of solicitation messages. Agent i first ignores the messages from undependable agents, even if it has received only such messages. This ignorance may be irrational behavior because accepting one solicitation may produce some payoffs, but we can think of it as a kind of punishment to the sender because the low CD value is the result of past unfair behavior. Agent i then selects one of the messages according to the CD values for senders with probability $1-\varepsilon$; otherwise i selects one randomly.

4.4 Response and Payoff Distribution Strategies in Reciprocal Agents

The CD values are updated using the new value of λ_{ij} after the TF game with the team members. This value is determined according to (not only the CD values of other agents but also) the responses to the specifications and the rates of the received payoffs. These responses and payoffs are decided according to the parameters of greediness and TRD. These values in agent i also learn to find the appropriate values using update functions:

$$g_i = (1 - \alpha_g) \cdot g_i + \alpha_g \cdot \delta_g \tag{6}$$

$$Trd_i = (1 - \alpha_{Trd}) \cdot Trd_i + \alpha_{Trd} \cdot \delta_{Trd_i}, \tag{7}$$

where α_g and α_{Trd} are the learning rates. We will describe when reciprocal agents update them and how δ_g and δ_{Trd_i} are decided.

The basic concept of reciprocal agents for greediness is that when the game succeeds, (1) they want to obtain larger payoffs only if no other members of the TF game express dissatisfaction and (2) the member agents learn the value of the greediness of the current leader because it is appropriate in this association. However, when the game fails (at least one agent declined the solicitation), (3) its leader's the greediness should

drop.[1]. The details are as follows. If leader i succeeds in the TF game with the members of M_i and no agents in M_i express dissatisfaction, then $j \in M_i \setminus \{i\}$ updates g_j with $\delta_g = g_i$, and i updates g_i with $\delta_g = 2/N$ using Formula (6). If a member expressed dissatisfaction or the game failed, i updates g_i with $\delta_g = 0$.

Parameter TRD is used to express the dissatisfaction with the payoffs received from the leader because agent i thinks that the payoffs are low and unfair. As mentioned previously, the dissatisfaction may lower the CD for other agents, resulting in punishments and/or the expulsion or withdrawal from joined associations in the future. Because agents in an association should have the same and similar TRD values, its values are adjusted as follows based on the response from members. Suppose that agent i is the leader of a successful TF game but agent j in the team expresses dissatisfaction. If the rate of payoffs that i distributed to j is higher than i's TRD, i.e., $Trd_i < (1 - g_i)/(N - 1)$, then Trd_i is updated with $\delta_{Trd_i} = (1-g_i)/(N-1)$. Furthermore, if j is the only agent expressing dissatisfaction, j's TRD is high, so the currently received payoffs have to be adjusted. Thus, Trd_i is updated with $\delta_{Trd_j} = (1 - g_i)/(N - 1)$. However, if i receives no invitation for a certain period (ω ticks), i assumes that other agents have low CD values for i because its TRD must be high, so the Trd_i is reduced by updating with $\delta_{Trd_j} = 0$.

4.5 Comparative agents

We introduce two types of agents, a self-interested agent and an *associating self-interested* (AS) agent, for comparison in the following experiments. Self-interested agents in this article behave so that they get more payoffs based on past game interactions and do not intend to build associations. AS agents try to build associations by estimating which ones are more beneficial according to past interactions, in addition to behaving to get more payoffs like self-interested agents.

We introduce two learning parameters for self-interested agent i. The *expected value of distributed payoffs* (EDP) e_{ij} for ($\forall j \in A \setminus \{i\}$) is the statistical value about how many payoffs can be expected when i accepts the solicitation from j. It is updated by

$$e_{ij} = (1 - \alpha_e) \cdot e_{ij} + \alpha_e \cdot v_j \tag{8}$$

after accepting the solicitation from j, where v_j is the received payoffs from j, and $0 < \alpha_e < 1$ is the learning rate. Note that $v_j = 0$ if the TF game failed. The parameter *expected acceptance rate* (EAR), h_{ij}, expresses the degree of acceptance of the solicitation by j; after leader i sends the solicitation message to j, h_{ij} is updated by

$$h_{ij} = (1 - \alpha_h) \cdot h_{ij} + \alpha_h \cdot \delta, \tag{9}$$

where $\delta = 1$ if j accepted the solicitation, and $\delta = 0$ otherwise. Parameter EDP is used when self-interested agents select one solicitation to pursue more payoffs. Parameter EAR is used to select more probable agents as members of TF games.

[1] Thus, the name of 'greediness' may not be appropriate because reciprocal agents adjust the values by observing the responses from other association members. However, this parameter is also defined self-interested agents below although it is adjusted in a different way (to increase the sharing payoffs). In the experiment, we compared how many payoffs is allocated to leaders in different types of agents. Thus, we would like to use the greediness in this paper.

Self-interested agent i as a leader selects members according to the descending order of EAR, h_{ij} ($j \in A$). When i plays as a member, i select one solicitation message based on the EDP values. In both situations, i selects a member agent and a solicitation message randomly with probability ε. Self-interested agents also have parameter greediness. However, they update the values of greediness using Formula (6) only when they work as leaders; if the game succeeded, $\delta_g = 1$; otherwise, $\delta_g = 0$.

AS agents additionally have CD values, and they try to build associations similar to reciprocal agents. The leader AS agent select the members from their associations, but the member AS agents selects the solicitation messages according to the EDP, instead of the CD values. Furthermore, because they are interested in which agent will directly provide more payoffs and which have no dissatisfaction related to unfair payoff distributions, the CD values are updated only in Cases 1 and 2 in Section 4.2.

5 Experiments

5.1 Experimental Setting

We investigates how the game performance, which is the success rates of TF games, improved over time in the society of reciprocal agents and compared the results with those in the society of self-interested or AS agents. We also compared the number of generated associations, their structures, and how learning parameters converged. The number of agents was 300 ($|A| = 300$), and twenty N-person TF games whose game duration d was 3 were initiated with twenty inactive agents selected randomly every tick ($v_o = 20$). We set $N = 4$, so the initial values of greediness and TRD were randomly selected between 0 and 0.5. The initial values of CD, EDP, and EAR were set to 0.5, 0.25, and 0.5, respectively. The threshold values were defined as $F_c = 0.7$ and $F_{exp} = 0.4$. Parameter ω was fixed to 10. All learning rates, α_c, α_e, α_h, α_{Trd}, and α_g were set to 0.05. The data indicated below are average values of 20 independent trials.

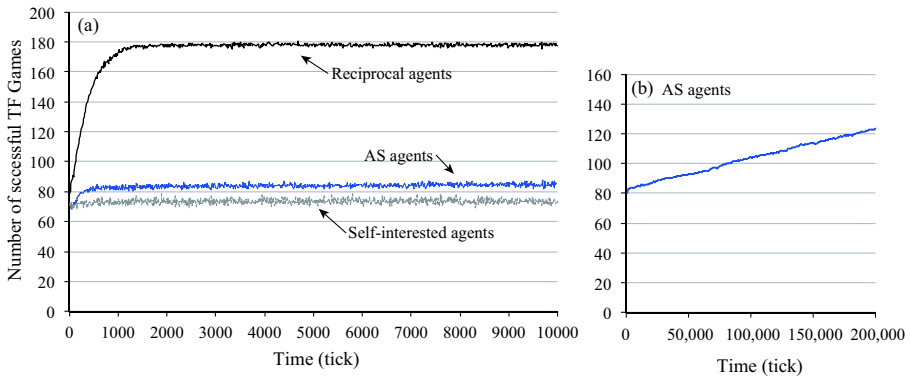


Fig. 1. Performance improvement over time.

5.2 Experimental Results – game performance and association structure

The number of successful TF games per 100 ticks are plotted in Fig. 1 (a). The figure indicates that reciprocal agents can perform TF games much more effectively than other agents. The self-interested agents pursue their own benefits. Thus, conflicts are likely to occur because they select members only based on locally learned results. The game performance (the number of successful TF games) of the AS agents was slightly better than that of the self-interested agents, so the advantage of associations seems to be limited. However, if we carefully look at Fig. 1 (a), we can see that the curve of AS agents did not converge yet; actually, when we continued the experiment to 200,000 ticks, the number of successful TF games gradually increased, as shown in Fig. 1 (b) but was still much lower than that of reciprocal agents. This suggested that associations could contribute to the game performance in AS agents but were formed very slowly; we will discuss the differences in the structures of associations. Hence, AS agents could not build associations within a reasonable time. Note that the numbers of successful TF games by self-interested and reciprocal agents did not vary after 10,000 ticks.

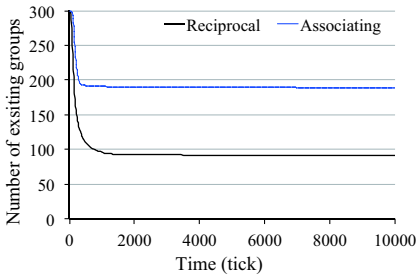


Fig. 2. Number of existing associations.

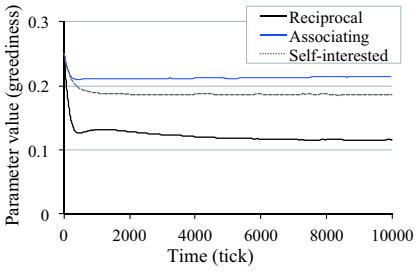


Fig. 3. Transition of average greediness.

Table 1. Association structures in societies of reciprocal and AS agents.

Agent type	size = 1	2	3	4	5	≥ 6	total
Reciprocal agent (at 10,000 ticks)	12.8	7.1	0.1	69.7	1.9	0	91.6
AS agent (at 10,000 ticks)	0	176.5	11.9	0.5	0	0	188.8
AS agent (at 200,000 ticks)	0	72.6	41.8	39.2	3.0	0	155.6

We investigated the association structure of agent societies to understand how associations contributed to the improvement in the game performance. Figure 2 plots the number of associations in the agent societies of reciprocal and AS agents. Table 1 lists the numbers of existing associations classified by size (the number of belonging agents) at the specified time. Initially, the number of associations was 300 because all agents belong to their own singleton association. Then, they gradually combined and formed

larger associations. Figure 2 indicates that for reciprocal agents, the number of associations quickly decreased. Then, they formed more associations, whose majority size was four (Table 1), which is the most ideal size of association. AS agents could also form associations whose size was four, but the formation was quite slow; even at 200,000 ticks, 4-size associations were only 39.2, which was only 60% of those of reciprocal agents.

The ideal game performance should be 200 per 10 ticks, but reciprocal agents succeeded in about 180 games. This difference was in part caused by the ε-greedy strategy because four agents in each game tried to randomly select agents with probability 0.05. Another reason was the existence of small-size (1, 2, or 3) associations; when agents in these associations initiate TF games, they may fail at high probabilities. However, these small size associations did not decrease after 10,000 ticks. Thus, merging these small size associations was one of the next objectives in our study.

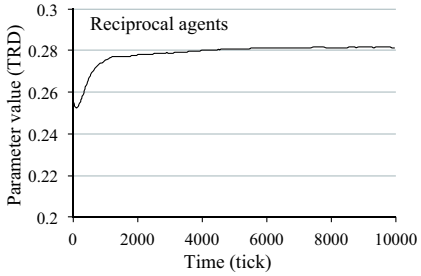


Fig. 4. Transition of average TRD.

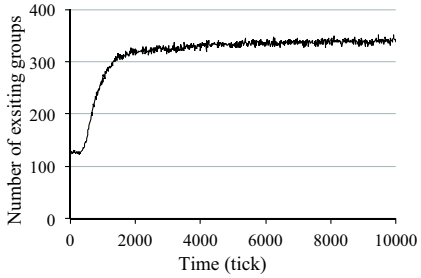


Fig. 5. Number of Dissatisfactions.

5.3 Payoff Distribution

Figures 3 and 4 indicate the values of control learning parameters deciding payoff distribution and the minimum acceptable payoff without declaring dissatisfaction. Note that greediness was defined for all types of agents. We found from Fig. 3 that the greediness values were 0.22, 0.19, and 0.12 at 10,000 ticks for the self-interested, AS, and reciprocal agents. Because a fair distribution is when $g_i = 0.25$, their own allocation seemed slightly small. The greediness in self-interested and AS agents were slightly higher. Particularly, the greediness values in AS agents continued to increase very slowly. For example, at 200,000 ticks, the average greediness value of AS agents was 0.31 and still increasing. In reciprocal agents, their greediness converged to a relatively low value (0.12). We can also see that the TRD value was approximately 0.28; these values of greediness and TRD are consistent with a small margin (because $0.28 \times 3 + 0.12 = 0.96 \approx 1$) to accept a payoff distribution without dissatisfaction.

We think that a message of dissatisfaction to leaders is very important to maintain associations stably. After 1500 ticks, no expulsion or withdrawal was observed, though we cannot show the graph here due to the page limit. However, Fig. 5 indicates

that many agents expressed dissatisfaction with payoff distributions. Particularly, their numbers stayed high after the association structure became stable. Thus, we can speculate that dissatisfaction could keep the values of greediness such that leaders did not increase them locally. In a society of reciprocal agents, none of the TRD and greediness values were important because all agents had the same chances to be leaders and members. For example, high greediness values cancelled each other if their greediness values were consistent and could be shared in the same association. The only possible unfairness was to increase them while ignoring other association members. Continuous learning of the CD values with the expression of dissatisfaction from members can prevent unfair behavior in the TF games.

6 Discussion

We previously conducted research to allocate subtasks to appropriate agents in a busy and large-scale distributed environment, and found that the conflicts in allocations were a major reason for a reduction in the entire efficiency [15]. We also found that building associations within the tasks to be done is an efficient and effective solution to this problem [15]. Teams within an association may not be the best from the viewpoint of the optimality, but if they can complete tasks with the required quality, an association-based team formation is acceptable, probably efficient, and practical in a large-scale system. The purpose of this study was to clarify the basic mechanism to build associations in a bottom-up manner by using an abstract and simplified model of allocation problems, called a TF game. We discuss what are important to build stable associations autonomously in this section.

Our experiments revealed that association-based team formation obviously contributes to the efficiency in TF games. We think two key pieces of information are needed to facilitate building associations in a MAS. First, the agent should memorize whom it worked with and share the information with other team members because the success of a TF game is owed to all members in the team. Conversely, the failure of a game is caused by an unacceptance by at least one agent, for one of two reasons: the leader's selection was not appropriate and/or one or more of the members was betrayed. These correspond to the learning of CD values in Case 3 (Section 4.2). This is quite different from the behavior of self-interested and rational agents that act based on which agents are likely to provide more payoffs directly.

Moreover, punishments and dissatisfaction contributed to building associations. The punishments, which represented refusals of the solicitation messages from agents whose confidence degree was low or expulsion of low confidence agents from associations in our context, affected the speed of building associations; actually, it could make the convergence faster, but it slightly reduced actual game performance. Dissatisfaction, which can be seen as advance notices of punishment, was necessary to make established associations stable, as mentioned previously.

Finally, we would like to discuss the convergent values of greediness and TRD. As previously mentioned, any value of greediness does not affect the received payoff values in reciprocal agents because they are always fair from a long-term viewpoint if the greediness and TRD values were consistent and shared in the associations. Hence,

if we want to distribute the payoffs fairly among team members, we can fix the value of TRD such as $Trd_i = 0.25(= 1/N)$ instead of learning these values. We think that reciprocal agents can converge to almost fair distribution, although this may affect the number of small-size associations.

7 Conclusion

We described reciprocal agents that build associations for team-based tasks to avoid possible conflicts in a large-scale, busy MAS. Our target was to perform task allocation problems efficiently. Thus, we first introduced an abstract form of this problem, called the team formation game, to identify what information and mechanism can facilitate building associations. We experimentally showed that the team formation method based on the associations the reciprocal agents belong to could successfully perform the games more efficiently than other types of agents: self-interested agents and associating self-interested agents. Finally, we discussed the important mechanism to build associations of reciprocal agents quickly. We have a number of research plans related to this paper. For example, we try to vary the team size, N, and the work load v_o and investigate how these parameters affect association structures. We also plan to explore this mechanism to combine the number of small-size associations to reduce the number of unworking associations.

References

1. Bardhan, R., Ghose, D.: Resource allocation and coalition formation for UAVs: A cooperative game approach. In: Proc. of the 2013 IEEE Int. Conf. on Control Applications. pp. 1200–1205 (2013)
2. Boloni, L., Khan, M., Turgut, D.: Agent-based coalition formation in disaster response applications. In: IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications. pp. 259–264 (2006)
3. Bornstein, G., Yaniv, I.: Individual and group behavior in the ultimatum game: Are groups more "rational" players? Experimental Economics 1(1), 101–108 (1998)
4. Bowles, S., Choi, J.K., Hopfensitz, A.: The co-evolution of individual behaviors and social institutions. Journal of Theoretical Biology 223(2), 135 – 147 (2003),
5. Bowles, S., Fehr, E., Gintis, H.: Strong procity may evolve with or without group selection. Theoretical Primatology, Project Newsletter 1(12), 1–25 (2003)
6. Chalkiadakis, G., Boutilier, C.: Sequentially optimal repeated coalition formation under uncertainty. Autonomous Agents and Multi-Agent Systems 24(3), 441–484 (2012).
7. Dunin-Keplicz, B.M., Verbrugge, R.: Teamwork in Multi-Agent Systems: A Formal Approach. Wiley Publishing, 1st edn. (2010)
8. Faye, P., Aknine, S., Sene, M., Sheory, O.: Stabilizing Agent's Interactions in Dynamic Contexts. In: Proc. of the IEEE 28th Int. Conf. on Advanced Information Networking and Applications (AINA 2014),. pp. 925–932 (2014)
9. Fehr, E., Fischbacher, U.: Why Social Preferences Matter - The Impact of Non-Selfish Motives on Competition. The Economic Journal 112(478), C1–C33 (2002)
10. Fehr, E., Fischbacher, U.: Third-party punishment and social norms. Evolution and Human Behavior 25(2), 63–87 (2004)

11. Fehr, E., Fischbacher, U., Gächter, S.: Strong reciprocity, human cooperation, and the enforcement of social norms. Human Nature 13(1), 1–25 (2002)
12. Gintis, H.: Strong reciprocity and human sociality. Journal of Theoretical Biology 206(2), 169 – 179 (2000)
13. Glinton, R., Scerri, P., Sycara, K.: Agent-based sensor coalition formation. In: Information Fusion, 2008 11th Int. Conf. on. pp. 1–7 (2008)
14. Güth, W., Schmittberger, R., Schwarze, B.: An experimental analysis of ultimatum bargaining. Journal of Economic Behavior & Organization 3(4), 367 – 388 (1982)
15. Hayano, M., Hamada, D., Sugawara, T.: Role and member selection in team formation using resource estimation for large-scale multi-agent systems. Neurocomputing 146(0), 164 – 172 (2014).
16. Jones, C., Barber, K.: Combining job and team selection heuristics. In: Hübner, J., Matson, E., Boissier, O., Dignum, V. (eds.) Coordination, Organizations, Institutions and Norms in Agent Systems IV, Lecture Notes in Computer Science, vol. 5428, pp. 33–47. Springer Berlin Heidelberg (2009).
17. Kahneman, D., Knetsch, J.L., Thaler, R.H.: Fairness and the Assumptions of Economics. The Journal of Business 59(4), S285–300 (1986),
18. Klusch, M., Gerber, A.: Dynamic coalition formation among rational agents. Intelligent Systems, IEEE 17(3), 42–47 (2002)
19. O.Sheholy, S.Kraus: Methods for task allocation via agent coalition formation. Journal of Artificial Intelligence 101, 165–200 (1998)
20. Panchanathan, K., Boyd, R.: Indirect reciprocity can stabilize cooperation without the second-order free rider problem. Nature 432(7016), 499–502 (2004)
21. Ray, D.: A game-theoretic perspective on coalition formation. The Lipsey lectures, Oxford University Press, Oxford (2007)
22. Shiba, Y., Sugawara, T.: Fair assessment of group work by mutual evaluation based on trust network. In: Proc. of 2014 IEEE Frontiers in Education Conf. (IEEE FIE 2014). pp. 821–827 (2014)
23. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press. (2008)
24. Sims, M., Goldman, C.V., Lesser, V.: Self-organization through bottom-up coalition formation. In: Proc. of the Second Int. Joint Conf. on Autonomous Agents and Multiagent Systems. pp. 867–874. ACM (2003)
25. Sless, L., Hazon, N., Kraus, S., Wooldridge, M.: Forming coalitions and facilitating relationships for completing tasks in social networks. In: Proc. of the 2014 Int. Conf. on Autonomous Agents and Multi-agent Systems. pp. 261–268. AAMAS '14, Int. Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2014),
26. Smith, J.M.: Group Selection. Quarterly Review of Biology 51(2), 277–283 (1976)
27. Wessner, M., Pfister, H.R.: Group Formation in Computer-supported Collaborative Learning. In: Proc. of the 2001 Int. ACM SIGGROUP Conf. on Supporting Group Work. pp. 24–31. GROUP '01, ACM, New York, NY, USA (2001)
28. Ye, D., Zhang, M., Sutanto, D.: Self-adaptation-based dynamic coalition formation in a distributed agent network: A mechanism and a brief survey. IEEE Transactions on Parallel and Distributed Systems 24(5), 1042–1051 (2013)

Interest-based Negotiation for Asset Sharing Policies

Christos Parizas[1], Geeth De Mel[2], Alun D. Preece[1], Murat Sensoy[3]
Seraphin B. Calo[2] and Tien Pham[4]

[1] School of Computer Science and Informatics, Cardiff University, UK
{C.Parizas, A.D.Preece}@cs.cardiff.ac.uk
[2] IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
{grdemel, scalo}@us.ibm.com
[3] Department of Computer Science, Ozyegin University, Istanbul, Turkey
murat.sensoy@ozyegin.edu.tr
[4] US Army Research Lab, Sensors & Electron Devices Directorate,
Adelphi MD, USA
tien.pham1.civ@mail.mil

Abstract. Resource sharing is an important but complex problem to be solved. The problem is exacerbated in a coalition context due to policy constraints placed on the resources. Thus, to effectively share resources, members of a coalition need to negotiate on policies and at times refine them to meet the needs of the operating environment. Towards achieving this goal, in this work we propose a novel policy negotiation mechanism based on the interest-based negotiation paradigm. Interest-based negotiation promotes collaboration when compared with more traditional negotiation approaches such as position-based negotiations.

1 Introduction

Negotiation is a form of interaction usually expressed as a dialogue between two or more parties with conflicting interests that try to achieve mutual agreement about the exchange of scarce resources, resolve points of difference and craft outcomes that satisfy various interests. In order to cooperate and search for mutual agreements, the involved parties make proposals, trade options and offer concessions. The automation of the negotiation process and its integration with autonomic, multi-agent environments has been well-researched over the last few decades [1, 2].

The approaches for automating negotiation can be classified into three major categories : (1) game theoretic (2) heuristic, and (3) argumentation based [2]. The first two approaches represent traditional bilateral negotiation protocols wherein each negotiation party exchanges offers aiming to satisfy their own interests. These are called position-based negotiations (PBN). In these approaches the participants attack the opposing parties' offers and try to convince them for the suitability of their own offers. Typically these approaches are formalized as

search problems in the space of possible deals by focusing on negotiation objectives. Argumentation-based negotiation (ABN) has been introduced as a means to enhance automated negotiation by exchanging richer information between negotiators. Interest-based negotiation (IBN) is a type of ABN where the agents exchange information about the goals that motivate their negotiation [3, 4]. IBN unlike PBN tackles the problem of negotiation by focusing on why negotiate for rather than on what to negotiate for and aims to lead negotiating parties to win-win solutions.

Multi-party teams are formed to support collective endeavors which otherwise would be difficult, if not impossible, to achieve by a single party. In order to support such activities, resources belonging to collaborating partners are shared among the team members; mechanisms to share resources in this context are actively and broadly explored in the research community. This is due to the impact that different sharing modifications (what to share, with who, when and under what conditions) can bring into the collaboration, with respect to domains such as security, privacy and performance to name only a few. Consider the following: a) crisis management situations where responders affiliated with different national and organizational groups form coalitions and share resources (e.g. sensors, network connectivity and data storage) in an ad-hoc manner, in order to provide humanitarian assistance; b) resource sharing in corporate environments such as the recent *MobileFirst*[5] partnership between IBM and Apple where cloud and other services are shared in a daily basis; or c) a short-lived opportunistic mobile network comprised of a few peer members, established for message routing or data sharing. In all these cases access control mechanisms that specify resource sharing need to be implemented. A suitable mechanism for managing access control on resources of such systems is the Policy-based Management System (PBMS).

This work presents a framework for enabling authorization policy negotiation in multi-party, cooperative and dynamic environments. This framework is aimed at policy makers who are not necessarily experts in IT or negotiation techniques, responsible for modifying policies responding to situational changes. To the best of our knowledge there is no mature work done on policy negotiation, while the vast majority of negotiation work in multi-agent environments: a) utilizes PBN approaches and b) invariably ignores the special characteristics of multi-party, collaborative environments. In this work we propose a novel, interest-based policy negotiation framework. It is our belief that by understanding the negotiating parties interests and crafting options that can meet their requirements, IBN could provide a negotiation mechanism which promotes good collaboration unlike PBN, which creates an adversarial negotiation atmosphere. Moreover PBN with its fixed, opposing positions is a cumbersome negotiation method to cope with dynamic environments [2]. The proposed negotiation framework can operate in parallel to a PBMS. It considers an approach that proposes modification of strict policies, in order to maximize overall usability of collab-

[5] http://www.ibm.com/mobilefirst/us/en/

orating assets while remaining faithful to existing authorization policies. The main contributions of this work are as follows:

- definition of an interest-based authorization policy negotiation model
- specification of an architecture for its integration with PBMS
- demonstration of its application on a user friendly policy representation
- presentation of a walkthrough for its execution utilizing a policy negotiation scenario

The remainder of the paper is organized as follows: in Section 2 we discuss previous literature on policy negotiation approaches and in Section 3 we present a walkthrough of a policy negotiation scenario. Section 4 describes the policy negotiation framework, the policy language, and its interface to PBMS by means of an architectural overview. Section 5 presents the algorithmic steps for IBN achievement through policy refinement. We conclude the document in Section 6 by summarizing our contribution and outlining future research directions.

2 Related Work

The first computer applications for supporting bilateral negotiations were developed in late 1960s [5]. The reason for their emergence was to assist human negotiators to overcome weaknesses related to negotiation process such as cognitive biases, emotional risks, and their inability to manage complex negotiation environments. Although there is rich literature on negotiation protocols in autonomous, multi-agent environments, there is very limited and no mature work done on policy negotiation. We see the role of policies in managing large, complex and dynamic systems as of a high importance and the existence of sophisticated ways to do so imperative. We believe that the integration of an effective negotiation mechanism on a PBMS works towards this direction. Moreover, no work had previously attempted to bring the IBN paradigm into policy negotiation.

The authors of [6] present requirements of policy languages which deal with trust negotiation and focuses on the technical aspects and properties of trust models to effectively evaluate access requests. It does not depend on any aspects of policy negotiation and the scenarios it deals with are less dynamic compared to our problem domain. [7] proposes an architecture that combines a policy-based management mechanism for evaluating privacy policy rules with a policy negotiation roadmap. The work is very generic and does not provide clear evidence of any effectiveness of the proposed approach, while lacking any evaluation. [8] is one of the first works that looks into policy negotiation and covers the area in depth. It also looks into collaborating environments and introduces the notion of ABN in policy negotiation. However it focuses on a very specific application domain in which it deals with writing insurance policies while maintaining a common and collaborative knowledge base.

The work discussed on [9] has several similarities to our work; it deals with cooperating environments and a PBMS is employed in support of service composition in a distributed setting. The authors have used a negotiation framework to

effectively compose services. Its main difference with the work proposed herein is that the objective of the negotiation performed in [9] is the services that are managed by policies, not the policies themselves. We believe that in order to decrease the management overhead the objective of negotiation should be the policies. This is because policies are the core of PBMS and the logical component where the systems management resides. Finally, [10] proposes a policy negotiation approach and presents its architecture. It lacks any effectiveness evaluation while it does not consider either multi-partner, dynamic environments or ABN and IBN paradigms.

3 Interest-based Policy Negotiation Scenario

Below we provide illustrative scenarios to motivate the use of IBN in policy negotiation in resource sharing situations. In Subsection 3.1 we revisit the classic orange scenario discussed in best-selling *Getting to YES* [11] and then expand it to a mobile resource sharing scenario in Subsection 3.2.

3.1 The Chefs-Orange Scenario

Two chefs who work in the same kitchen both want to use orange for their recipes. Unfortunately there is only one orange left in the kitchen. Instead of starting negotiating on who is going to get the orange (as in a PBN, zero-sum approach), the two chefs opt to follow the IBN approach. Thus, they ask each other why they need the orange. In other words they try to better understand their underlying goals of using the orange. Answering the why question it turns out that one chef needs only the oranges flesh (to execute a sauce recipe) while the other needs only its peel (for executing a dessert recipe) and so they share the orange accordingly achieving a win-win negotiation outcome.

3.2 Authorization Policy Negotiation

An individual P2 wants to access a smartphone device SMD owned by an individual P1. However, P1 has a set of restrictions which are captured by policy set R on how to share SMD with other people. These restrictions may reflect privacy concerns (e.g. by accessing their smartphone one can have access to their photos), security, and so forth. For the sake of clarity, in this example, we assume that the set R contains the following policy constraint R1: do not share the device SMD with anyone else but its owner P1. When P2 asks for permission to use the physical device SMD, R1 prohibits this action. Ostensibly there is little room for negotiation here, if one follows a PBN approach with the current set of policies.

However by applying IBN and trying to understand the underlying interests of the involving parties, we believe the situation could be handled in a satisfactory manner for both parties. For example, asking the why question it turns out

that P2 needs a data service (as opposed to the physical device) in order to execute the task *Email submission* and P1 does not mind sharing a data connection as a hotspot with a trusted party; if P1 could get to know why P2 needs the device, the situation could be solved to the satisfaction of both parties. All an IBN mechanism needs to do in this case is to introduce another policy – actually a refinement of the existing policy – to R1 to say that data service can be shared among trusted parties. Thus, we argue that in such cases by understanding the situation and broadening the space of possible negotiation deals, one can reach a win-win solution.

The intuition behind ABN is that the negotiating parties can improve the way they negotiate by exchanging explicit information about their intentions. This information exchange reveals unknown, non-shared, incomplete, and imprecise information about the underlying attitudes of the parties involved in the negotiation [12]. As stated earlier, IBN is a type of ABN where the negotiating parties exchange information about their negotiation goals, which then guide the negotiation process. Thus, the why party of the intention is of major importance when compared with the what part. Finally, we would say that the IBN is more of a negotiation shortcut method rather than a typical negotiation process. By attacking the problem of negotiation, IBN skips the proposals making, the options trading and the need for negotiating parties to offer concession as in PBN cases. In the next section, we shall introduce our IBN-based policy framework and provide our intuition behind the approach.

4 Interest-based Policy Negotiation Framework

The design and development of frameworks for establishing negotiation needs to achieve some desirable outcomes that are secured by meeting a set of systematic properties: guaranteed success (i.e., negotiation protocol that guarantees agreement), simplicity (i.e., easy for the optimal decision to be determined by participants), maximizing social welfare (i.e., maximization of the utilities sum of negotiation participants) to name a few [13]. The main objective of the negotiation framework we propose is to maximize social welfare.

In environments that often suffer from asset scarcity (demand exceeds supply), and many tasks may be competing for the same resource like the ones described in Section 1, paragraph 3, the formation of coalitions offers alleviation by bringing more resources to the table. The relationships between coalition parties in those scenarios are mostly peer-to-peer (P2P). However, we do not assume fully cooperative scenarios. Partners often pursue cooperation but they do not want to share sensitive intelligence that can deliver greater value to the opponents [14]. In the literature this kind of relationship model, where parties have cooperative and competitive attitudes from time to time, is called coopetition [15]. The PBMS and its sets of policies is in charge here, playing a regulative role in order to keep balance between asset sharing and asset "protection".

The more strict the partners' policies are, the higher the barriers towards collaboration are set. This is where the IBN mechanism comes in, trying to

lower these barriers in order to establish better collaboration trough asset sharing (i.e., increase overall the number of executed tasks and thus increase the social welfare) while maintaining the compromise from the asset owners point of view at the same levels.

The framework presented herein allows negotiation on policies with minimal human intervention. In traditional system management, policies associated with PBMS are static (or rarely change); these systems, however, fail miserably in dynamic environments where policies need to adopt according to situational changes. We note that it is not prudent to assume human operators in these environments can effectively be on top of every change to manage PBMS(s) effectively; they require automated assistance.

Summarizing its contribution, the IBN negotiation framework considers a cooperative negotiation approach which modifies strict policies aiming to a) maximize social welfare by increasing the overall usability of collaborating assets while b) remaining faithful to existing authorization policies, maintaining their core trends. Utilizing such a tool, a multilateral policy transformation can be achieved considering multi-party input and criteria for the benefit of the coalition. Each negotiation session considers sets of two negotiators (bilateral negotiation approach). The issue that needs to be settled during any negotiation process is the granting (or not) of access to non-sharable assets. From that perspective the framework deals with single-attribute negotiations.

4.1 Policies Under Negotiation

Several policy-based management systems that utilize different policy languages have been proposed in the literature. KAoS is a management tool for governing software agent behavior in grid computing using an ontological representation encoded in OWL [16]. Ponder is an object-oriented policy language used for managing systems and networks [17] while XACML is the OASIS standard access control policy language for web services [18]. The proposed policy negotiation framework is applied on authorization policies expressed in the Controlled English (CE) policy language [19]. CE policy language is an ontological approach that uses a Controlled Natural Language (CNL) for defining a policy representation that is both human-friendly (CNL representation) and unambiguous for computers (using a CE reasoner) [20]. CE is used to define domain models that describe the system to be managed. The domain models take the form of concept definitions and comprise objects, their properties, and the relationships among them. These domain model components are the building blocks of the attribute-based CE policy language.

Each policy rule follows the if-condition(s)-then-action form and consists of four basic grammatical blocks as shown below:

- **Subject:** specifies the entities (human/machine) which interpret obligation policies or can access resources in authorization policies
- **Action:** what must be performed for obligations and what is permitted for authorization

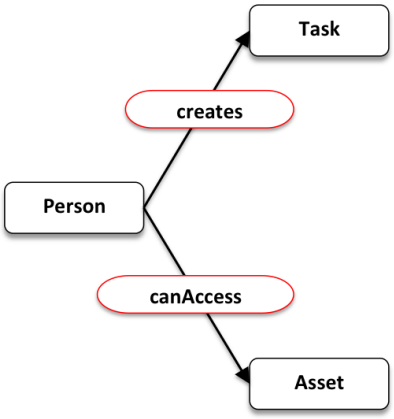


Fig. 1. Authorization Policy Negotiation Scenario: Domain Model

- **Target:** objects on which actions are to be performed
- **Constraints:** boolean conditions

The utilization of CE here is two-folde. It does not only is the user friendly formal representation of the system to be managed but also helps decision makers who lack technical expertise to understand in a more transparent way the complexities associated with policy negotiation. Figure 1 provides a graphical depiction of the CE-based domain model, which describes the smartphone access scenario of Section 3.2 , while the CE representation of policy R1 is shown below.

```
Policy R1
If
( there is an asset A named SMD ) and
( there is a person P named P1 )
then
( the person P canAccess the asset A )
.
```

4.2 The IBN in Asset Sharing process

The role of policies in managing a system is to guide its actions, towards behaviors that would secure optimal systems outcomes. Authorization policies manage actions of both, hard (sensing devices, distributed databases, smartphone devices) and soft resources (human-in-the-loop asset owners/requestors). Different users have different rights, relationships and interests in regards to deployed resources. Non-owner users want to gain access to the resources in order to serve their tasks needs, while owners want to protect their resources from unauthorized use. There is a monopolistic asset usage case. The proposed approach considers both concerns in a single mechanism providing a framework that pursues a

win-win negotiation outcome for any sets of negotiators. In other words, it tries through negotiation to redefine what is a suboptimal system outcome given: a) the currently-deployed resources and b) the tasks needs of the system that is managed.

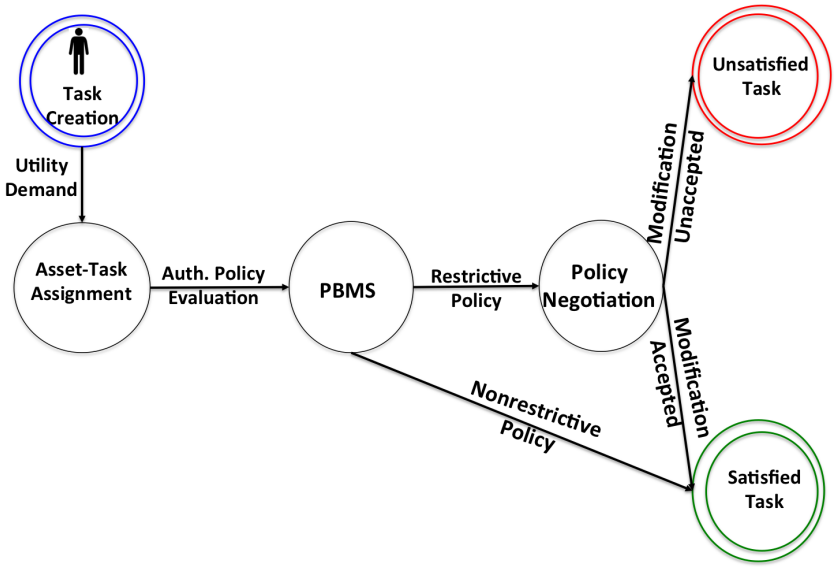


Fig. 2. Interest-based policy negotiation and task implementation

The finite state diagram of Figure 2 provides a depiction of the role the policy negotiation framework plays in the tasks implementation of collective endeavors. The human task creator, wanting to serve their appetite for information, creates tasks, which require a utility demand. The asset-task assignment component is in charge of optimizing the task utility by allocating the appropriate resources (information-providing assets) to each task. The PBMS component is responsible then for evaluating and enforcing authorization policies developed by multi-party collaborators. In the case of a non-restrictive authorization policy the task creator gets their task served. If the policy rule is restrictive, the policy negotiation component takes over. It modifies the policy rule accordingly, and passes it to the asset owner for confirmation. Depending on the asset owners decision the task is either satisfied or unsatisfied.

4.3 IBN Enabled PBMS

The policy negotiation framework can be integrated into a PBMS as a plug-in, enabling negotiation in policy enforcement process. A PBMS, as defined by stan-

dards organizations such as IETF and DMTF, consists of four basic components as shown in Figure 3: a) the policy management tool, b) the policy repository, c) the policy enforcement point (PEP), and d) the policy decision point (PDP) [21]. The policy management tool is the entry point through which policy makers define authorization policies to be enforced by the system. The policy repository is the component where the policies generated by the management tool are stored (step A1). PEP is the logical component that can take actions on enforcing the policies. Given the access request conditions, the PEP contacts PDP (step A2), which is then responsible for fetching the necessary polices from the policy repository (step A3, A4), evaluates them and decides which of them need to be enforced on PEP (step A5).

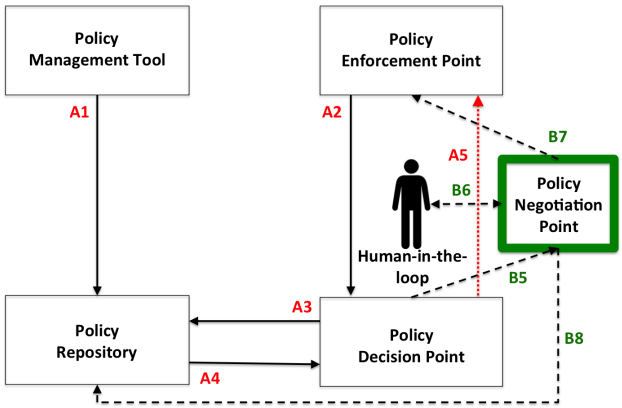


Fig. 3. IBN extended PBMS

In addition to the four basic PBMS elements, Figure 3 also includes a human-in-the-loop element, representing the roles played by the asset requestor and owner in the negotiation process. The additional component where the IBN framework resides is called the Policy Negotiation Point (PNP) and lies between the PEP and PDP, interfacing also with the human-in-the-loop element. As mentioned before, the PNP is triggered to attempt to modify authorization policies when a user creates a task that cannot be served due to restrictive policies. The dashed lines show optional communication between the components which is only established when a policy negotiation incident occurs. The red numbered parts of the figure (flow paths which are prefixed by As) describe the PBMS operational flow, while the green parts (flow paths which are prefixed by Bs) replace step A5 (red, dotted line) with the policy negotiation extension. Note that the separation between the components can be only logical when they reside in the same physical device. When PNP detects a restrictive policy (step B5) it

modifies it following the steps described at the following section and passes it to the asset owner for confirmation (step B6). If the asset owner confirms the replacement, the proposed policy is then enforced on the PEP (step B7) and it is also stored in the policy repository replacing its predecessor (step B8). Otherwise step A5 is executed as before.

5 Achieving IBN through Policy Refinement

In general, negotiation protocols contain the set of rules that manage the interaction between negotiating parties [2]. These rules define who is permitted to participate in the negotiation process and under what conditions (i.e. negotiating and any non-negotiating third parties). The rules also manage the participants actions throughout the process. In addition they define the decision of the negotiators towards the proposals.

The negotiating parties in our scenario as mentioned before are essentially decision makers who generally lack negotiation expertise. Thus the IBN mechanism tries to take, as much as possible, the negotiation weight off their shoulders rather than providing them the means for making proposals and trade options themselves. However, it does not exclude them completely from the negotiation process as in fully automated models. To achieve such behavior it simply applies the IBN principles described in Chefs-Orange scenario of Section 3.1, exploiting the domain models semantics, the semantics of the polices and the seamless relation between them as they both share the same CE representation.

The objective of the negotiation is the restrictive policies themselves. Asking the why question like in Chefs-Orange scenario to the requestor side, the PNP gets as a reply the reason why they need the asset for. Asking the why question to the asset owners/policy authors side, it gets the reasons why they do not want to grant access to their assets respectively. The prerequisite for the PNP here is to have full and accurate knowledge of the managed system. This is achieved by having unlimited and unconditional access to both domain model and policies. Unlike the majority of the proposed PBN approaches, the human-in-the-loop negotiators in our case are ignorant of the preferences of their opponents, while their knowledge in terms of the domain model reaches only the ground of their own expertise.

Utilizing CE for the formal representation of the environment to be managed, and as the language for expressing policies, the IBN, human-machine communication (i.e. communication between PNP and non-IT expert negotiators) for exchanging information regarding the negotiation is a transparently achievable task. The CE human-machine communication has been described in previous work [22]. However, trying to automate as much as possible the negotiation process, the why question is rather rhetorical here. In the requestors case the answer to the why question is quite simple and straightforward and the PNP is aware of it just by looking at the domain model. The asset requestor clearly wants to access the asset in order to execute their task. Hence, a desired negotiation outcome as far as the requestor is concerned, is the derivation of a policy that

has them included in the set of *Subject* policy block with positive access (i.e., canAccess) *Action* to a *Target* set that includes the prohibited asset capable of serving their task's needs.

Inferring the answer to the why question from the asset owners side for understanding their interests and broadening the negotiation space is a more challenging task. In general any application of authorization systems aims to specify access rights to resources. Thus, a simple answer would be including the reasons why they want to decline access rights to their own resources. Looking carefully at the policy, these reasons are basically described from the policys *Constraints* block. The policy R1 of Section 4.1 is rather a simple one referring deliberately to a simple scenario and this might not be easily inferred. Considering other more complex policy rules with several conditions describing constraints such as the age of the requestor or their expertise this is easier inferred.

However this is not exactly the answer to the why question we are looking for here. Considering the policies as the means for guiding systems actions towards behaviors to achieve optimal outcomes, the *Constraints* policy block refers to the actions level of the policy. Our focus here is on the higher level, this of the systems behavior. Focusing on a higher level, gives us the agility to find different policies as far as the actions is concerned, that provides the same functionality in terms of behavior; and the different policies we are looking for are those which serve the needs of the asset requestors as well. Achieving this goal we achieve a win-win negotiation outcome like the one described in Chefs-Orange scenario. The next four steps describe the process to reach such an outcome.

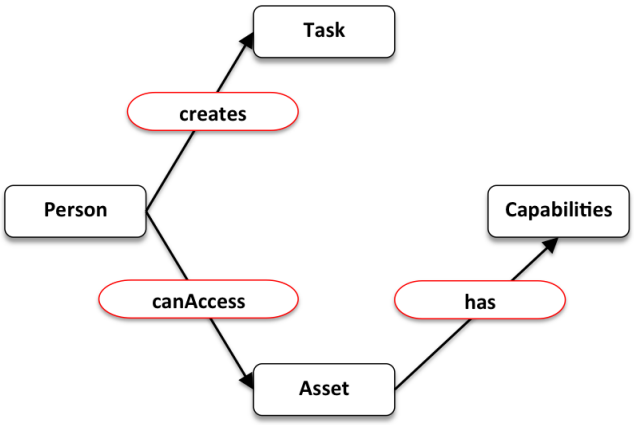


Fig. 4. Ontology Modification: Step one

Step 1: The simplistic domain model of Figure 1 presents only the concepts involved in the smartphone scenario of Section 3.2 and their relationships. It hides however their properties. Assume that the concept Asset has a property named *Provided capability* and that the Assset instance named SMD has the

Provided capability property named *Tethering*. Thus, the policy R1 by denying access to SMD, it denies access to any of SMDs provided capability as well. The IBN process starts taking as input the policy's *Target* block first. Trying to broaden the negotiation space in order to find alternative policies that satisfy both negotiators it separates the SMD from its Provided capability property and updates accordingly the ontology as shown in Figure 4 generating the respective CE sentences. The concept Capability and the respective relationship between Asset and Capability is now created.

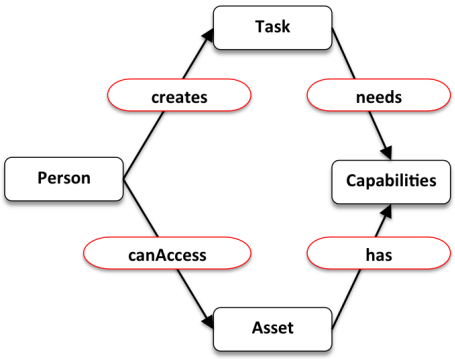


Fig. 5. Ontology Modification: Step two

Step 2: Each Task of Figure 1 requires a set of capabilities in order to be served. The concept Task has a property named *Required capability* and the Task instance *Email submission* has a number of required capabilities including that of *Tethering*. The second step of IBN process gets as input the Task and separates it from its Required capability property and updates accordingly the ontology as shown in Figure 5 generating the respective CE sentences as in Step 1.

Step 3: Often the tasks' capability needs might span outside the capabilities offered by one particular asset (e.g., a task might need to utilize capabilities provided by a number of assets). The IBN process, taking input from the previous two steps makes the matching between Asset's provided capabilities and Task's required capabilities. It matches this way the subset of the prohibited SMD's properties that are needed for the implementation of the desired Task and updates accordingly the ontology as shown in Figure 6 generating the respective CE sentences as in Step 1. The asset requestor now can access a subset/subsystem of asset that of its provided capability.

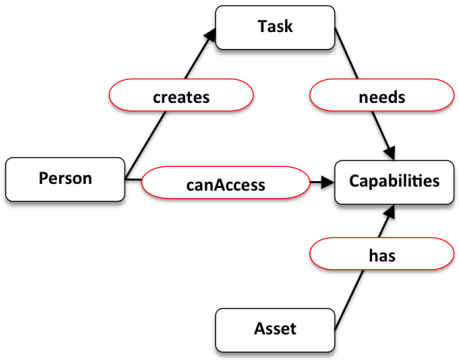


Fig. 6. Ontology Modification: Step four

Step 4: This step performs the *policy refinement*[6]. The asset requestor (i.e., P2) is the *Subject* block of the refined policy, which has as *Action* block a positive authorization action (i.e., canAccess) and its *Target* block contains, the provided by the prohibited Asset and required by the desired Task Capability (i.e., Tethering). The CE refined policy *R1-Refined* below is passed then to the asset owner for approval.

```
Policy R1-Refined
if
  ( there is a capability C named Tethering ) and
  ( there is a person P named P2 )
then
  ( the person P canAccess the capability C ).
```

The asset owner P1 is in charge of confirming or not the replacement of policy R1 from the proposed policy R1-Refined. In the case of confirmation the refined policy is then enforced on SMD providing access to SMD's tethering capability, and is also stored in the policy repository replacing its predecessor. The successful completion of IBN leads the negotiating parties to a win-win negotiation, with the asset requestor getting the task of *Email submission* served and the asset owner prohibiting any physical access to SMD.

6 Conclusion & Future Work

In summary the proposed IBN framework provides an effective policy negotiation mechanism for revising asset sharing policies in dynamic, multi-party environments. The framework is seamlessly interfaced with standardized PBMS and it

[6] Note that the term policy refinement herein refers to a different process than the policy refinement in [23], which describes the process of interpreting more general, business layer policies to more specific, system layer ones.

provides means to directly negotiate with policies. Our belief is that this is an important feature to have as PBMS is where the core components of the systems management logic resides. Moreover the IBN approach fits in multi-party environments where collaboration is promoted to achieve mutually satisfactory negotiation outcomes. Finally, utilizing CE-based policies in the framework eases the burden of the non-technical user in managing the PBMS and negotiate on them. As for the future research, there are plans for extending the IBN steps with regards to broadening the negotiation space considering components such as the users and the tasks of the system to be managed. In addition we plan to evaluate the proposed policy negotiation framework a) by conducting human-lead experiments, and b) by running simulations and comparing the results with respect to PBN approaches, especially in collaborative setting.

Acknowledgment: This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. D. Walton and E. Krabbe, "Commitment in dialogue: Basic concepts of interpersonal reasoning." *Albany, New York: SUNY Press*, 1995.
2. N. Jennings *et al.*, "Automated negotiation: Prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.
3. R. Fisher and W. Ury, "Getting to yes: Negotiating agreement without giving in," *New York: Penguin Books*, 1983.
4. P. Pasquier *et al.*, "An empirical study of interest-based negotiation," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 2, pp. 249–288, 2011.
5. M. Delaney, A. Foroughi, and W. Perkins, "An empirical study of the efficacy of a computerized negotiation support system (nss)," *Decision Support Systems*, vol. 20, no. 3, pp. 185–197, 1997.
6. K. Seamons *et al.*, "Requirements for policy languages for trust negotiation," *3rd International Workshop on Policies for Distributed Systems and Networks, POLICY 2002*, pp. 68–79, 2002.
7. I. Jang, W. Shi, and H. Yoo, "Policy negotiation system architecture for privacy protection," *Proceedings - 4th International Conference on Networked Computing and Advanced Information Management, NCM 2008*, vol. 2, pp. 592–597, 2008.
8. M. Comuzzi and C. Francalanci, "Agent-based negotiation in cooperative processes: Automatic support to underwriting insurance policies," *ACM International Conference Proceeding Series*, vol. 60, pp. 121–129, 2004.
9. J.-F. Tang and X.-L. Xu, "An adaptive model of service composition based on policy driven and multi-agent negotiation," *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, vol. 2006, pp. 113–118, 2006.

10. V. Cheng, P. Hung, and D. Chiu, "Enabling web services policy negotiation with privacy preserved using xacml," *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2007.
11. R. Fisher *et al.*, *Getting to yes: Negotiating an agreement without giving in; 2nd repr. ed.* London: Random House Business Books, 1999.
12. I. Rahwan *et al.*, "Argumentation-based negotiation," *Knowledge Engineering Review*, vol. 18, no. 4, pp. 343–375, 2003.
13. T. Sandholm, "Distributed rational decision making," *Multiagent Systems (ed. G. Weiss) MIT Press*, pp. 201–258, 1999.
14. C. Loebecke, P. C. Van Fenema, and P. Powell, "Co-opetition and knowledge transfer," *SIGMIS Database*, vol. 30, no. 2, pp. 14–25, 1999.
15. A. Brandenburger and A. Nalebuff, "Co-opetition," *New York: Doubleday*, 1997.
16. A. Uszok *et al.*, "Kaos policy management for semantic web services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 32–41, 2004.
17. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," *In proceedings of Workshop on Policies for Distributed Systems and Networks*, 2001.
18. OASIS. (2013) extensible access control markup language (xacml) version 3.0. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf
19. C. Parizas, D. Pizzocaro, A. Preece, and P. Zerfos, "Managing isr sharing policies at the network edge using controlled english," *In proceedings of Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IV*, 2013.
20. D. Mott, "Summary of controlled english," *ITACS*, 2010.
21. R. Yavatkar, K. Pendarakis, and R. Guerin. (2000) A framework for policy-based admission control. [Online]. Available: http://www.ietf.org/rfc/rfc2753.txt
22. A. Preece, D. Braines, D. Pizzocaro, and C. Parizas, "Human-machine conversations to support mission-oriented information provision," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, pp. 43–49, 2013.
23. A. Bandara, E. Lupu, J. Moffett, and A. Russo, "A goal-based approach to policy refinement," *Proceedings - Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY 2004*, pp. 229–239, 2004.

The Open Agent Society: A Retrospective and Future Perspective

Jeremy Pitt[1] and Alexander Artikis[2,3]

[1]Imperial College London, Exhibition Road, SW7 2BT, UK
[2]University of Piraeus, Greece
[3]NCSR 'Demokritos', Athens 15310, Greece

Abstract. It is now more than 10 years since the EU FET project ALFEBIITE finished, during which it's researchers made pioneering contributions to (inter alia) formal models of trust, model-checking, and action logics. ALFEBIITE was also a highly inter-disciplinary project, with partners from computer science, philosophy, cognitive science and law. In this paper, we wish to reflect on how the interaction between the computer scientists and the computational lawyers on the idea of the 'open agent society' inspired a programme of research whose trajectory has carried it through logic-based virtual organisations, dynamic norm-governed systems, self-organising electronic institutions, complex event recognition, computational justice, collective awareness and design contractualism. The objective of this paper is, by tracing the course of this trajectory, to expose a number of future challenges, research issues and other legal/ethical issues for the COIN community to consider.

Keywords: Multi-agent systems, electronic institutions, norms, self-organising systems, complex event recognition, computers and law.

1 Introduction

It is now more than 10 years since the EU FET project ALFEBIITE[1] finished: the terms 'infohabitants' and 'universal information ecosystem' never caught on; it's logical framework for ethical behaviour was never fully realised; it's domain name was porn-napped; and it's end-of-project collected volume, though often referenced, was never actually published.

On the other hand, some of its researchers made pioneering contributions to (inter alia) formal models of trust [10], model-checking [16], and logics for action and agency [39]. It was also the starting point for the multi-agent animation and simulation platform that became PreSage-2 [17], a formal model of forgiveness [43], and a methodology for the design of socio-technical systems with socially intelligent, and socially-aware, technical components [12].

[1] Pronounced $\alpha\beta$: the acronym stood for "A Logical Framework for Ethical Behaviour between Infohabitants in the Information Trading Economy of the Universal Information Ecosystem". No-one ever asked twice.

Furthermore, ALFEBIITE was also a highly inter-disciplinary project, with partners from computer science, philosophy, cognitive science and law. In this paper, we wish to reflect on how the interaction between the computer scientists and the computational lawyers on the idea of the 'open agent society' [27] inspired a programme of research whose trajectory has carried it through logic-based virtual organisations, dynamic norm-governed systems, self-organising electronic institutions, complex event recognition, computational justice, collective awareness and design contractualism.

The objective of this paper is, by tracing the course of this trajectory, to expose a number of future challenges, research issues and other legal/ethical issues for the COIN community to consider. Therefore, this paper is structured accordingly. Section 2 presents more detail on the background to the ALFEBIITE project and one element of its output, an executable framework for a kind of computational agent society. Given this starting point, the next three sections trace the path from the 'open agent society' through the computational framework of dynamic norm-governed multi-agent systems (Section 3) onto self-organising electronic institutions (Section 4) and logic-based complex event recognition (Section 5). Section 6 presents some research challenges for socio-technical systems, while Section 7 some further legal and ethical issues. We conclude with some remarks on the evaluation of speculative research.

2 Background

In their paper on *Institutionalised Power* [11], Jones and Sergot gave the first formal characterisation of the notion of *counts as*, the cornerstone of speech act theory [37], according to which *X counts-as Y in context C"*. The term institutionalised power refers to that characteristic feature of institutions, whereby designated agents, often acting in specific roles, are empowered to create or modify facts of special significance in that institution (*institutional facts*), through the performance of a designated action, e.g. often a speech act.

In the same year that the Jones and Sergot paper was published, 1996, the Foundation for Intelligent Physical Agents held it's inaugural meeting. Ostensibly intended to address the issue of interoperability in distributed systems with 'intelligent' agents (it's unstated purpose was to de-risk a potentially disruptive technology), one of its key technical specifications was on agent communication. This specified an ACL (agent communication language) semantics based on an internal 'mentalistic' approach. In so doing, the specification overlooked Searle's contention that speaking a language was to engage in a rule-governed form of behaviour (like playing a game), and consequently the ACL 'calculus' omitted the constitutive aspect of conventional communication, in particular 'counts as'.

This meant that, for example, in the FIPA standard for a contract-net protocol, it was not clear which action established the contract (and this matters; in different legal contexts, when the contract is recognised in law can depend on when it is signed, when it is posted, when it is delivered it, or when it is delivered). To address this misrepresentation, one of the primary objectives of the

ALFEBIITE project was to establish the logical and computational foundations of socially-organised, norm-governed, distributed multi-agent systems: what we came to call the 'open agent society' [27]

In the course of the project, an executable specification was developed for a sub-class of computational societies that exhibited the following characteristics:

- It adopts the perspective of an external observer and views societies as instances of normative systems, that is, it describes the permissions and obligations of the members of the societies, considering the possibility that the behaviour of the members may deviate from the ideal.
- It explicitly represents the institutionalised powers [25] of the member agents, a standard feature of any norm-governed interaction. Moreover, it maintains the long established distinction (in the study of social and legal systems) between physical capability, institutionalised power and permission.
- It provides a declarative formalisation of the aforementioned concepts by means of two temporal action languages with clear routes to implementation: in particular (but not only) the Event Calculus [14]. The specification could be executed and validated, both at design-time and run-time, by automated systems, including the agents themselves.

This sub-class of computational society, as a computational instantiation of 'the open agent society', was, over the next few years, the basis for exploring various ideas in virtual organisations and for piecemeal formulation of various different protocols (e.g. for voting, dispute resolution, argumentation, e-commerce, etc.), until it coalesced in the concept of dynamic norm-governed multi-agent systems.

3 Dynamic Norm-Governed Multi-Agent Systems

The first attempt to codify more fully the abstract concept of "the open agent society" in computational form, taking into account the requirements identified by the ALFEBIITE project (including norms, protocols, and adaptation), was the framework of dynamic norm-governed multi-agent systems [2].

This framework was designed to support self-modification of the rules or protocols of a norm-governed system, by the agents themselves, at runtime. The framework defined three components: a specification of a norm-governed system, a protocol-stack for defining how to change the specification, and a topological space for expressing the 'distance' between one specification instance and another.

The study of legal, social and organisational systems has often been formalised in terms of norm-governed systems. The framework maintains the standard and long established distinction between physical capability, institutionalised power, and permission (see e.g. [11] for illustrations of this distinction). Accordingly, a specification of a norm-governed system expresses five aspects of social constraint: the physical capabilities; the institutionalised powers; the permissions, prohibitions and obligations of the agents; the sanctions and enforcement policies that deal with the performance of prohibited actions and non-compliance with obligations; and the designated roles of empowered agents.

Underpinning this specification is a communication language. This language is used to define a set of protocols for conducting the business of the institution. In the framework, the protocol *stack* is used by the agents to modify the rules or protocols of a norm-governed system at runtime. This stack defines a set of object level protocols, and assumes that during the execution of an object protocol the participants could start a meta-protocol to (try to) modify the object-level protocol. The participants of the meta-protocol could initiate a meta-meta protocol to modify the rules of the meta-protocol, and so on. In addition to object- and meta-protocols, there are also 'transition' protocols. These protocols define the conditions in which an agent may initiate a meta-protocol, who occupies which role in the meta-protocol, and what elements (the *degrees of freedom*: DoF) of an object protocol can be modified as a result of the meta-protocol execution.

Each type of method is a DoF, and with two values for each method, this gives four possible *specification instances*.

Each DoF could take one out of possible range of values: a specification where each DoF had a specific value was called a specification instance. This was the basis for defining a *specification space* as a 2-tuple, where one component is the set of all possible specification instances and the other component is a function d which defines a 'distance' between any pair of elements in the set. Note that we can imagine more access control and exclusion mechanisms, so more specification instances, and so a larger specification space.

The framework for dynamic norm-governed multi-agent systems was the stepping stone to two further concepts: self-organising electronic institutions (Section 4) and logic-based complex event recognition (Section 5).

4 Self-Governing Institutions

Electronic institutions are used to represent the structures, functions and processes of an institution in mathematical, logical or computational form, cf. [42].

In terms of functional representation, an institution's rules can be divided into three levels, from lower to higher [23]: *operational-choice rules* (OC) are concerned with the provision and appropriation of resources, as well as with membership, monitoring and enforcement; *social collective-choice rules* (SC) drive policy making and selection of operational-choice rules; and *constitutional-choice rules* (CC) deal with eligibility and formulation of the collective-choice rules.

In terms of structural representation, a formal hierarchy is straightforward to identify, and this needs to be associated with a specification of the remit of each unit of the hierarchy, for example, in terms of which social-, collective or operational-rules, and which DoF, the unit is allowed to apply and modify.

A formal representation of institutional processes can be given, which identifies their procedural, temporal and normative aspects (as stated, typically of concern in the study of social and organisational systems) and this can be done using, for example, the Event Calculus, a logic programming formalism for representing and reasoning about events and their effects [14].

One of the most frequently encountered problems in open systems is the requirement to distribute resources which have been pooled in some ways – this is of course a collective action problem. One solution to the problem is the theory of Elinor Ostrom [23], who was awarded the Nobel Prize for Economic Science in 2009 for her work on self-governing institutions for common-pool resource management, and how there were eight common features of such institutions which led to stable and sustainable solutions, and one or more were missing from those institutions which failed to do so. Ostrom then further recommended that instead of trying to evolve such institutions, it would be better to *design* them, and codified the fears instead as eight institutional design principles

By applying the sociologically-inspired computing methodology [12] to Ostrom's institutional design principles, using the computational framework of dynamic norm-governed systems as the target 'calculus' for the formal characterisation, two significant results were achieved:

- Showing that Elinor Ostrom's institutional design principles for enduring self-governing institutions [23], which embody many principles of natural and retributive justice, can be axiomatised in computational logic and then used for specifying and implementing self-organising electronic institutions with corresponding properties of endurance and sustained membership [31];
- Showing that Nicholas Rescher's theory of distributive justice [35] based on the canon of legitimate claims can also be axiomatised in computational logic and as complement to Ostrom's principles, used to ensure fairness in resource distribution over time (according to a chosen fairness measure, the Gini index) [26].

5 Complex Event Recognition & Run-Time Event Calculus

The Event Calculus has been frequently used, as in the previous section, for specifying and reasoning about (open) multi-agent systems (MAS) due to its simplicity and flexibility. However, the EC also has a number of well-known limitations. One of these is an issue of *scale*; that is, as the number of agents increases, and/or the number of events in the narrative increases, then the performance and efficiency deteriorate unacceptably. One approach to overcoming such limitations is to use caching [5].

Building on some of these ideas for efficiency improvement, in order deal with very large agent societies, we have developed the 'Event Calculus for Run-Time reasoning' (RTEC) [3]. RTEC includes various optimisation techniques for an important class of computational tasks, specifically those in which given a record of what events have occurred (a 'narrative') and a set of axioms (expressing the specification of a MAS), we compute the values of various facts (denoting institutional powers, permissions, and other normative relations) at specified time points. RTEC thus provides a practical means of informing the decision-making of the agents and their users, and the system designers. In what follows we briefly discuss the use of RTEC for specifying and executing very large MAS.

Table 1: Main predicates of RTEC.

Predicate	Meaning
happensAt$(E,\ T)$	Event E occurs at time T
holdsAt$(F=V,\ T)$	The value of fluent F is V at time T
holdsFor$(F=V,\ I)$	I is the list of the maximal intervals for which $F=V$ holds continuously
initiatedAt$(F=V,\ T)$	At time T a period of time for which $F=V$ is initiated
terminatedAt$(F=V,\ T)$	At time T a period of time for which $F=V$ is terminated
union_all$(L,\ I)$	I is the list of maximal intervals produced by the union of the lists of maximal intervals of list L
intersect_all$(L,\ I)$	I is the list of maximal intervals produced by the intersection of the lists of maximal intervals of list L
relative_complement_all$(I',\ L,\ I)$	I is the list of maximal intervals produced by the relative complement of the list of maximal intervals I' with respect to every list of maximal intervals of list L

An *event description* in RTEC includes rules that define the event instances with the use of the happensAt predicate, the effects of events with the use of the initiatedAt and terminatedAt predicates, and the values of the fluents with the use of the holdsAt and holdsFor predicates, as well as other, possibly atemporal, constraints. Table 1 summarises the RTEC predicates available to the system specification developer.

We represent the actions of the agents and the environment by means of happensAt, while the state of the agents and the environment are represented as fluents. In MAS execution, therefore, the task is to compute the maximal intervals for which a fluent representing an agent or environment variable (such as the institutional powers of an agent) has a particular value continuously.

RTEC includes additional optimisation techniques that allow for very efficient and scalable MAS execution. A form of caching stores the results of subcomputations in the computer memory to avoid unnecessary recomputations. A simple indexing mechanism makes RTEC robust to events that are irrelevant to the computations we want to perform and so RTEC can operate without data filtering modules. The set of interval manipulation constructs mentioned in the previous section simplify MAS specifications and improve reasoning efficiency. Finally, the 'windowing' mechanism supports real-time MAS execution. One main motivation for RTEC is that it should remain efficient and scalable in applications where events arrive with a (variable) delay: RTEC can update the already computed MAS state (including for example the institutional pow-

ers, permissions and obligations of the agents), when events arrive with a delay. The code of RTEC is available at <http://users.iit.demokritos.gr/~a.artikis/EC.html>.

The anticipated scale of future multi-agent systems requires the capability to process many thousands of events per second. This is beyond the computational practicality of simple versions of the Event Calculus, a new dialect, such as RTEC, is required in order to support computational tasks such as scalable, pro-active, event-driven decision-making and complex event recognition.

6 Research Challenges

In this section, we outline a quartet of research challenges for which the concepts of self-organising electronic institutions and logic-based complex event recognition are critical for socio-techncal systems. These challenges are:

- **computational justice**, as the study of some form of 'correctness' in the outcomes from qualitative algorithmic deliberation and decision-making;
- (interoceptive) **collective awareness**, as an attribute (internal sense of well-being) of communities that helps solve collective action problems;
- (electronic) **social capital**, as a complexity-reducing short-cut in cooperation dilemmas that have multiple equilibria or are computationally intractable (e.g. n-player games);
- **polycentric self-governance**, reconciling potentially conflicting interests by giving consideration to "common purposes" within multiple centres of decision-making.

6.1 Computational Justice

Computational justice [30] is an interdisciplinary investigation at the interface of computer science and philosophy, economics, psychology and jurisprudence, enabling and promoting an exchange of ideas and results in both directions. One of its main goals is to introduce notions borrowed from Social Sciences, such as fairness, equity, transparency and openness into computational settings. It is also concerned with exporting the developed mechanisms back to Social Sciences, both to better understand their role in social settings, as well as to leverage the knowledge gained in computational settings and transport it to the social one.

Although 'justice' is a concept open to many definitions, the study of computational justice focuses on the following 'qualifiers', which have been used in the social sciences:

- *Natural* or *social* justice is concerned with the right of inclusion and participation in the decision making processes affecting oneself.
- *Distributive* justice deals with fairly allocating resources (e.g. goods, tasks, benefits) amongst a set of agents.
- *Retributive* justice addresses the issue of sanctioning non-compliant behaviour.

- *Procedural* justice considers evaluating whether procedures are 'fit-for-purpose' (e.g. for achieving inclusivity, transparency, and balance (in terms of trading-off cost-effectiveness vs. efficiency, for example)).
- *Interactional* justice is concerned with the subjective view of the agents and whether they feel they are being treated fairly by the decision makers.

All these qualifiers are important in socio-cognitive technical systems, since they address different issues that require 'conditioning' in such situations. For example, when distributing collectivised resources, it might be tempting to go for an optimal allocation in terms of maximising some overall utility. While such an utilitarian view might be the most appropriate in some cases, in socio-cognitive technical systems (or any other system where participants can somehow evaluate their satisfaction and act depending on it) it might be better to seek allocations that may be sub-optimal but that take into account the notion of fairness, thereby increasing sustainability in the longer-term at the expense of optimality in the short-term.

Similarly, if a participant violates some norm or rule, it could be subject to some kind of punishment, penalty or sanction. This can be seen either as a direct consequence of the wrong-doing (retributivism view) or as a deterrent for future wrong-doing by the agent being punished as well as by the agents observing the punishment (utilitarianism view). However, the punishment should be proportional to the offence including both the extent of the violation, as well as recidivism (i.e. repeated offences by the same agent). The punishing entity can either be some authorised body (e.g. judicial system, elected board) or the participants themselves if they have the capability of punishing each other. However, note the role of forgiveness (as discussed earlier) is an important element of retributive justice.

Natural or social justice is concerned with issues such as the inclusivity of participants in decision-making, for instance to decide how resources are allocated (e.g. choose the allocation mechanism), or how a decision should be made (e.g. which voting protocols are to be used). Inclusive participation of this kind would provide the system with the feature of self-organisation, in the sense of self-governing the resources, and ensure that those who are subject to.

Procedural justice is required to provide governance mechanisms which are 'fit for purpose' [29], i.e. addressing the following sorts of question. Are the rights of members of the institution to participate in collective-choice arrangement adequately represented and protected? Is an institution where decisions are made by one actor who justifies its decision 'preferable' to an institution where the decision is made by a committee that does not offer such justification? Is an institution which expends significant resources on determining the most equitable distribution 'preferable' to one that uses a cheaper method to produce a less fair outcome, but has more resources to distribute as a consequence?

Interactional justice allows the participants to make a subjective assessment of whether they are being treated fairly, that the decision makers should provide enough information to them (e.g. justification of allocation). For instance, in a case of scarcity of resources, if the participants were informed about this scarcity,

they would probably better understand not being allocated any resources. Contrarily, if they are not informed about this issue, they might think that they are not being allocated resources because the decision makers are biased towards other participants.

6.2 Collective Awareness

The development of collective awareness has been advocated as enhancing the choice of sustainable strategies by the members of a community and therefore ensuring the successful adaptation process [40]. In communities in which collective awareness is barely present individuals may experience a diminished appreciation of the global situation and present constrained flexibility in adjustment to change because they do not share the same comprehension of situation with others. They are also less willing to obey the norms and rules set by the community because they do not feel themselves as members of community and are not aware of others seeing them as ones. They understand the situation they are in from a micro-level perception and might additionally recognize the macro-level description of the situation, however, they might not be aware of interactions occurring at the meso level. As a result, individuals make decisions that are sub-optimal from the perspective of the whole system making it less fair, more inefficient and so vulnerable to collapse through instability. Therefore, collective awareness is critical to the formation of socio-cognitive technical systems.

It has been argued that collective awareness occurs "when two or more people are aware of the same context and each is aware that the others are aware of it" [13]. This awareness of others awareness has been indicated as a critical element of collaboration within the communities, especially virtual ones such as computer-mediated communities [6]. In socio-cognitive technical systems, an alternative approach might be sought by moving away from mutual knowledge and taking a multi-modal approach [10], and define collective awareness of some proposition ϕ as a two part relation: firstly a *belief* that there is a group, and secondly an *expectation* that if someone is a member of that group, then they believe the proposition ϕ.

Collective awareness is "an attribute of communities that helps them solve collective action problem", i.e. analogous to the way that social capital is defined by Ostrom and Ahn [21] as "an attribute of individuals that helps them solve collective action problems". Without this community attribute, individuals may take actions that are sub- optimal from a community-wide perspective, leading to diminished utility and sustainability. Individuals may understand the situation they are in from a micro-level perspective (e.g. in a power system, reducing individual energy consumption) and might additionally recognise the macro-level requirement (e.g. meeting national carbon dioxide emission pledges); however, they might not be aware of interactions occurring at the meso-level which are critical for mapping one to the other.

Therefore, collective awareness has a critical role in the formation of electronic institutions, the regulation of behaviour within the context of an institution, and the direction (or selection) of actions intended to achieve a com-

mon purpose. If we consider collective awareness as being different from mutual knowledge, and base it on expectations for resolving collective action problems instead, then from the human-participant perspective we identify certain requirements as necessary conditions for achieving collective awareness as a precursor to collective action in socio-cognitive technical systems. These requirements are:

- Interface cues for collective action, i.e. that participants are engaged in a collective action situation;
- Visualisation: appropriate presentation and representation of data, making what is conceptually significant perceptually prominent;
- Social networking: fast, convenient and cheap communication channels to support the propagation of data;
- Feedback: individuals need to know that their ('small', individual) action X contributed to some ('large', collective) action Y which achieved beneficial outcome Z;
- Incentives: typically in the form of social capital [21], itself identified as an attribute of individuals that helps to solve collective action problems.

However, preliminary experiments have shown that people have insufficient attention to be sufficiently pro-active in monitoring and responding to all the changes in their environment. Complex event recognition is clearly going to be a critical technology in being able to detect situations that genuinely require user attention, from which the appropriate interface cues and visualisation methods can be drawn.

6.3 Electronic Social Capital

In social systems, it has been observed that *social capital* is an attribute of individuals that enhances their ability to solve collective action problems [21]. Social capital takes different forms, for example trustworthiness, social networks and institutions. Each of these forms is a subjective indicator of one individual's expectations of how another individual will behave in a strategic game: i.e., if the individual has a high reputation (trustworthiness) for honouring agreements and commitments, or it is known personally to be reliable (social network), or a belief there is a set of (institutional) rules (triggering expectations that someone else's behaviour will conform to those rules, and that they will be punished if they do not) (cf. [10]).

Therefore, we propose that an electronic form of social capital could be used as an attribute of participants in a socio-cognitive technical system, to enhance their ability to solve collective action problems, by reducing the costs and complexity in joint decision-making in repeated pairwise interactions [25]. Two key features of this framework are firstly, the use of institutions as one social capital attribute, and secondly, the use of the Event Calculus to process events which update all the social capital attributes.

We note, en passant, that as a further direction of research, the role of electronic social capital and its relation to cryptocurrencies such as Bitcoin and Venn, for example in the creation of incentives and alternative market arrangements, needs to be fully explored.

6.4 Polycentric (Self-)Governance

It is well-known that managing critical infrastructure, like a national energy generation, transmission and distribution network, will necessarily involve multiple agencies with differing (possibly competing or even conflicting) interests, effectively creating an "overlay" network of relational dynamics which also needs to be resolved. Furthermore, there is some, not always well-understood, interconnection of public and private ownership that makes the overall system both stable and sustainable.

Therefore, in analysing any such complex system, it is critical to identify the agencies and determine its institutional *common purpose*, what the agency (through its institution) is trying to achieve or maintain, by coordination with other institutions and by the decision-making of its members. Such analysis makes it possible to understand the 'ecosystem' of institutions and how they fit together as collaborators or competitors, based on the nature of their purposes and the scope of their influence.

It is then an open question if the ecosystem of institutions can be represented using holonic systems architectures, which are capable of achieving large-scale multi-criteria optimisation [7]. The key concept here is the idea of *holonic institutions*, whereby each institution is represented as a holon, which can be aggregated or decomposed into supra- and sub-holons respectively.

The outcome of a positive answer to this question would be twofold. Firstly, that would support polycentric self-governance at all scales of the system (i.e. multiple centres of autonomous decision-making), and in particular would support *subsidiarity* (the idea that problems are solved as close to the local source as possible). Secondly, it would encourage the institutions to recognise their role in the "scheme of things" in relation to institutions at the same, higher and lower levels. This is an essential requirement for *adaptive* institutions [33] and this establishment of "systems thinking" as a commonplace practice.

7 Some Legal and Ethical Issues

Finally, we note that the idea of collective intelligence, including both humans and software agents, and agents as an enabling technology for added-value services, was a key part of the original vision: the open agent society for the user-friendly digital society. In this section, we consider some legal and ethical issues that arise when the concept of the open agent society, with its computational instantiations and associated technologies, converges with other technological developments like ubiquitous computing, implant devices and sensor networks.

7.1 Big Data and Knowledge Commons

[9] was concerned with treating knowledge as a shared resource, motivated by the increase in open access science journals, digital libraries, and mass-participation user-generated content management platforms. It then addressed the question

of whether it was possible to manage and sustain a knowledge commons, using the same principles used to manage ecological systems with natural resources. A significant challenge in the democratisation of Big Data is the extent to which formal representations of intellectual property rights, access rights, copy-rights, etc. of different stakeholders can be represented in a system of computational justice and encoded in Ostrom's principles for knowledge commons. As observed in [41], the power of Big Data and associated tools for analytical modelling: "... should not remain the preserve of restricted government, scientific or corporate élites, but be opened up for societal engagement and critique. To democratise such assets as a public good, requires a sustainable ecosystem enabling different kinds of stakeholder in society".

It could be argued that Ostrom's institutional design principles reflect a pre-World Wide Web era of scholarship and content creation, and despite their original insightful work [22], these developments make it difficult to apply the principles to non-physical shared sources such as data or knowledge commons, and a further extension of the theory is required to develop applications based on participatory sensing for the information-sharing economy [18].

7.2 Privacy and Ubersurveillance

The issue here is whether intelligent agents will exacerbate a perceived trend from pervasive computing to persuasive computing and ultimately to coercive computing. The threat is sufficient for the central premise of Dave Eggers' science fiction novel *The Circle* (McSweeney's, 2013), that opting out from providing personal data is tantamount to theft, to appear plausible.

It is in this context that programmes for Privacy by Design [4], imposing limits on uberveillance [20] and the ethical issues of wearable, bearable and implant technology in the context of Big Data and the Internet of Things (IoT) [19, 24] are set. Designers should heed "the principles incorporated in the European Union and international treaties as well as the laws of EU member states: the precautionary principle, purpose specification principle, data minimization principle, proportionality principle, and the principle of integrity and inviolability of the body, and dignity" [24].

7.3 Design Contractualism

This requirement is based on the observation that affective and pervasive applications are implemented in terms of a sense/respond cycle, called the affective loop [8] or the biocybernetic loop [38]. The actual responses are determined by decision-making algorithms, which should in turn be grounded within the framework of a mutual agreement, or a social contract [32]. This contract should specify how individuals, government, and commercial organizations should interact in a digital, and digitized, world.

In the context of affective computing, this contractual obligation has been called design contractualism by Reynolds and Picard [36]. Under this principle, the designer makes moral or ethical judgements, and encodes them in the system.

In fact, there are already several prototypical examples of this, from the copyleft approach to using and modifying intellectual property, to the IEEE Code of Ethics and the ACM Code of Conduct, to and TrUSTe self-certifying privacy seal. In some sense, these examples are a reflection of ideas of Lessig that Code is Law [15], or rather in this case, Code is Moral Judgement.

Returning to the security implications of user-generated content and Big Data (see above), design contractualism [28] also underpins the idea of using implicitly-generated data as input streams for Big Data, and treating that data as a knowledge commons [9, 18]. Using the principles of self- governing institutions for managing common pool resources identified by Ostrom [23], we advocate managing Big Data from the perspective of a knowledge commons. Design contractualism, from this perspective, effectively defines an analytical framework for collecting and processing user-generated con-tent input to Big Data as a shared resource with both normative and social dimensions. The normative dimension is the existence of institutional rules embodying the social contract. The social dimension is the belief that there are these rules and that others behaviour will conform to these rules, as a trust shortcut [10].

7.4 The New Economic Model

Reich [34] suggests that we are moving towards a economy in which all routine, predictable work is automated or performed by robots, and the profits go to the owners of the robots; and all less predictable work is performed by human beings, and all the profits got to the owners of the middleware (i.e. in a peer-to-peer economy, while the intelligence is at the edge, the value is in the network. This model affects not just taxi drivers, plumbers and hotel owners, but will increasingly effect professions such as law, medicine and academia, given the rise of online legal services (such as dispute resolution), health provision, and MOOCs (massive open online courses), and (in the UK) the rise of zero-hour contracts to service those courses.

7.5 Summary

This discussion of some legal and ethical issues is necessarily limited in this context, and the interaction of technology with society also has 'political', 'cultural' and even 'generational' dimensions that need to be addressed. For example, it could be argued that conflicts of objective interests cannot necessarily be resolved by collective awareness, collective intelligence and collective action. In addressing climate change, there is a complex interaction between (at least) loss aversion bias (people's preference for avoiding losses over acquiring gains, especially when the loss is incurred by themselves and the gains are accrued by others) and political 'framing'.

Similarly, the symbiotic partnership of people and technology might not be as benign as might be imagined. The idea of 'designing' institutions for 'fairer' societies arguably presents serious ideological, political (and even moral [1]) problems. The issue of Artificial Intelligence dominating society has been a staple

premise of science fiction, and attracts the attention of leading scientists from other fields.[2] However, with the ever increasing power of data mining and both predictive and prescriptive analytics, it is arguably the case that we should still be more wary of the programmers than the programs.

8 Summary and Conclusions

In summary, we have traced the development of the concept of the 'open agent society' from its origins and early formalisation in ALFEBIITE to its reification as dynamic norm-governed systems, and its metamorphosis into electronic institutions for fair and sustainable resource management and run-time cacluli for logic-based complex event recognition.

In this paper, we have necessarily focussed on the institutional aspects that resulted from the collaboration between the computer scientists and lawyers and philosophers, in particular Jon Bing and Andrew Jones. A separate paper could be written on the outcomes of the collaboration between the computer scientists and the cognitive scientists, in particular Cristiano Castelfranchi, whose original work inspired formal models of trust and economic reasoning, emotions, forgiveness and anticipation.

It was strictly *not* the objective of this paper to catalogue the authors' own contributions, the objectives of this paper were:

- to highlight and emphasise the importance of inter-disciplinary research, and acknowledge the beneficial collaborations: it is likely that none of this would have been possible without the insights and understanding that stemmed from the fields of law and legal information systems;
- to consider the research programme looking back but in particulate looking forwards, and trying to raise a number of research challenges and legal/ethical/political issues considered to be relevant to COIN: COIN research is increasingly having transformative social impact and the precautionary principle needs to be taken into account; and
- to demonstrate the difficulty of evaluating the impact of basic research.

On this last point, the current trend of demanding that basic research projects should specify 'pathways to impact', or that research papers should be evaluated according to their 'impact statements' within some fixed and arbitrary period, should be treated with some caution and no little skepticism. The fact remains that the pathway to impact is probably even less predictable or manageable than the research programme itself; and as for the 'impact statement', this is so time-dependent as to require continual re-assessment, not just at a fixed point.

Acknowledgements

We would particularly like to thank the anonymous reviewers for their very helpful and encouraging comments.

[2] See, for example, http://www.bbc.co.uk/news/technology-30290540 and Stephen Hawking's recent interventions on Artificial Intelligence.

References

1. Allen, C., Wallach, W., Smit, I.: Why machine ethics. IEEE Intelligent Systems 21(4), 12–17 (2006)
2. Artikis, A.: Dynamic specification of open agent systems. Journal of Logic and Computation 22(6), 1301–1334 (2012)
3. Artikis., A., Sergot, M., Paliouras, G.: An event calculus for event recognition. IEEE Transactions on Knowledge and Data Engineering (2015)
4. Cavoukian, A.: Privacy by design [leading edge]. IEEE Technol. Soc. Mag. 31(4), 18–19 (2012), http://dx.doi.org/10.1109/MTS.2012.2225459
5. Chittaro, L., Montanari, A.: Efficient temporal reasoning in the cached event calculus. Computational Intelligence 12(3), 359–382 (1996)
6. Daassi, M., Favier, M.: Developing a measure of collective awareness in virtual teams. International Journal of Business Information Systems 2(4), 413–425 (2007)
7. Frey, S., Diaconescu, A., Menga, D., Demeure, I.M.: A holonic control architecture for a heterogeneous multi-objective smart micro-grid. In: 7th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2013, Philadelphia, PA, USA, September 9-13, 2013. pp. 21–30 (2013)
8. Goulev, P., Stead, L., Mamdani, A., Evans, C.: Computer aided emotional fashion. Computers & Graphics 28(5), 657–666 (2004)
9. Hess, C., Ostrom, E.: Understanding Knowledge as a Commons. MIT Press (2006)
10. Jones, A.: On the concept of trust. Decision Support Systems 33(3), 225–232 (2002)
11. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Journal of the IGPL 4(3), 427–443 (1996)
12. Jones, A.J., Artikis, A., Pitt, J.: The design of intelligent socio-technical systems. Artificial Intelligence Review 39(1), 5–20 (2013)
13. Kellogg, W., Erickson, T.: Social translucence, collective awareness, and the emergence of place. a position paper for the role of place in shaping virtual community. In: Proceeding CSCW (2002)
14. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing 4, 67–95 (1986)
15. Lessig, L.: Code: And Other Laws of Cyberspace, Version 2.0. Basic Books (2006)
16. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: A model checker for the verification of multi-agent systems. In: Bouajjani, A., Maler, O. (eds.) Computer Aided Verification. Lecture Notes in Computer Science, vol. 5643, pp. 682–688. Springer (2009)
17. Macbeth, S., Busquets, D., Pitt, J.: Principled operationalization of social systems using presage2. In: Gianni, D., D'Ambrogio, A., Tolk, A. (eds.) Modeling and Simulation-Based Systems Engineering Handbook. CRC Press (2014)
18. Macbeth, S., Pitt, J.: Self-organising management of user-generated data and knowledge. The Knowledge Engineering Review FirstView(1-28) (2014)
19. Michael, K., Michael, M.: Implementing 'namebers' using microchip implants: The black box beneath the skin. In: Pitt, J. (ed.) This Pervasive Day, chap. 10. IC Press (2012)
20. Michael, K., Michael, M., Perakslis, C.: Be vigilant: There are limits to veillance. In: Pitt, J. (ed.) The Computer After Me, chap. 13. IC Press (2014)
21. Ostrom, E., Ahn, T.: Foundations of Social Capital. An Elgar Reference Collection, Edward Elgar Pub. (2003)
22. Ostrom, E., Hess, C.: A framework for analyzing the knowledge commons. In: Hess, C., Ostrom, E. (eds.) Understanding Knowledge as a Commons: From Theory to Practice, pp. 41–82. MIT Press, Cambridge, MA (2006)

23. Ostrom, E.: Governing the commons: The evolution of institutions for collective action. Cambridge, UK: Cambridge University Press (1990)
24. Perakslis, C., Pitt, J., Michael, K., Michael, M.: Pervasive technologies: Principles to consider. International Journal of Medical Implants and Devices (2015)
25. Petruzzi, P.E., Busquets, D., Pitt, J.V.: Experiments with social capital in multi-agent systems. In: PRIMA 2014: Principles and Practice of Multi-Agent Systems. pp. 18–33 (2014)
26. Pitt, J., Busquets, D., Macbeth, S.: Distributive justice for self-organised common-pool resource management. ACM Trans. Auton. Adapt. Syst. 9(3), 14 (2014)
27. Pitt, J.: The open agent society as a platform for the user-friendly information society. AI Soc. 19(2), 123–158 (2005)
28. Pitt, J.: Design contractualism for pervasive/affective computing. IEEE Technol. Soc. Mag. 31(4), 22–29 (2012), http://dx.doi.org/10.1109/MTS.2012.2225458
29. Pitt, J., Busquets, D., Riveret, R.: Procedural justice and 'fitness for purpose' of self-organising electronic institutions. In: PRIMA 2013: Principles and Practice of Multi-Agent Systems, Lecture Notes in Computer Science, vol. 8291, pp. 260–275. Springer Berlin Heidelberg (2013)
30. Pitt, J., Busquets, D., Riveret, R.: The pursuit of computational justice in open systems. AI & SOCIETY pp. 1–20 (2013)
31. Pitt, J., Schaumeier, J., Artikis, A.: Axiomatisation of socio-economic principles for self-organising institutions: Concepts, experiments and challenges. ACM Trans. Auton. Adapt. Syst. 7(4), 39:1–39:39 (Dec 2012)
32. Rawls, J.: A Theory of Justice. Harvard, MA: Harvard University Press (1971)
33. RCEP: 28th report: Adapting institutions to climate change. Royal Commission on Environmental Protection, The Stationery Office Limited, UK (2010)
34. Reich, R.: The sharing economy is hurtling us backwards. Salon (February 2015)
35. Rescher, N.: Distributive Justice. Bobbs-Merrill (1966)
36. Reynolds, C., Picard, R.: Affective sensors, privacy, and ethical contracts. In: Proceedings CHI 2004 extended abstracts on Human factors in computing systems. pp. 1103–1106 (2004)
37. Searle, J.: Speech Acts. CUP (1969)
38. Serbedzija, N.: Reflective computing – naturally artificial. In: Pitt, J. (ed.) This Pervasive Day, chap. 5. IC Press (2012)
39. 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 VIII, Lecture Notes in Computer Science, vol. 4995, pp. 1–54. Springer (2008)
40. Sestini, F.: Collective awareness platforms: engines for sustainability and ethics. IEEE Technology and Society Magazine 31(4), 54–62 (2012)
41. Shum, S.B., Aberer, K., Schmidt, A., Bishop, S., Lukowicz, P., Anderson, S., Charalabidis, Y., Domingue, J., de Freitas, S., Dunwell, I., Edmonds, B., Grey, F., Haklay, M., Jelasity, M., Karpistenko, A., Kohlhammer, J., Lewis, J., Pitt, J., Sumner, R., Helbing, D.: Towards a global participatory platform: Democratising open data, complexity science and collective intelligence. European Physical Journal: Special Topics 214 (2012)
42. Sierra, C., Rodriguez-Aguilar, J., Noriega, P., Esteva, M., Arcos, J.: Engineering multi-agent systems as electronic institutions. European Journal for the Informatics Professional 4(4), 33–39 (2004)
43. Vasalou, A., Hopfensitz, A., Pitt, J.: In praise of forgiveness: Ways for repairing trust breakdowns in one-off online interactions. Int. J. Hum.-Comput. Stud. 66(6), 466–480 (2008)

Agent Teams for Design Problems

Leandro Soriano Marcolino[1], Haifeng Xu[1], David Gerber[3], Boian Kolev[2], Samori Price[2], Evangelos Pantazis[3], and Milind Tambe[1]

[1] Computer Science Department, University of Southern California, CA, USA
[2] Computer Science Department, California State University, Dominguez Hills, USA
[3] School of Architecture, University of Southern California, CA, USA
Department of Civil Engineering, University of Southern California, CA, USA
{sorianom,haifengx,dgerber}@usc.edu,
{bkolev1,sprice25}@toromail.csudh.edu, {epantazi,tambe}@usc.edu

Abstract. Design imposes a novel social choice problem: using a team of voting agents, maximize the number of optimal solutions; allowing a user to then take an aesthetical choice. In an open system of design agents, team formation is fundamental. We present the first model of agent teams for design. For maximum applicability, we envision agents that are queried for a single opinion, and multiple solutions are obtained by multiple iterations. We show that diverse teams composed of agents with different preferences maximize the number of optimal solutions, while uniform teams composed of multiple copies of the best agent are in general suboptimal. Our experiments study the model in bounded time; and we also study a real system, where agents vote to design buildings.

Keywords: Collaboration, Distributed AI, Team Formation

1 Introduction

Teams of voting agents are a power tool for finding the optimal solution in many applications [15, 1, 16, 18, 10], as there are theoretical guarantees in finding one optimal choice [5]. For design problems, however, finding one optimal solution is not enough, and we actually want to find as many optimal solutions as possible, allowing a human to choose according to her aesthetical taste. Even if a user does not want to consider too many solutions, they can be filtered and clustered [7], allowing her to easily make an aesthetical choice. Hence, a system of voting agents that produces a unique optimal solution is insufficient; and we propose the novel social choice problem of maximizing the number of optimal alternatives found by a voting system. As ranked voting may suffer from noisy rankings when using existing agents [11], we study multiple voting iterations.

Traditionally, social choice studies the optimality of voting rules, assuming certain noise models for the agents, and rankings composed of a linear order over alternatives [5, 4, 14]. Hence, there is a single optimal choice, and a system is successful if it can return that optimal choice with high probability. More recently, several works have been considering cases where there is a partial order over alternatives [22, 19], or when the agents output pairwise comparisons

instead of rankings [6]. However, these works still focus on finding an optimal alternative, or a fixed-sized set of optimal alternatives (where the size is known beforehand). Therefore, they still provide no help in finding the maximum set of optimal solutions. Moreover, they assume agents that are able to output comparisons among all actions with fairly good precision, and the use of multiple voting iterations has never been studied. When considering agents with different preferences, the field is focused on verifying if voting rules satisfy a set of axioms that are considered to be important to achieve fairness [17].

In this work we offer a completely different perspective: we show that, unless we have an idealized agent, we only maximize the number of optimal solutions if we have agents with different preferences. Motivated by the need of selecting agents from an open system, for greater applicability we only consider agents that output a single action. We present a theoretical study of which teams are desirable for design problems, and how their size may effect optimality. We show that, contrary to traditional social choice models, increasing the team size may significantly harm performance; and that a diverse team of agents with different preferences is highly desirable for achieving optimality. In doing so, we draw a novel connection between social choice and *number theory*; allowing us to show, for example, that the optimal diverse team size is constant with high probability, and a prime number of optimal actions may impose problems. We present synthetic experiments to further study our model, providing realistic insights into what happens with bounded computational time. Finally, we present experiments in a highly relevant domain: architecture design, where we show teams of agents that vote to design energy-efficient buildings. Hence, this is the first work exploring and showing the potential of voting systems in being *creative*, by actually creating new alternatives from the opinions of existing agents.

2 Related Work

As mentioned, traditional works in social choice concern finding a correct ranking in domains where there is a unique optimal decision [5, 4, 14, 20]. Recent works, however, are considering more complex domains. Xia and Conitzer (2001) [22] study the problem of finding k optimal solutions, where k is known beforehand, by aggregating rankings from each agent. However, not only do they need strong assumptions about the quality of the rankings of such agents, but they also show that calculating the MLE from the rankings is an NP-hard problem.

Procaccia *et al.* [19] study a similar perspective, where the objective is to find the top k options given rankings from each agent, where, again, k is known in advance. However, in their case, they assume there still exists one unique truly optimal choice, hidden among these top k alternatives. Elkind and Shah (2014) [6] study the case where instead of rankings, the voters output pairwise comparisons among all actions, which may not follow transitivity. However, their final objective is still to pick a single winner.

Finally, outputting a full comparison among all actions can be a burden for an agent [3]. Jiang *et al.* (2014) [11] show that actual agents can have very noisy

rankings, and therefore do not follow the assumptions of previous works in social choice. Hence, as any agent is able to output at least one action (i.e., a single vote), we study here systems where agents vote across multiple iterations.

3 Design Domains

We consider in this work domains where the objective is to find the highest number of optimal solutions. We show that design is one of such domains. One of the most common computational design approaches is to use *parametric designs* [21, 9, 7], where a human designer creates an initial design of a product using *computer-aided design* tools. However, instead of manually deciding all aspects of the product, she leaves *free parameters*, whose values can be modified to change the design. This approach is used because the number of different possibilities that a human can manually create while looking for optimality is limited, so a computer system is used to refine the design and find optimal solutions.

Design problems are in general multi-objective [12], since a product normally must be optimized across different factors. For example, a product should have a low cost, but at the same time high quality, two highly-contradictory objectives. Hence, there are a large number of optimal solutions, all tied in a *pareto frontier*. For the computational system, these optimal solutions are all equivalent. However, a human may dynamically decide to value some factor over another, and/or pick the option that most pleases her own aesthetical taste or the one of the target public/client.

Note that choosing a design according to aesthetics is an undefined problem, since there are no formal definitions to compare among different options. Hence, the best that a system can do is to provide a human with a large number of optimal solutions (according to other measurable factors), allowing her to freely decide among equally optimal solutions — but most probably with not equal aesthetical qualities.

Therefore, it is natural that in design problems we are going to have many possible solutions, and we want to find as many optimal ones as possible. In fact, there are many benefits in discovering a large number of optimal solutions:

Knowledge Does not Hurt: Having more optimal solutions to choose from is never worse than having less. For example, if a designer has enough time to analyze only x solutions, she can do so with a system that provides more than x optimal solutions by sampling the exact amount that she desires. However, she will never be able to do so with a system that provides less than x optimal solutions. Moreover, we can easily identify and eliminate solutions that are similar by applying clustering and analysis techniques [7], so that every solution that the human looks at is meaningful.

Knowledge Increases Confidence in Optimality: In general design problems, the true pareto frontier is unknown. Genetic Algorithms are widely used in order to *estimate* it. The only knowledge available for the system to evaluate the optimality is in comparison with the other solutions that are also being evaluated during the optimization process [13]. Many apparently "opti-

mal" solutions are actually discovered to be sub-optimal as we find more solutions. Hence, finding a higher number of optimal solutions decreases the risk of a designer picking a wrong choice that was initially outputted as "optimal".

Knowledge Increases Aesthetical Qualities: If a human has a larger set of optimal solutions to choose from, there is a greater likelihood that at least one of these solutions is going to be of high aesthetical quality according to her preferences, or the ones of the target public.

Knowledge Increases Diversity of Options: In general, when a system x has more optimal solutions available than a system y, it does not necessarily imply that the solutions in the system x are more similar, while the optimal solutions in y are more different/diverse. In fact, all things equal, the greater the amount of optimal solutions, the higher the likelihood that we have more diverse solutions available.

4 Agent Teams

We present our theory of agent teams for design problems. Consider a team that vote together at each possible decision point of the design of a product (for example, they may vote for the value of each parameter, in a parametric design). Hence, let $\mathbf{\Phi}$ be a set of agents ϕ, and $\mathbf{\Omega}$ a set of world states ω. Each ω has an associated set of possible actions $\mathbf{A}_\omega$. At each world state, each agent ϕ_i outputs an action a_j, an optimal action according to the agent's imperfect evaluation – which may or may not be a true optimal action. Hence, there is a probability p_j that the agent outputs a certain action a_j. The teams take the action decided by *plurality voting* (ties are broken randomly). We assume that the world states are independent, and by taking an optimal action at all world states we find an optimal solution for the entire problem.

In this paper our objective goes beyond finding one optimal solution, we want to maximize the number of optimal solutions that we can find. For greater applicability, we consider here agents that output a single action. Hence, we generate multiple solutions by re-applying the voting procedure across all world states multiple times (which are called *voting iterations* – one *iteration* goes across all world states, forming one solution). Formally, let $\mathbf{S}$ be the set of (unique) optimal solutions that we find by re-applying the voting procedure through z iterations. Our objective is to maximize $|\mathbf{S}|$. We will show that, under some conditions, we can achieve that when $z \to \infty$ (we study bounded time in Section 5).

We consider that at each world state ω there is a subset $\mathbf{Good}_\omega \subset \mathbf{A}_\omega$ of optimal actions in ω. An optimal solution is going to be composed by assigning any $a_j \in \mathbf{Good}_\omega$ in world state ω – for all world states. Conversely, we consider the complementary subset $\mathbf{Bad}_\omega \subset \mathbf{A}_\omega$, such that $\mathbf{Good}_\omega \cup \mathbf{Bad}_\omega = \mathbf{A}_\omega, \mathbf{Good}_\omega \cap \mathbf{Bad}_\omega = \emptyset$. We drop the subscripts ω when it is clear that we are referring to a certain world state.

One fundamental problem is selecting which agents should form a team. By the classical voting theories, one would expect the best teams to be uniform teams composed of multiple copies of the best agent [5, 14]. Here we show, how-

ever, that for design problems uniform teams need very strong assumptions to be optimal, and in most cases they actually converge to always outputting a single solution – an undesirable outcome. However, diverse teams are optimal as long as the team's size grows *carefully*, as we explain below in Theorem 1.

We call a team *optimal* when $|\mathbf{S}| \to \prod_\omega |\mathbf{Good}_\omega|$ as $z \to \infty$, and all optimal actions are chosen by the team with the same probability. Otherwise, even though the team still produces all optimal solutions, it would tend to repeat already generated solutions whose probability is higher. Since in practice there are time bounds, such condition is fundamental to have as many solutions as possible in limited time.

We first consider agents that are independent and identically distributed. Let p_j^{Good} be the probability of voting for $a_j \in \mathbf{Good}$, and p_k^{Bad} be the probability of voting for $a_k \in \mathbf{Bad}$. Let $n = |\mathbf{\Phi}|$ be the size of the team, and N_l be the number of agents that vote for a_l in a certain voting iteration. If $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$, $p_j^{Good} > p_k^{Bad}$, the team is going to find one optimal solution with probability 1 as $n \to \infty$, as we show in the following observation:

Observation 1 *The probability of a team outputting one optimal solution goes to* 1 *as* $n \to \infty$*, if* $p_j^{Good} > p_k^{Bad}$*,* $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$*.*

Proof. Note that as the agents are independent and identically distributed, we can model the process of pooling the opinions of n agents as a multinomial distribution with n trials (and the probability of any class k of the multinomial corresponds to the probability p_k of voting for an action a_k).

Hence, for each action a_l, the expected number of votes is given by $E[N_l] = n \times p_l$. Therefore, by the *law of large numbers*, if $p_j^{Good} > p_k^{Bad}$ $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$, we have that $N_j > N_k$. Hence, the team will pick an action $a_j \in \mathbf{Good}$, in all world states, if n is large enough (i.e., $n \to \infty$). ■

However, with a team made of copies of the same agent, the system is likely to lose the ability to generate new solutions as n increases. If, for each ω, we have an action a_m^ω such that $p_m^{Good} > p_j^{Good}$ $\forall a_m^\omega \neq a_j^\omega$, the team converges to picking only action a_m^ω (Proposition 1 below). Hence, $|\mathbf{S}| = 1$, which is a very negative result. Therefore, contrary to traditional social choice, here it is not the case that increasing the team size always improves performance.

Let $p_{Good} = \sum_j p_j^{Good}$ be the probability of picking any action in **Good**. We re-write the probability of an action a_j^{Good} as: $p_j^{Good} = \frac{p_{Good}}{|\mathbf{Good}|} + \lambda_j$, where $\sum_j \lambda_j = 0$. Let $\boldsymbol{\lambda}^+$ be the set of $\lambda_j > 0$. Let λ^{High} be the maximum possible value for $\lambda_j \in \boldsymbol{\lambda}^+$, such that the relation $p_j^{Good} > p_k^{Bad}$, $\forall a_j \in \mathbf{Good}, a_k \in \mathbf{Bad}$ is preserved. We show that when $z \to \infty$, $|\mathbf{S}|$ is the highest as $\max \boldsymbol{\lambda}^+ \to 0$, and the lowest (i.e., one) as $\min \boldsymbol{\lambda}^+ \to \lambda^{High}$.

Proposition 1. *The maximum value for* $|\mathbf{S}|$ *is* $\prod_\omega |\mathbf{Good}_\omega|$*. When* $z, n \to \infty$*, as* $\max \boldsymbol{\lambda}^+ \to 0$*,* $|\mathbf{S}| \to \prod_\omega |\mathbf{Good}_\omega|$*. Conversely, as* $\min \boldsymbol{\lambda}^+ \to \lambda^{High}$*,* $|\mathbf{S}| \to 1$*.*

Proof. As $\max \boldsymbol{\lambda}^{+} \to 0$, $\lambda_j \to 0$, $\forall a_j$. Hence, $E[N_j] \to n \times \frac{p_{Good}}{|\mathbf{Good}|}$, $\forall a_j \in \mathbf{Good}$. Because ties are broken randomly, at each world state ω, each $a_j \in \mathbf{Good}_\omega$ is selected by the team with equal probability $\frac{1}{|\mathbf{Good}_\omega|}$. As $E[N_j] = E[N_k]\ \forall a_j, a_k \in \mathbf{Good}$, we have that at each ω it is possible to choose $|\mathbf{Good}_\omega|$ different actions. Hence, there are $\prod_\omega |\mathbf{Good}_\omega|$ possible combinations of solutions. At each voting iteration, ties are broken at each ω randomly, and one possible combination is generated. As $z \to \infty$, eventually we cover all possible combinations, and $|\mathbf{S}| \to \prod_\omega |\mathbf{Good}_\omega|$.

Conversely, as $\min \boldsymbol{\lambda}^{+} \to \lambda^{High}$, $E[N_j] \to n \times p_j^{Good}$ for one fixed a_j such that $p_j^{Good} > p_k^{Good}, \forall a_j \neq a_k \in \mathbf{Good}$ (and, consequently, $E[N_j] > E[N_k]$), at each ω. Hence, there is no tie in any world state, and the team picks a fixed a_j at each world state. Therefore, even if $z \to \infty$, $|\mathbf{S}| \to 1$. ■

Therefore, uniform teams need a very strong assumption to be optimal: the probability of voting for optimal actions must be uniformly distributed over all optimal actions ($\max \boldsymbol{\lambda}^{+} \to 0$). We show that, alternatively, we can use agents with different "preferences" (i.e., "diverse" agents), to maximize $|\mathbf{S}|$. We consider here agents that have about the same ability in problem-solving, but they prefer different optimal actions. As the agents have similar ability, in order to simplify the analysis we consider the probabilities to be the same across agents, except for the actions in **Good**, as each agent ϕ_i has a subset $\mathbf{Good^i} \subset \mathbf{Good}$ consisting of its preferred actions (which are more likely to be chosen than other actions). We denote by p_{ij} the probability of agent ϕ_i voting for action a_j. $\forall a_j \in \mathbf{Good}^i$, let $p_{Good^i} = \sum_j p_{ij}$, $p_{ij} = \frac{p_{Good^i}}{|\mathbf{Good}^i|}$, and $p_{ij} > p_{ik}, \forall a_k \notin \mathbf{Good^i}$. $\mathbf{Good}^i \cap \mathbf{Good}^l$ (of agents ϕ_i and ϕ_l) is not necessarily $\emptyset$. Consider we can draw diverse agents from a distribution $\mathcal{F}$. Each agent ϕ_i has $r < |\mathbf{Good}|$ actions in its $\mathbf{Good}^i$, and we assume that all actions in **Good** are equally likely to be selected to form $\mathbf{Good}^i$ (since they are all equally optimal). Note that each agent can even prefer a single action ($r = 1$), so this is a realistic assumption. We show that by drawing n agents from $\mathcal{F}$, the team is optimal for large n with probability 1, as long as n is a multiple of a divisor (> 1) of each $|\mathbf{Good}_\omega|$. We also show that the minimum necessary optimal team size is constant with high probability as the number of world states grows. We start with the following proposition:

Proposition 2. *If a team of size n is optimal at a world state, then* $gcd(n, |\mathbf{Good}|) > 1$ *(n and* $|\mathbf{Good}|$ *are not co-prime).*

Proof. Prove by contradiction. By the optimality requirements we must have $nr = k|\mathbf{Good}|$, where k is a constant $\in \mathbb{N}_{>0}$ representing the number of agents that have a given action a_j in its $\mathbf{Good}^i$ – note that it must be the same for all optimal actions. If n and $|\mathbf{Good}|$ are co-prime, then it must be the case that r is divisible by $|\mathbf{Good}|$. However, this yields $r \geq |\mathbf{Good}|$, which contradicts our assumption. Therefore, n and $|\mathbf{Good}|$ are not co-prime.■

This implies hard restrictions for world states where $|\mathbf{Good}|$ is prime, or for teams with prime size n: if n is prime, $|\mathbf{Good}|$ must be a multiple of n; and if $|\mathbf{Good}|$ is prime, n must be a multiple of $|\mathbf{Good}|$.

Now we consider all world states $\mathbf{\Omega}$. For a team of fixed size n, Proposition 2 applies across all world states. Hence, the team's size must be a multiple of a divisor (> 1) of each $|\mathbf{Good}_\omega|$. Note that the pdfs of the agents (and also r) may change according to ω. Let $\mathbf{D}$ be a set containing one divisor of each world state (if two or more world states have a common divisor x, it will be representable by only one $x \in \mathbf{D}$). Hence, $\forall\omega$, $\exists d \in \mathbf{D}$, such that $d \mid |\mathbf{Good}_\omega|$; and $\forall d \in \mathbf{D}$, $\exists \mathbf{Good}_\omega$, such that $d \mid |\mathbf{Good}_\omega|$. There are multiple possible $\mathbf{D}$ sets, from the superset of all possibilities $\mathscr{D}$.

Therefore, we can now study the minimum size necessary for an optimal team. Applying Proposition 2 at each world state ω, we have that the minimum size necessary for an optimal team is $n = \min_{\mathbf{D}\in\mathscr{D}} \prod_{d\in\mathbf{D}} d$. Hence, our worst case is when each $|\mathbf{Good}_\omega|$ is a unique prime, as the team will have to be a product of each (unique) optimal action space sizes. This means that:

Proposition 3. *In the worst case, the minimum team size is exponential in the size of the world states $|\mathbf{\Omega}|$. In the best case, the minimum necessary team size is a constant with $|\mathbf{\Omega}|$.*

Proof. In the worst case, each added world state ω has a unique prime optimal action space size. Hence, the minimum team size is at least the product of the first $|\mathbf{\Omega}|$ primes, which, by the *prime number theorem*, has growth rate $\exp((1 + o(1))|\mathbf{\Omega}| \log |\mathbf{\Omega}|)$. In the best case, each added $\mathbf{Good}_\omega$ has a common divisor with previous ones, and the minimum necessary team size does not change. ■

However, we show that the worst case happens with low probability, and the best case with high probability. Let N be the maximum possible $|\mathbf{Good}|$, and assume that each new world state ω_j will have a uniformly randomly drawn number of optimal actions, denoted as m_j, for all $j = 1, \ldots, M$.

Proposition 4. *The probability that the minimum necessary team size grows exponentially tends to 0, and the probability that it is constant tends to 1, as $M, N \to \infty$.*

Proof. We need to show that the probability that $m_1, \ldots, m_{M-1}$ are all co-prime with m_M tends to 0 as $M, N \to \infty$. Assume $N \to \infty$, then given any prime p, the probability that at least one of any independently randomly generated $M-1$ numbers $m_1, ..., m_{M-1}$ has factor p is $1 - (1 - \frac{1}{p})^{M-1}$, while the probability that one independently randomly generated number m_M has factor p is $\frac{1}{p}$. Therefore, the probability m_M shares common factor p with at least one of $m_1, \ldots, m_{M-1}$ is $\frac{1-(1-\frac{1}{p})^{M-1}}{p}$. The probability that m_M is co-prime with all $m_1, \ldots, m_{M-1}$ is $\prod_{\text{all primes } p}[1 - \frac{1-(1-\frac{1}{p})^{M-1}}{p}]$; which tends to $\prod_{\text{all primes } p}(1 - \frac{1}{p}) = \frac{1}{\zeta(1)} = 0$, where $\zeta(s)$ is the Riemann zeta function, $\zeta(1) = \prod_{\text{all primes } p} \frac{1}{1-p^{-1}} = \sum_{i=1}^{\infty} \frac{1}{i} \to \infty$ (as shown by Euler). Hence, with high probability, when adding a new world state ω_j, $|\mathbf{Good}_{\omega_j}|$ will share a common factor with a world state already in $\mathbf{\Omega}$. ■

Finally, we show that a diverse team of agents is always optimal as the team grows, as long as it grows *carefully*:

Theorem 1. *Let* $\mathbf{D} \in \mathscr{D}$ *be a set containing one factor from each* $\mathbf{Good}_\omega$*. For arbitrary* n*, the probability that we can generate an optimal team of size* n *converges to 0 as* $|\mathbf{\Omega}| \to \infty$*. However, if* $n = c\prod_{d\in\mathbf{D}} d$*, then the probability that the team is optimal tends to 1 as* $c \to \infty$*.*

Proof. For arbitrary n, let $\mathbf{P}$ be the set of its prime factors. Given one $p \in \mathbf{P}$, the probability that p is not a factor of $|\mathbf{Good}_\omega|$ is $1 - 1/p$. The probability that all $p \in \mathbf{P}$ are not factors is: $\prod_p(1 - 1/p)$. As $0 < \prod_p(1 - 1/p) < 1$, the probability that at least one $p \in \mathbf{P}$ is a factor of $|\mathbf{Good}_\omega|$ is $1-\prod_p(1-1/p) < 1$. For $|\mathbf{\Omega}|$ tests, the probability that at least one p is a factor in all of them is: $(1-\prod_p(1-1/p))^{|\mathbf{\Omega}|}$, which $\to 0$, as $|\mathbf{\Omega}| \to \infty$. Hence, the probability that $\gcd(n, |\mathbf{Good}_\omega|) = 1$ for at least one ω tends to 1, and the probability that the team can be optimal tends to 0. However, if $n = c\prod_{d\in\mathbf{D}} d$, then $gcd(n, |\mathbf{Good}_\omega|) \neq 1\ \forall\omega \in \mathbf{\Omega}$. Let N_j be the number of agents ϕ_i that have a_j in its $\mathbf{Good}^i$. As each a_j has equal probability of being selected to be in a $\mathbf{Good}^i$, for a large number of drawings ($c \to \infty$), $P(N_i = N_j) \to 1, \forall a_i, a_j \in \mathbf{Good}_\omega, \forall\omega$ (*law of large numbers*). ■

If it is expensive to test values for n such that Theorem 1 is satisfied, we can choose $n = c\prod_\omega |\mathbf{Good}_\omega|$, as it immediately follows the conditions of the theorem. Moreover, if we know the size of all $|\mathbf{Good}_\omega|$, we can check if n and $|\mathbf{Good}_\omega|$ are co-prime in $O(h)$ time (where h is the number of digits in the smaller number), using the *Euclidean algorithm*. Hence, we can test all world states in $O(|\mathbf{\Omega}|h)$ time.

4.1 Generalizations

We first show that Theorem 1 still applies for agents ϕ_i with different probabilities over optimal actions p_{Good^i}. We consider a more general definition of *optimal* team: the difference between the probabilities of picking each optimal action must be as small as possible; i.e., let $p_j^{\mathbf{\Phi}}$ be the probability of team $\mathbf{\Phi}$ picking optimal action a_j, the optimal team is such that $\Delta := \sum_{a_k}\sum_{a_l} |p_k^{\mathbf{\Phi}} - p_l^{\mathbf{\Phi}}|, \forall a_k, a_l \in \mathbf{Good}$ is minimized (hence in the previous case $\Delta = 0$).

Proposition 5. *Theorem 1 still applies when* $|p_{Good^i} - p_{Good^j}| \leq \epsilon$*,* $\forall\phi_i, \phi_j$*, for small enough* $\epsilon > 0$*.*

Proof. Let $\mathbf{\Phi}$ be an optimal team, where p_{Good^i} is the same for all agents ϕ_i. Hence, the probability of all actions in $\mathbf{Good}$ being selected by the team is the same. I.e., $p_k^{\mathbf{\Phi}} = p_l^{\mathbf{\Phi}}, \forall a_k, a_l \in \mathbf{Good}$, and $\Delta = 0$.

We prove by mathematical induction. Assume we change the p_{Good^i} of x agents ϕ_i, and Δ is as small as possible. Now we will change $x + 1$ agents. Let's pick one agent ϕ_i and increase its p_{Good^i} by $\delta \leq \epsilon$. It follows that $p_k^{\mathbf{\Phi}} > p_l^{\mathbf{\Phi}}, \forall a_k \in \mathbf{Good^i}, a_l \notin \mathbf{Good^i}$, and the new $\Delta' := \sum_{a_k\in\mathbf{Good}}\sum_{a_l\in\mathbf{Good}} |p_k^{\mathbf{\Phi}} - p_l^{\mathbf{\Phi}}| > \Delta$.

If we add one more agent ϕ_j, such that $\mathbf{Good^j} \cap \mathbf{Good^i} = \emptyset$, the probability of voting for actions $a_m \in \mathbf{Good^j}$ increases. For small enough ϵ, p_{Good^j} will be too large to precisely equalize the probabilities, and it follows that

$p^{\mathbf{\Phi}}_m > p^{\mathbf{\Phi}}_k > p^{\mathbf{\Phi}}_l, \forall a_m \in \mathbf{Good^j}, a_k \in \mathbf{Good^i}, a_l \notin \mathbf{Good^i} \cup \mathbf{Good^j}$, and $\Delta'' := \sum_{a_k \in \mathbf{Good}} \sum_{a_l \in \mathbf{Good}} |p^{\mathbf{\Phi}}_k - p^{\mathbf{\Phi}}_l| > \Delta'$. The same applies for each newly added agent, until we have a new team such that $n = c \prod_{d \in \mathbf{D}} d$ (again, satisfying the conditions of the theorem).

The base case follows trivially. If we did not change the probability of any agent (i.e., $x = 0$), and we now increase p_{Good^i} of a single agent ϕ_i, $p^{\mathbf{\Phi}}_k > p^{\mathbf{\Phi}}_l, \forall a_k \in \mathbf{Good^i}, a_l \notin \mathbf{Good^i}$, and $\Delta' > \Delta$. By the same argument as before, adding more agents will only increase Δ', until $n = c \prod_{d \in \mathbf{D}} d$. ■

We also generalize to the case where the number of preferred actions r changes for each agent. Let the number of actions in the $\mathbf{Good^i}$ of agent ϕ_i (r^i) be decided according to a uniform distribution in the interval $[1, r']$.

Proposition 6. *If $n = r' \times c \prod_{d \in \mathbf{D}} d$, the probability that the team is optimal $\to 1$ as $c \to \infty$.*

Proof. For large n, the number of agents with $r^i = 1, \ldots, r'$ is the same. Therefore, if for each subset $\mathbf{\Phi^i} \subset \mathbf{\Phi}$, such that $r^\phi = i, \forall \phi \in \mathbf{\Phi^i}$, we have that $p^{\mathbf{\Phi^i}}_k = p^{\mathbf{\Phi^i}}_l$, $\forall a_k, a_l \in \mathbf{Good}$, we will have that $p^{\mathbf{\Phi}}_k = p^{\mathbf{\Phi}}_l, \forall a_k, a_l \in \mathbf{Good}$. Given an optimal team of size n, we have r' subsets $\mathbf{\Phi^i}$ of size n/r' each. It follows by Theorem 1 that $n/r' = c \prod_{d \in \mathbf{D}} d$, and $n = r' \times n/r' = r' \times c \prod_{d \in \mathbf{D}} d$. ■

In the next section we perform experiments with agents whose pdfs differ, and diverse teams still significantly outperform uniform teams.

5 Experiments

We run experiments with diverse and uniform teams (henceforth *diverse* and *uniform*). First, we run synthetic experiments, where we randomly create pdfs for the agents, and simulate voting iterations across a series of world states. We repeat all our experiments 100 times, and in the graphs we plot the average and the confidence interval of our results (with $p = 0.01$). We run 1000 voting iterations (z), and measure how many optimal solutions the team is able to find. We study a scenario where the number of actions ($|\mathbf{A}|$) = 100, and the number of optimal actions per world state ($|\mathbf{Good}_\omega|$) is, respectively: $< 2, 3, 5, 5, 5 >$, in a total of 750 optimal solutions.

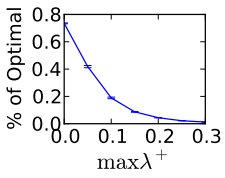


Fig. 1: Percentage as max $\boldsymbol{\lambda}^+$ grows.

At each repetition of our experiment, we randomly create a pdf for the agents. We start by studying the impact of $\max \boldsymbol{\lambda}^+$ in *uniform*. When creating the *uniform* team, the total probability of playing any of the optimal actions (i.e., p_{Good}) is randomly assigned (uniform distribution) between 0.6 and 0.8. We fix the size of the team (25) and evaluate different $\max \boldsymbol{\lambda}^+$ (Figure 1). As expected from Proposition 1, for $\max \boldsymbol{\lambda}^+ = 0$ the system finds the highest number of optimal solutions; and as $\max \boldsymbol{\lambda}^+$ increases, it quickly drops.

We then study the impact of increasing the number of agents, for *uniform* and *diverse*. To generate a *diverse* team, we draw randomly a r_ω in an interval U for each world state, that will be the size of $|\mathbf{Good}^i|$. We study three variants: *diverse**, where $U = (0, |\mathbf{Good}_\omega|]$; *diverse*, where $U = (0, |\mathbf{Good}_\omega|)$, and *diverse*$\Delta$, where we allow agents to have different r^i_ω, also drawn from $(0, |\mathbf{Good}_\omega|)$. We independently create pdfs randomly for each agent ϕ_i. For each agent we draw a number between 0.6 and 0.8 to distribute over the set of optimal actions, and randomly decide r_ω actions to compose its $\mathbf{Good}^i$ set. We distribute equally 80% of the probability of voting over optimal actions on the actions of that set.

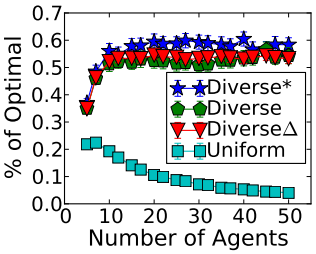


Fig. 2: Percentage of optimal solutions as # agents grows.

As we can see (Figure 2), the number of solutions *decreases* for *uniform* as the number of agents grows. Normally, in social choice, we expect the performance to improve, so this is a novel result. It is, however, expected from our Proposition 1. *Diverse*, on the other hand, improves in performance for all 3 versions, as predicted by our theory. However, the system seems to converge for a fixed z, as the performance does not increase much after around 20 agents. Hence, in Figure 3 we study larger *diverse* (continuous line) and *diverse*Δ teams (dashed line), going all the way up to 1800 agents. We also study four different number of voting iterations (z, shown in the figure by different lines): 1000, 2000, 3000, 4000. As we can see, although adding more agents was not really improving the performance in the experimental scenario under study, there is clearly a statistically significant improvement by increasing the number of voting iterations, with the system improving from around 53% of the optimal solutions, all the way up to finding more than 80% of them. However, there is a *diminishing returns* effect, as the impact of adding more iterations decreases as the actual number of iterations grows larger. We also note that *diverse*Δ is better than *diverse*, and the difference increases as z grows.

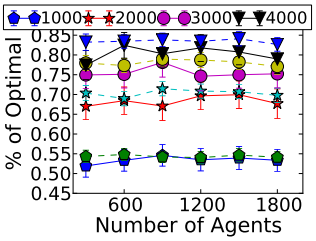
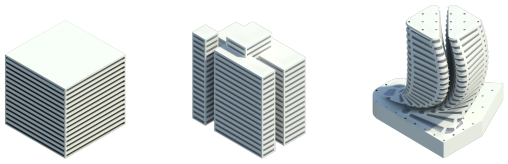


Fig. 3: Percentage for larger teams.

5.1 Experiments in Architecture Design

We study a real system for architectural building design. This is a fundamental domain, since the design of a building impacts its energy usage during its whole life-span [2, 13]. We use Beagle [8], a multi-objective design optimization software that assists users in the early stage design of buildings. Hence, the experiments presented here were run in an actual system, that performs expensive

energy evaluations over complex architectural designs, and represent months of experimental work.



(a) Base (b) Office Park (c) Contemporary

Fig. 4: Parametric designs with increasing complexity used in our experiments.

First, the designer creates a parametric design, containing (as discussed in Section 3) a set of parameters that can be modified within a specified range, allowing the creation of many variations. We use designs from Gerber and Lin (2013) [8]: *base*, a simple building type with uniform program (i.e., tenant type); *office park*, a multi-tenant grouping of towers; and *contemporary*, a double "twisted" tower that includes multiple occupancy types, relevant to contemporary architectural practices. We show the designs in Figure 4.

Beagle uses a Genetic Algorithm (GA) to optimize the building design based on three objectives: energy efficiency, financial performance and area requirements. In detail, the objective functions are: $S_{obj} : \max SPCS$; $E_{obj} : \min EUI$; $F_{obj} : \max NPV$. SPCS is the Spatial Programming Compliance Score, EUI is the Energy Use Intensity and NPV is the Net Present Value, defined as follows.

SPCS defines how well a building conforms to the project requirements (by measuring how close the area dedicated to different activities is to a given specification). Let **L** be a list of activities (in our designs, **L**=<Office, Hotel, Retail, Parking>), $area(l)$ be the total area in a building dedicated to activity l and $requirement(l)$ be the area for activity l given in a project specification. SPCS is defined as: $SPCS = 100 * \left(1 - \frac{\sum_{l \in \mathbf{L}} |area(l) - requirement(l)|}{|\mathbf{L}|}\right)$

EUI regulates the overall energy performance of the building. This is an estimated overall building energy consumption in relation to the overall building floor area. The process to obtain the energy analysis result is automated in Beagle through Autodesk Green Building Studio (GBS) web service.

Finally, **NPV** is a commonly used financial evaluation. It measures the financial performance for the whole building life cycle, given by: $NPV = \left(\sum_{t=1}^{T} \frac{c_t}{(1+r)^t}\right) - c_0$, where T is the Cash Flow Time Span, r is the Annual Rate of Return, c_0 is the construction cost, and $c_t = \text{Revenue} - \text{Operation Cost}$.

Many options affect the execution of the GA, including: initial population size, size of the population, selection size, crossover ratio, mutation ratio, maximum iteration. Further details about Beagle are at Gerber and Lin (2013) [8].

In the end of the optimization process, the GA outputs a set of solutions. These are considered "optimal", according to the internal evaluation of the GA, but are not necessarily so. As in our theory, for each parameter the assigned value is going to be one of the optimal ones with a certain probability. In fact, most of the solutions outputted by the GAs are later identified as sub-optimal and eliminated in comparison with better ones found by the teams.

We model each run of the GA as an agent ϕ. Each parameter of the parametric design is a world state ω, where the agents decide among different actions $\mathbf{A}$ (i.e., possible values for the current parameter). Our model assumes independent multiple voting iterations across all world states. However, as in general it could be expensive to pool agents for votes in a large number of iterations, we test a more realistic scenario by pooling only 3 solutions per agent, but running multiple voting iterations by aggregating over all possible combinations of them, in a total of 81 voting iterations.

Agent	PZ	SZ	CR	MR
Agent 1	12	10	0.8	0.1
Agent 2	18	8	0.6	0.2
Agent 3	24	16	0.55	0.15
Agent 4	30	20	0.4	0.25

Table 1: GA parameters for the diverse team. Initial Population and Maximum Iteration were kept as constants: 10 and 5, respectively. PZ = Population Size, SZ = Selection Size, CR = Crossover Ratio, MR = Mutation Ratio.

We create 4 different agents, using different options for the GA (as shown in Table 1). Contrary to the previous synthetic experiments, we are dealing here with real (and consequently complex) design problems. Hence, the true set of optimal solutions is unknown. We approach the problem in a comparative fashion: when evaluating different systems, we consider the union of the set of solutions of all of them. That is, let $\mathbf{H}_x$ be the set of solutions of system x; we consider the set $\mathcal{H} = \bigcup_x \mathbf{H}_x$. We compare all solutions in $\mathcal{H}$, and consider as optimal the best solutions in $\mathcal{H}$, forming the set of optimal solutions $\mathcal{O}$. We use the concept of pareto dominance: the best solutions in $\mathcal{H}$ are the ones that *dominate* all other solutions (i.e., they are better in all 3 factors). As we know which system generated each solution $o \in \mathcal{O}$, we estimate the set of optimal solutions $\mathbf{S_x}$ of each system.

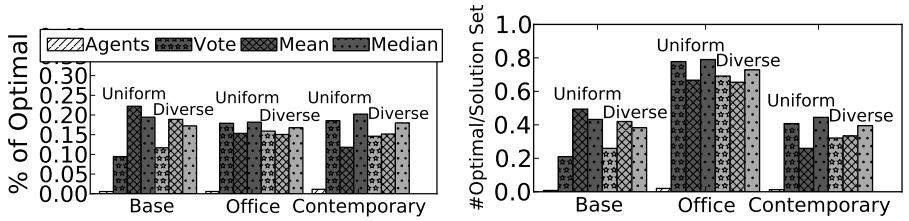
Although our theory focuses on *plurality voting* as the aggregation methodology, we also present results using the *mean* and the *median* of the opinions of the agents. That is, given one combination (a set of one solution from each agent), we also generate a new solution by calculating the *mean*/*median* across all parameters.

Concerning *uniform*, we evaluate a team composed of copies of the "best" agent. By "best", we mean the agent that finds the highest number of optimal

solutions. According to Proposition 1, such an agent should be the one with the lowest $\max \boldsymbol{\lambda}^+$, and we can predict that voting among copies of that agent generates a large number of optimal actions. Hence, for each design, we first compare all solutions of all agents (i.e., construct $\mathcal{H}$ as the union of the solutions of all agents), to estimate which one has the largest set of optimal solutions $\mathbf{S}$. We, then, run that agent multiple times, creating *uniform*. For *diverse*, we consider one copy of each agent.

We aggregate the solutions of *diverse* and *uniform*. We run 81 aggregation iterations (across all parameters/world states), by selecting 3 solutions from each agent ϕ_i, in its set of solutions $\mathbf{H}_i$, and aggregating all possible combinations of these solutions. We evaluate together the solutions of all agents and all teams (i.e., we construct $\mathcal{H}$ with the solutions of all systems), in order to estimate the size of $\mathbf{S}_\mathbf{x}$ of each system.

In Figure 5 (a), we show the percentage of optimal solutions for all systems, in relation to $|\mathcal{O}|$. For clarity, we represent the result of the individual agents by the one that had the highest percentage. As we can see, in all parametric designs the teams find a significantly larger percentage of optimal solutions than the individual agents. The agents find less than 1% of the solutions, while the teams are in general always close to or above 15%. In total (considering all aggregation methods and all agents), for all three parametric designs the agents find only about 1% of the optimal solutions, while *uniform* finds around 51% and *diverse* 47%. Looking at *vote*, in *base* *diverse* finds a larger percentage of optimal solutions than *uniform* (around 9.4% for *uniform*, while 11.6% for *diverse*). In *office park* and *contemporary*, however, *uniform* finds more solutions than *diverse*. Based on Proposition 1, we expect that this is caused by the best agent having a lower $\max \boldsymbol{\lambda}^+$ in *office park* and *contemporary* than in *base*.



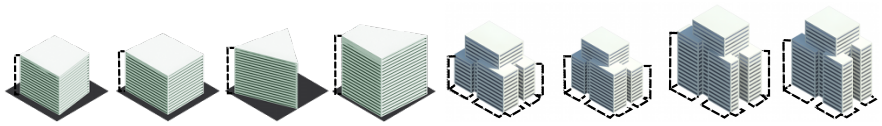
(a) Percentage in relation to all solutions found by all systems.

(b) Percentage in relation to the # of solutions of each system.

Fig. 5: Percentage of optimal solutions of each system.

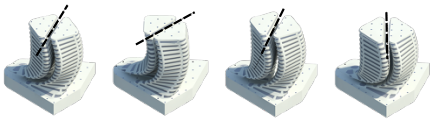
Figure 5 (b) shows the percentage of optimal solutions found, in relation to the size of the set of evaluated solutions of each system. That is, let $\mathbf{O}_x$ be the set of optimal solutions of system x, in $\mathcal{O}$. We show $\frac{|\mathbf{O}_x|}{|\mathbf{H}_x|}$. Concerning *vote*, the teams are able to find a new optimal solution around 20% of the time for *base*, around

73% of the time for *office park* and around 36% of the time for *contemporary*. Meanwhile, for the individual agents it is close to 0%. We can see that teams have a great potential in generating new optimal solutions, as expected from our theory. However, as studied in our synthetic experiments, we can expect some *diminishing returns* when increasing the number of voting iterations. We show examples of solutions created by the teams in Figure 6.



(a) Base. Shaded area shows variance of building's footprint in relation to site. Dashed line indicates height variance.

(b) Office Park. Dashed line shows variance in volume.



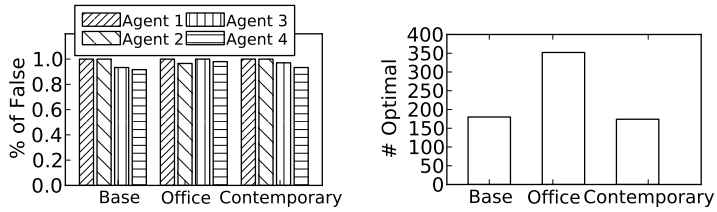
(c) Contemporary. Line shows variance in orientation.

Fig. 6: Some building designs generated by the teams.

We also plot in Figure 7 (a) the percentage of solutions that were reported to be optimal by each agent, but were later discovered to be suboptimal by evaluating $\mathcal{H}$. A large amount of solutions are eliminated (close to 100%), helping the designer to avoid making a poor decision, and increasing her confidence that the set of optimal solutions found represent well the "true" pareto frontier. Moreover, we test for duplicated solutions across different aggregation methods, different teams and different agents. The number is small: only 4 in *contemporary*, and none in *base* and *office park*. Hence, we are providing a high coverage of the pareto frontier for the designer. We show the total number of optimal solutions in Figure 7 (b). Finally, to better study the solutions proposed by the agents and teams, we plot all the optimal solutions in the factors space in Figure 8, where we show that the solutions give a good coverage of the pareto frontier.

6 Conclusion

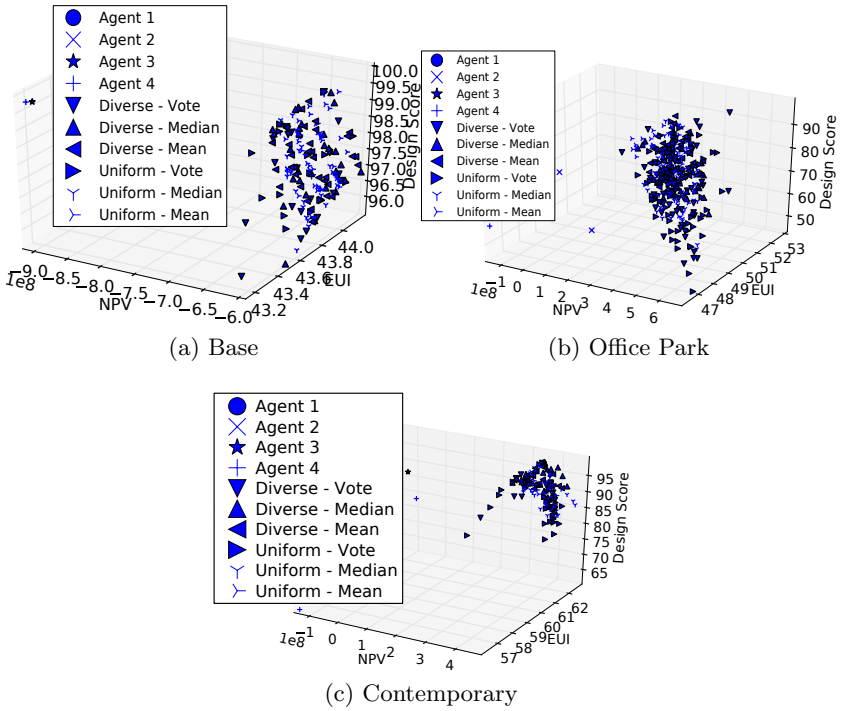
Design imposes a novel problem to social choice: maximize the number of optimal solutions. We present a novel model for agent teams, that shows the potential of a system of voting agents to be *creative*, by generating a large number of optimal solutions to the designer. Our analysis, which builds a new connection with *number theory*, presents several novel results: (i) uniform teams are in general



(a) False optimal solutions that are eliminated.

(b) Number of unique optimal solutions.

Fig. 7: Additional analysis.



(a) Base

(b) Office Park

(c) Contemporary

Fig. 8: All the optimal solutions in the factor space.

suboptimal, and converge to a unique solution; (ii) diverse teams are optimal as long as the team's size grows *carefully*; (iii) the minimum optimal team size is constant with high probability; (iv) the worst case for teams is a prime number of optimal actions. Our experiments consider bounded time and relaxed assumptions, and diverse teams still perform well. We show results in architecture, where teams find a large number of solutions for designing energy-efficient buildings.

Acknowledgments: This research is supported by MURI grant W911NF-11-1-0332, and the National Science Foundation under grant 1231001.

References

1. Bachrach, Y., Graepel, T., Kasneci, G., Kosinski, M., Van Gael, J.: Crowd IQ: aggregating opinions to boost performance. In: AAMAS. pp. 535–542 (2012)
2. Bogensttter, U.: Prediction and optimization of life-cycle costs in early design. Building, Research & Information 28, 376–386 (2000)
3. Boutilier, C.: A POMDP formulation of preference elicitation problems. In: AAAI (2002)
4. Caragiannis, I., Procaccia, A.D., Shah, N.: When do noisy votes reveal the truth? In: EC. pp. 143–160. ACM, New York, NY, USA (2013)
5. Conitzer, V., Sandholm, T.: Common voting rules as maximum likelihood estimators. In: UAI'05. pp. 145–152 (2005)
6. Elkind, E., Shah, N.: Electing the most probable without eliminating the irrational: Voting over intransitive domains. In: UAI (2014)
7. Erhan, H., Wang, I., Shireen, N.: Interacting with thousands: A parametric-space exploration method in generative design. In: Proceedings of the 2014 Conference of the Association for Computer Aided Design in Architecture. ACADIA (2014)
8. Gerber, D.J., Lin, S.H.E.: Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. Simulation (Apr 2013)
9. Globa, A., Donn, M., Moloney, J.: Abstraction versus cased-based: A comparative study of two approaches to support parametric design. In: ACADIA (2014)
10. Isa, I.S., Omar, S., Saad, Z., Noor, N., Osman, M.: Weather forecasting using photovoltaic system and neural network. In: Proceedings of the 2010 Second International Conference on Computational Intelligence, Communication Systems and Networks. pp. 96–100. CICSyN (July 2010)
11. Jiang, A.X., Marcolino, L.S., Procaccia, A.D., Sandholm, T., Shah, N., Tambe, M.: Diverse randomized agents vote to win. In: NIPS (2014)
12. Lin, S.h.E., Gerber, D.J.: Designing-In Performance : A Framework for Evolutionary Energy Performance Feedback in Early Stage Design. Automation and Construction 38, 59–73 (2012)
13. Lin, S.H.E., Gerber, D.J.: Evolutionary energy performance feedback for design: Multidisciplinary design optimization and performance boundaries for design decision support. Energy and Buildings 84, 426–441 (2014)
14. List, C., Goodin, R.E.: Epistemic democracy: Generalizing the Condorcet Jury Theorem. Journal of Political Philosophy 9, 277–306 (2001)
15. Mao, A., Procaccia, A.D., Chen, Y.: Better Human Computation Through Principled Voting. In: AAAI (2013)
16. Marcolino, L.S., Xu, H., Jiang, A.X., Tambe, M., Bowring, E.: Give a hard problem to a diverse team: Exploring large action spaces. In: AAAI (2014)
17. Nurmi, H.: Comparing Voting Systems. Springer (1987)
18. Polikar, R.: Ensemble learning. In: Zhang, C., Ma, Y. (eds.) Ensemble Machine Learning: Methods and Applications. Springer (2012)
19. Procaccia, A.D., Reddi, S.J., Shah, N.: A maximum likelihood approach for selecting sets of alternatives. In: UAI (2012)
20. Soufiani, H.A., Parkes, D.C., Xia, L.: Random utility theory for social choice. In: NIPS. pp. 126–134 (2012)
21. Vierlinger, R., Bollinger, K.: Accommodating change in parametric design. In: ACADIA (2014)
22. Xia, L., Conitzer, V.: A maximum likelihood approach towards aggregating partial orders. In: IJCAI (2011)

Quantified Degrees of Group Responsibility

Vahid Yazdanpanah and Mehdi Dastani

Utrecht University, The Netherlands
v.yazdanpanah@students.uu.nl, m.m.dastani@uu.nl

Abstract. This paper builds on an existing notion of group responsibility and proposes two ways to define the degree of group responsibility: *structural* and *functional* degrees of responsibility. These notions measure potential responsibilities of agent groups for avoiding a state of affairs. According to these notions, a degree of responsibility for a state of affairs can be assigned to a group of agents if, and to the extent that, the group of the agents have potential to preclude the state of affairs. These notions will be formally specified and their properties will be analyzed.

1 Introduction

The concept of responsibility has been extensively investigated in philosophy and computer science. Each proposal focuses on specific aspects of responsibility. For example, [1] focuses on the causal aspect of responsibility and defines a notion of graded responsibility, [2] focuses on the organizational aspect of responsibility, [3] argues that group responsibility should be distributed to individual responsibility, [4] focuses on the interaction aspect of responsibility and defines an agent's responsibility in terms of the agent's causal contribution, and [5] focuses on the strategic aspect of group responsibility and defines various notions of group responsibility. In some of these proposals, the concept of responsibility is defined with respect to a realized event "in past" while in other approaches it is defined as the responsibility for the realization of some event "in future". This introduces a major dimension of responsibility, namely backward-looking and forward-looking responsibility [6]. Backward-looking approaches reason about level of causality or contribution of agents in the occurrence of an already realized outcome while forward-looking notions are focused on the capacities of agents towards a state of affairs.

Although some of the existing approaches are designed to measure the degree of responsibility, they either constitute a backward-looking (instead of forward-looking) notion of responsibility [1], provide qualitative (instead of quantitative) levels of responsibility [7, 8], or focus on individual (instead of group) responsibility [4]. To our knowledge, there is no forward-looking approach that could measure the degree of group responsibility quantitatively. Such notion would enable reasoning on the potential responsibility of an agent group towards a state of affairs in strategic settings, e.g., collective decision making scenarios. In this paper, we build on a forward-looking approach to group responsibility and define two notions of responsibility degrees. The first concept is based on the partial or complete power of an agent group to preclude a state of affairs while the second concept is based on the potentiality of an agent group to reach a state where the

agent group possesses the complete power to preclude the state of affairs. This results in a distinction between what we will call the "structural responsibility" versus the "functional responsibility" of an agent group. In our proposal, an agent group has the full responsibility, if it has an action profile to preclude the state of affairs. All other agent groups that do not have full responsibility, but may have contribution to responsible agent groups, will be assigned a partial degree of responsibility.

The paper is structured as follows. In Section 2 we provide a brief analysis of the concept of group responsibility from a power-based point of view. Section 3 presents the framework in which our proposed notions will be formally characterized. In Sections 4 and 5 we introduce the notions that capture our conception of *degree of group responsibility* with respect to a given state of affairs and analyse their properties. Finally, concluding remarks and future work directions will be presented in Section 6.

2 Group Responsibility: A Power-based Analysis

In order to illustrate our conception of group responsibility and the nuances in degrees of responsibility, we follow [1] and use a voting scenario to explain the degree of responsibility of agents' groups for voting outcomes. The voting scenario considers a small congress with ten members consisting of five Democrats (D), three Republicans (R), and two Greens (G). We assume that there is a voting in progress on a specific bill (B). Without losing generality and to reduce the combinatorial complexity of the setting, we assume that all members of a party vote either in favour of or against the bill B. Table 1 illustrates the eight possible voting outcomes. Note that in this scenario, six positive votes are sufficient for the approval of B. For example, row 4 shows the case where R and D vote against B and the bill is disapproved. For this case we say that the coalition (group) of R and D vote against B. It should also be noted that our assumption reduces parties to individual agents with specific weights such that the question raises as why we use this party setting instead of a simple voting of three agents whose votes have different weights. The motivation is that this setting is realistic and makes the weighted votes of each agent (party) more intuitive.

Table 1. Voting results

	G(2)	R(3)	D(5)	Result
0	−	−	−	−
1	−	−	+	−
2	−	+	−	−
3	−	+	+	+
4	+	−	−	−
5	+	−	+	+
6	+	+	−	−
7	+	+	+	+

Table 2. War incidence

	Congress	President	War
0	−	−	−
1	−	+	−
2	+	−	−
3	+	+	+

Following [5] we believe that it is reasonable to assign the responsibility for a specific state of affairs to a group of agents if they jointly have the power to avoid the state of affairs[1]. According to [9], the preclusive power is a ability of

[1] See [5] for a detailed discussion on why to focus on avoiding instead of enforcing a state of affair.

a coalition to preclude a given state of affairs which entails that a coalition with preclusive power, has the potential but might not practice the preclusion of a given state of affairs. For our voting scenario, this suggests to assign responsibility to the coalition consisting of parties R and D since they can jointly disapprove B. Note that the state of affair to be avoided can also be the state of affairs where B is disapproved. In this case, the coalition can be assigned the responsibility to avoid disproving B. Similarly, coalitions GD and GRD have preclusive power with respect to the approval of B as they have sufficient members (weights) to avoid the approval of B. Note that none of the other four coalitions, i.e., G, R, D, and GR, could preclude the approval of B independently. Hence, we consider GD, RD, and GRD as being responsible coalitions for the approval of B. The intuition for this concept of responsibility is supported by the fact that the lobby groups are willing (i.e., it is economically rational) to invest resources in parties that have the power to avoid a specific state of affairs.

We build on the ideas in [5] and propose two orthogonal approaches to capture our conception of *degree* of group responsibility towards a state of affairs. Our intuition suggests that the *degree* of responsibility of a group of agents towards a state of affairs should reflect the extent they structurally or functionally can contribute to the coalitions that have preclusive power with respect to the state of affairs. In the sequel, we will explain the conception of *degree of responsibility* according to the structural and functional approaches, and illustrate both approaches by means of our voting scenario example.

Our conception of *structural responsibility degree* is based on the following observation in the voting scenario. We deem that regarding the approval of B, although the coalitions G, R, D, and GR have no preclusive power independently, they nevertheless have a share in the composition of coalitions GD, RD and GRD with preclusive power with respect to the approval of B. Hence, we say that any coalition that shares members with responsible coalitions, should be assigned a degree of responsibility that reflects its proportional contribution to the coalitions with preclusive power. For example, coalition D with five members, has larger share in GD than the coalition G has. Moreover, coalition D has a larger share in GD than in RD and GRD. Therefore, we believe that the relative size of a coalition and its share in the coalitions with the preclusive power are substantial parameters in formulation of the notion of responsibility degree. In this case, the larger share of D in GD in comparison with the share of G in GD will be positively reflected in D's responsibility degree, and the inequality of shares of D in three coalitions GD, RD and GRD, will be taken into account in formulating the responsibility degree of D. These parameters will be explained in details later. We would like to emphasize that this concept of responsibility degree is supported by the fact that the lobby groups do proportionally support the parties that can play a role in some key decisions.

The second approach in capturing the notion of *functional responsibility degree* addresses the dynamics of preclusive power of a specific coalition. Suppose that the bill B was about declaration of the congress to the President (P) which enables P to start a war (Table 2). Roughly speaking, P will be in charge only after the approval of the congress. When we are reasoning at the moment when

the voting is in progress in the congress, it is reasonable to assume that coalitions GD, RD, and GRD are responsible as they have preclusive power to avoid the war. Moreover, after the approval of B, the President P is the only coalition with preclusive power to avoid the war. Hence, we believe that although P alone would not have the preclusive power before the approval of B in the congress, it is rationally justifiable for an anti-war campaign to invest resources on P, even before the approval voting of the congress, simply because there exists possibilities where P will have the preclusive power to avoid the war. Accordingly, a reasonable differentiation could be made between the coalitions which do have the chance of acquiring the preclusive power and those they do not have any chance of power acquisition. This functional notion of responsibility degree addresses the eventuality of a state in which an agent group possesses the preclusive power regarding a given state of affairs.

Note that following [5], our notions of group responsibility are locally bounded as they will be defined with respect to some source state. Hence, a coalition might be responsible in a specific state and not responsible in the other states regarding a given state of affairs. Additionally, our proposed notions for responsibility degree have dependency to the global setting. In the voting scenario, the global setting that ten voters are situated in three parties of G (2 members), R (3 members) and D (5 members), is crucial for the responsibility degrees that are assigned to various coalitions. Any change in the global setting may alter the responsibility degree of various coalitions. For example, when two members of the Republican party secede from R and form a new Tea Party T, we face a different global setting, which in turn causes the responsibility degrees assigned to various coalition to change. This is due to the fact that the new setting introduces new coalitions such as TD with preclusive power regarding the approval of B. Our analysis is not limited to the voting scenarios, but can be applied to other situations as shown later in this paper.

3 Preliminaries: Coalitional Responsibility

The behaviour of a multi-agent system is often modelled by concurrent game structures (CGS) [10]. A concurrent game structure is a tuple $M = (N, Q, Act, d, o)$, where $N = \{1, ..., k\}$ is a nonempty finite set of agents, Q is a nonempty set of system states, Act is a nonempty and finite set of atomic actions, $d : N \times Q \to \mathcal{P}(Act)$ is a function that identifies the set of available actions for each agent $i \in N$ at each state $q \in Q$, and o is a deterministic and partial transition function that assigns a state $q' = o(q, \alpha_1, ..., \alpha_k)$ to a state q and an action profile $(\alpha_1, ..., \alpha_k)$ such all k agents in N choose actions in the action profile respectively. An action profile $\bar{\alpha} = (\alpha_1, ..., \alpha_k)$ is a sequence that consists of actions $\alpha_i \in d(i, q)$ for all players in N. In addition, in case $o(q, \alpha_1, ..., \alpha_k)$ is undefined then $o(q, \alpha'_1, ..., \alpha'_k)$ is undefined for each action profile $(\alpha'_1, ..., \alpha'_k)$. For the sake of notation simplicity, $d(i, q)$ will be written as $d_i(q)$ and $d_C(q) := \prod_{i \in C} d_i(q)$. Finally, in the rest of this paper a *state of affairs* refers to a set $S \subseteq Q$, $\bar{S}$ denotes the set $Q \setminus S$, and $(\alpha_C, \alpha_{N \setminus C})$ denotes the action profile, where α_C is the actions of the agents in coalition C and $\alpha_{N \setminus C}$ denotes the actions of the rest of the agents.

Following the settings of [5], for a CGS M, a specific state q and a state of affairs S, we recall the definitions of *q-enforce*, *q-avoid*, *q-responsible* and *weakly q-responsible* (See [5] for details and properties of these notions).

- Coalition C can *q-enforce* S in M iff there is a joint action $\alpha_C \in d_C(q)$ such that for all joint actions $\alpha_{N\setminus C} \in d_{N\setminus C}(q)$, $o(q,(\alpha_C,\alpha_{N\setminus C})) \in S$.
- Coalition C can *q-avoid* S in M iff for all $\alpha_{N\setminus C} \in d_{N\setminus C}(q)$ there is $\alpha_C \in d_C(q)$ such that $o(q,(\alpha_C,\alpha_{N\setminus C})) \in Q\setminus S$.
- Coalition $C \subseteq N$ is *q-responsible* for S in M iff C can q-enforce $\bar{S}$ and for all other coalitions C' that can q-enforce $\bar{S}$, we have that $C \subseteq C'$.
- A coalition $C \subseteq N$ is *weakly q-responsible* for S in M [2] iff C is a minimal coalition that can q-enforce $\bar{S}$.

Considering the voting scenario from the Section 2, coalitions GD, RD and GRD can both q_s-enforce and q_s-avoid the approval of B where q_s denotes the starting moment of the voting progress. Note that the notions of q-enforce and q-avoid correlate with the notions of, respectively, α-effectivity and β-effectivity in [11]. In this scenario, we have no q_s-responsible coalition for approval of B and two coalitions of GD and RD are weakly q_s-responsible for the approval of B. Note that the coalition GRD is not weakly q_s-responsible for the approval of B as it is not minimal. The concept of (weakly) q-responsible coalition, assigns responsibility to only coalitions with preclusive power and considers all other coalitions as not being responsible. As we have argued in section 2, we believe that responsibility can be assigned to all coalitions, even those without preclusive power, though to a certain degree including zero degree. In order to define our notions of responsibility degree, we first introduce two notions of *structural power difference* and *power acquisition sequence*. Given an arbitrary coalition C, a state q, and a state of affair S, the first notion concerns the number of missing elements in C that when added to C makes it a (weakly) q-responsible coalitions for a S, and the second notion concerns a sequence of action profiles from state q that leads to a state q' where C is (weakly) q'-responsible for S. According to the first notion, coalition C can gain preclusive power for S if supported by some additional members, and according to the second notion C can gain preclusive power for S in some potentially reachable state.

Definition 1 (Power measures) *Let M be a multi-agent system, S a state of affairs in M, C an arbitrary coalition, and $\hat{C}$ be a q-responsible coalition for S in M. We say that the structural power difference of C and $\hat{C}$ in state q of M with respect to S, denoted by $\Theta_q^{S,M}(\hat{C},C)$, is equal to cardinality of $\hat{C}\setminus C$. Moreover, we say that C has a power acquisition sequence $\langle\bar{\alpha_1},...,\bar{\alpha_n}\rangle$ in q for S in M iff $o(q_i,\bar{\alpha_i}) = q_{i+1}$ for $1 \leq i \leq n$ such that $q = q_1$ and $q_{n+1} = q'$ and C is (weakly) q'-responsible for S in M.*

Consider the war approval declaration of the Congress to the President (P) in Section 2. Here, we can see that the structural power difference of the coalition D and the weakly q_s-responsible coalition GD is equal to 2. Moreover, the singleton coalition P that is not responsible in q_s has the opportunity of being responsible

[2] In further references, "in M" might be omitted wherever it is clear from the context.

for the war in states other than q_s. Note that power acquisition sequence does not necessarily need to be unique. If the coalition C is not (weakly) responsible in q, the existence of any power acquisition sequence with a length higher than zero implies that the coalition could potentially reach a state q' (from the current state of q) where C is (weakly) q'-responsible for S. This notion also covers the cases where C is already in a (weakly) responsible state where the minimum length of power acquisition sequence is taken to be zero. In this case, the coalition is already (weakly) q-responsible for S. For example, in the voting scenario, coalition GD is weakly responsible for the state of affairs and therefore, the minimum length of a power acquisition sequence is zero. When we are reasoning in a source state of q, the notion of *power acquisition sequence*, enables us to differentiate between the non (weakly) q-responsible coalitions that do have the opportunity of becoming (weakly) q'-responsible for a given state of affairs ($q \neq q'$) and those they do not. Moreover, we emphasise that the availability of a *power acquisition sequence* for an arbitrary coalition C from a source state q to a state q' in which C is (weakly) q-responsible for the state of affairs, does not necessitate the existence of an independent strategy for C to reach q' from q.

4 Structural Degree of Responsibility

Structural degree of responsibility addresses the preclusive power of a coalition for a given state of affairs by means of the *maximum* contribution that the coalition has in a (weakly) responsible coalition for the state of affairs. To illustrate the intuition behind this notion, consider again the voting scenario in the section 2. If an anti-war campaign wants to invest its limited resources to prevent the bill to go to a war, we deem that it is reasonable to invest more on D than G, if the resources admit such a choice. Although neither D nor G could prevent the war individually, larger contribution of D in coalitions with preclusive power, i.e. GD, RD and GRD, entitles D to be assigned with larger degree of responsibility than G. This intuition will be reflected in the formulation of *structural degree of responsibility*.

Definition 2 (Structural degree of responsibility) *Let $\mathbb{W}_q^{S,M}$ denote the set of all (weakly) q-responsible coalitions for S in M and C be an arbitrary coalition. The structural degree of q-responsibility of C for S in M, denoted $\mathcal{SDR}_q^{S,M}(C)$, is defined as follows:*

$$\mathcal{SDR}_q^{S,M}(C) = max(\{1 - \frac{\Theta_q^{S,M}(\hat{C},C)}{|\hat{C}|} \quad | \quad \hat{C} \in \mathbb{W}_q^{S,M}\}).$$

Intuitively, $\mathcal{SDR}_q^{S,M}(C)$ measures the highest contribution of a coalition C in a (weakly) q-responsible $\hat{C}$ for S. Hence, for all possible coalitions, structural degree of responsibility is in range of $[0, 1]$. In sequel, we write $\mathcal{SDR}_q^{S}(C)$ and $\mathbb{W}_q^{S}$ instead of $\mathcal{SDR}_q^{S,M}(C)$ and $\mathbb{W}_q^{S,M}$, respectively.

Proposition 1 (Full structural responsibility). *The structural degree of q-responsibility of coalition C for S is equal to 1 iff C is either a (weakly) q-responsible coalition for S or $C \supseteq \hat{C}$ such that $\hat{C}$ is (weakly) q-responsible for S.*

Proof. Follows directly from Definition 2 and definition of (weak) responsibility in [5]. □

Example 1. Consider again the voting scenario from Section 2 (Figure 1). In this scenario, we have an initial state q_s in which all voters can use their votes in favour or against the approval of the bill B (no abstention or null vote is allowed). The majority of six votes (or more) in favour of B will be considered as the state of affairs consisting of states q_7, q_5 and q_3. This multi-agent system can be modelled as CGS $M = (N, Q, Act, d, o)$, where $N = \{1, ..., 10\}$, $Q = \{q_s, q_0, ..., q_7\}$, $Act = \{0, 1, wait\}$, $d_i(q_s) = \{0, 1\}$ and $d_i(q) = \{wait\}$ for all $i \in N$ and $q \in Q \setminus \{q_s\}$. Voters are situated in three parties such that $G = \{1, 2\}$, $R = \{3, 4, 5\}$ and $D = \{6, 7, 8, 9, 10\}$. For notation convenience, actions of party members will be written collectively in the action profiles, e.g., we write $(0, 1, 0)$ to denote the action profile $(0, 0, 1, 1, 1, 0, 0, 0, 0, 0)$. The outcome function is as illustrated in Figure 1 (e.g., $o(q_s, (0, 0, 1)) = q_1$ is illustrated by the arrow from q_s to q_1). Moreover, the simplifying assumption that all party members vote collectively is implemented by $o(q_s, \bar{\alpha}') = q_s$ for all possible action profiles $\bar{\alpha}'$ in which party members act differently. We observe that the set of weakly q_s-responsible coalitions in this example is $\{GD, RD\}$. Using Definition 2, the structural degree of q_s-responsibility of GR will be equal to $max(\{2/7, 3/8\}) = 3/8$. A similar calculation leads to the conclusion that the structural degree of q_s-responsibility for all (weakly) q_s-responsible coalitions and the coalition GRD is equal to 1. The structural degree of q_s-responsibility of empty coalition ($\emptyset$) is equal to 0 as the structural power difference of the empty coalition with all (weakly) q_s-responsible coalitions $\hat{C}$ is equal to the cardinality of $\hat{C}$. The coalition D shares members with both coalitions of GD and RD where D has its largest share in GD. So, $\mathcal{SDR}^S_{q_s}(D) = max(\{5/7, 3/8\}) = 5/7$. A similar calculation shows that $\mathcal{SDR}^S_{q_s}(R) = 3/8$ and $\mathcal{SDR}^S_{q_s}(G) = 2/7$.

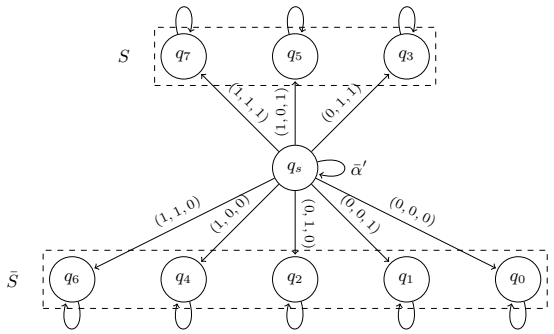


Fig. 1. Voting scenario

As illustrated, a coalition C might share members with various (weakly) q-responsible coalitions, therefore the largest structural share of C in (weakly) q-responsible coalitions for S, will be considered to form the $\mathcal{SDR}^S_q(C)$. We would like to stress that our notions for responsibility degrees are formulated based on the maximum expected power of a coalition to preclude a state of

affairs. While we believe that in legal theory, and with respect to its backward-looking approach, the minimum preclusive power of a coalition need be taken into account for assessing culpability, our focus as a forward-looking approach will be on maximum expected preclusive power of a coalition regarding a given state of affairs.

The following lemma introduces a responsibility paradox case in which our presented notion of *structural degree of responsibility* is not applicable as a notion for reasoning about responsibility of groups of agents.

Lemma 1 (Applicability constraint: responsibility paradox). *The empty coalition is (unique) q-responsible for S iff the structural degree of q-responsibility of all possible coalitions C for S is equal to 1.*

Proof. "$\Rightarrow$": Based on Proposition 1, if the empty coalition ($\emptyset$) is q-responsible for S, the structural degree of q-responsibility of the empty coalition and all its super-coalitions, i.e., all possible coalitions, is equal to 1.

"$\Leftarrow$": According to Proposition 1, and because the empty coalition is only a super-coalition of itself, the premise entails that the empty coalition must be either a weakly q-responsible coalition for S or the unique q-responsible coalition for S. Based on [5], if the empty coalition is weakly q-responsible for S, then it is the q-responsible coalition for S. □

The common avoidability of S implies that the occurrence of S is impossible by means of any action profile in q. In other words, given the specification of a CGS model M, a state of of affairs S and a source state q in M, no action profile $\bar{\alpha}$ leads to a state $q_s \in S$. Common avoidability of a state of affairs, correlates with the impossibility notion $\neg\Diamond S$ in modal logic [12]. An impossible state of affairs S in q, entitles all possible coalitions to be "fully responsible". The impossibility of S neutralizes the space of coalitions with respect to their structural degree of q-responsibility for S. Therefore, we believe that in cases where the empty coalition is responsible for a given state of affairs, as S is impossible, full degree of structural responsibility of a coalition is not an apt measure, does not imply the preclusive power of any coalition, and hence, not an applicable reasoning notion for one who is willing to invest resources in the groups of agents that have the preclusive power over S. Note that in case the empty set is not responsible for S, its structural degree of responsibility is equal to 0 because its structural power difference with all (weakly) responsible coalitions $\hat{C}$ is equal to the cardinality of $\hat{C}$.

The next theorem illustrates a case in which a singleton coalition posses the preclusive power over a state of affairs. The existence of such a *dictator* agent in a state q, polarizes the space of all possible coalitions with respect to their structural degree of q-responsibility for the state of affairs.

Theorem 1 (Polarizing dictatorship). *Let $\hat{C}$ be a singleton coalition, q an arbitrary state and S a possible state of affairs (in sense of Lemma 1). Then, $\hat{C}$ is a (unique) q-responsible coalition for S iff for any arbitrary coalition C, $\mathcal{SDR}_q^S(C) \in \{0,1\}$, where $\mathcal{SDR}_q^S\ (C \in I) = 1$ and $\mathcal{SDR}_q^S(C \in O) = 0$ for $I = \{C | C \supseteq \hat{C}\}$ and $O = \{C | C \not\supseteq \hat{C}\}$.*

Proof. "⇒": Based on Proposition 1, the structural degree of q-responsibility of any coalition $C \supseteq \hat{C}$ is equal to 1. In other cases, the structural degree of q-responsibility of $C \not\supseteq \hat{C}$ is equal to 0 because C shares no element with $\hat{C}$, which is the singleton (unique) q-responsible coalition for S.

"⇐": Here we have a partition $W = \{I, O\}$ of all possible coalitions. As S is not an impossible state of affair in sense of Lemma 1, the empty coalition is not q-responsible for S but has the structural degree of q-responsibility equal to 0; and therefore a member of O. I as a set of all coalitions with structural degree of responsibility equal to 1, is a non-empty set either; because there exists at least one coalition in I which is $\hat{C}$. Hence, $\mathcal{SDR}_q^S(\hat{C} \in I) = 1$ and necessarily there exists at least one non-empty weakly q-responsible coalition for S, i.e., $\mathbb{W}_q^S \neq \emptyset$. Accordingly, based on Proposition 1, and as $\hat{C}$ is a singleton, $\hat{C} \in \mathbb{W}_q^S$. Moreover, based on Proposition 1, we have that $\mathbb{W}_q^S \subseteq I$. As $\hat{C}$ is a subset of all coalitions in I, we conclude that $\hat{C} \subseteq \mathbb{W}_q^S$. Thus, $\hat{C}$ is a weakly q-responsible coalition and is a subset of all possible weakly q-responsible coalitions for S. Therefore, $\hat{C}$ is the unique singleton q-responsible coalition or the *q-dictator* for S. □

Example 2 (Operating room scenario). Consider a surgery operation room where a patient is going to be operated. In this surgery operation a surgeon D, a surgeon assistant A and an anesthesiologist N are involved. In this scenario, each agent, i.e., D, A and N, can decide to perform her role in health-care delivery or to refuse. If the anesthesiologist chooses to refuse or if both the surgeon and the assistant decide to refuse, the patient will die. When all three agents choose to perform their tasks, the patient will recover in the state of *good health*. Finally, an exclusive refusal of the assistant or the surgeon, results in *medium health* or *infirm health*, respectively. This multi-agent scenario can be modelled as a CGS M, as shown in Figure 2. This CGS is specified as $M = (\{D, A, N\}, \{q_s, q_1, q_2, q_3, q_4\}, \{perform, refuse, wait\}, d, o)$ where $d_i(q_s) = \{perform, refuse\}$ and $d_i(q) = \{wait\}$ for all $i \in \{D, A, N\}$ and $q \in \{q_1, q_2, q_3, q_4\}$. The outcome function o is shown in the Figure 2, e.g. $o(q_s, (perform, refuse, perform)) = q_2$. The star $\star$ represents any available action, i.e. $\star \in \{perform, refuse\}$. In this example the weakly q_s-responsible coalitions for death of the patient (q_4) are DN and AN. Hence, in the structural degree of q_s-responsibility of all possible coalitions, i.e., D, A, N, DA, DN, AN, and DAN, for q_4, could be measured based on their maximum contribution in DN and AN. Accordingly, the structural degree of q_s-responsibility of coalitions D, A, N and DA will be 1/2. All coalitions of DN, AN and DAN have the structural degree of q_s-responsibility equal to 1 which reflects their preclusive power to avoid the death of P.

As our concept of group responsibility is based on the preclusive power of a coalition over a given state of affairs, the following monotonicity property shows that increasing the size of a coalition by adding new elements, does not have a negative effect on the preclusive power. This property, as formulated below, correlates with the monotonicity of power and power indices [13, 14].

Proposition 2 (Structural monotonicity). *Let C and C' be two arbitrary coalitions such that $C \subseteq C'$. Then, $\mathcal{SDR}_q^S(C) \leq \mathcal{SDR}_q^S(C')$.*

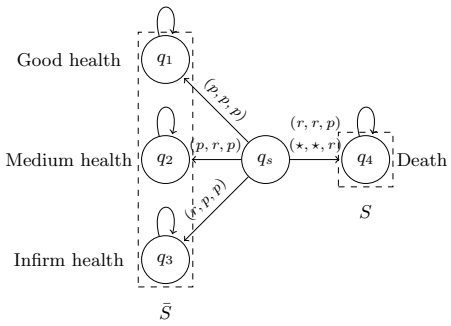


Fig. 2. Operating room scenario

Proof. By definition, structural degree of q-responsibility of C for S, reflects the maximum share of C in all possible (weakly) q-responsible coalitions for S. Hence, as the structural degree of q-responsibility has a value in range $[0, 1]$, the elements in $C' \setminus C$ could have no negative effect on this degree. □

Note that the other way does not hold in general; because the structural degree of q-responsibility of the coalitions C and C', might be formulated based on their maximum contribution in two distinct weakly q-responsible coalitions. Consider the operating room scenario in Example 2. As presented, $\mathcal{SDR}^S_{q_s}(A) = 1/2 \leq \mathcal{SDR}^S_{q_s}(DN) = 1$ but $A \nsubseteq DN$.

The following theorem shows that in case of existence of a unique nonempty q-responsible coalition for a state of affairs, the structural degree of q-responsibility of any coalition could be calculated cumulatively based on the degrees of disjoint subsets. In this case, for any two arbitrary coalitions C_1 and C_2, the summation of their structural degree of q-responsibility will be equal to the degree of the unified coalition.

Theorem 2 (Conditional cumulativity). *If there exists a nonempty (unique) q-responsible coalition for S, then for any arbitrary coalition C and partition $P = \{C_1, ..., C_n\}$ of C, we have $\sum_{i=1}^{n} \mathcal{SDR}^S_q(C_i) = \mathcal{SDR}^S_q(C)$.*

Proof. Suppose $\hat{C}$ is the q-responsible coalition for S. Then, as $\hat{C}$ is unique, the structural degree of q-responsibility of any coalition $C_i \in P$, could be reformulated based on its contribution to $\hat{C}$. Thus, $\sum_{i=1}^{n} \mathcal{SDR}^S_q(C_i)$ is equal to $\sum_{i=1}^{n} \frac{|\hat{C} \cap C_i|}{|\hat{C}|}$. The whole equation is equal to $\frac{1}{|\hat{C}|} \sum_{i=1}^{n} |\hat{C} \cap C_i|$. Hence, as P is a partition of C, we have $\frac{|\hat{C} \cap C|}{|\hat{C}|}$ which is equal to $\mathcal{SDR}^S_q(C)$. □

Note that in general, the other way does not hold. Consider the cases that the state of affairs is not avoidable by any coalition. Thus, no (weakly) q-responsible coalition does exist and the structural degree of q-responsibility for all possible coalitions is equal to 0. This situation satisfies the premise that for any arbitrary coalition C and partition P of C, summation of structural q-responsibility degrees of coalitions C_i is equal to the structural degree of q-responsibility of C; but no (weakly) q-responsible coalition for S do exists.

Example 3. Consider a variation of the operating room scenario in Example 2. Here we audit the structural degree of q_s-responsibility of coalitions for the state of affairs $S = \{good-health, medium-health, infirm-health, death\}$. Hence, no coalition has the preclusive power over S and therefore the structural degree of q_s-responsibility of all possible coalitions is equal to 0. For example, although the summation of structural degrees of q_s-responsibility of coalitions AD and N is equal to 0 and equal to structural degree of q_s-responsibility of ADN, there exists no (weakly) q_s-responsible coalition for S.

5 Functional Degree of Responsibility

Functional degree of responsibility addresses the dynamics of preclusive power of a specific coalition with respect to a given state of affairs. We remind the example from Section 2 where the President will be in charge, regarding the war decision, only after the approval of the Congress. It is our understanding that the existence of a sequence of action profiles that leads to a state where the President becomes responsible for the war decision rationalizes the investment of an anti-war campaign on the President, even before the approval of the Congress.

The functional degree of responsibility of a coalition C in a state q will be calculated based on the notion of *power acquisition sequence* by tracing the number of necessary state transitions from q, in order to reach a state q' in which the coalition C is (weakly) q'-responsible for S. The length of a shortest power acquisition sequence form q to q', illustrates the *potentiality* of preclusive power of the coalition C. If two coalitions have the capacity of reaching a state in which they have the preclusive power over the state of affairs S, we say that a coalition which has the shorter path has a higher potential preclusive power and thus gets the larger functional degree of responsibility. Accordingly, the coalition which is already in a responsible state, has full potential to avoid a state of affairs. Hence, it will be assigned with maximum functional degree of responsibility equal to one.

Definition 3 (Functional degree of responsibility) *Let $\mathbb{P}_q^{S,M}(C)$ denote the set of all power acquisition sequences of coalition C in q for S in M. Let also $\ell = min(\{length(k) \mid k \in \mathbb{P}_q^{S,M}(C)\})$ be the length of a shortest power acquisition sequence. The functional degree of q-responsibility of C for S in M, denoted by $\mathcal{FDR}_q^{S,M}(C)$, is defined as follows:*

$$\mathcal{FDR}_q^{S,M}(C) = \begin{cases} 0 & \text{if } \mathbb{P}_q^{S,M}(C) = \emptyset \\ \frac{1}{(\ell+1)} & \text{otherwise} \end{cases}$$

The notion of $\mathcal{FDR}_q^{S,M}(C)$ is formulated based on the minimum length of power acquisition sequences , which taken to be 0 if C is a (weakly) q-responsible coalition for S. In such a case, C has already an action profile to avoid S in q, and hence, the functional degree of q-responsibility of C for S will be equal to 1. If no power acquisition sequence k does exist for C (i.e., $\mathbb{P}_q^{S,M}(C) = \emptyset$), then the minimum length of power acquisition sequences is taken to be ∞ such that the functional degree of q-responsibility of C for S becomes 0. In other cases $\mathcal{FDR}_q^{S,M}(C)$ will be strictly between zero and one. In sequel, we write $\mathcal{FDR}_q^{S}(C)$ and $\mathbb{P}_q^{S}(C)$ instead of $\mathcal{FDR}_q^{S,M}(C)$ and $\mathbb{P}_q^{S,M}(C)$, respectively.

Proposition 3 (Full functionality implies full responsibility). *Let $\hat{C}$ be any coalition, q an arbitrary state and S a given state of affairs. If $\mathcal{FDR}_q^S(\hat{C}) = 1$, then the structural degree of q-responsibility of $\hat{C}$ for S is equal to* 1.

Proof. According to Definition 3, for any non (weakly) q-responsible coalition C, $\mathcal{FDR}_q^S(C) \neq 1$. Hence, based on Proposition 1, for the coalition $\hat{C}$ with functional degree of q-responsibility equal to 1, $\mathcal{SDR}_q^S(\hat{C}) = 1$. □

Note that the other side does not hold in general because $\mathcal{SDR}_q^S(C) = 1$ also includes the cases in which C is a proper super-set of a responsible coalition.

Example 4 (War powers resolution). Consider again the voting scenario in the congress, as explained in Section 2, but now extended with a new *President* agent P. The decision of starting a war W should first be approved by a majority of the congress members (six votes or more in favour of W) after which the President makes the final decision. Hence, P has the preclusive power which is conditioned on the approval of the congress members. Moreover, we have a simplifying assumption that no party member acts independently and thus assume that all members of a party vote in favor of or against the W. In this scenario, which is illustrated in Figure 3, we have an initial state q_s in which all the congress members could use their votes in favour or against the approval of W (no abstention or null vote is allowed). In this example, W will be considered as the state of affairs consisting of states of q_{11}, q_{12}, and q_{13}. This multi-agent scenario can be modelled by the CGS $M = (N, Q, Act, d, o)$, where $N = \{1, ..., 11\}$ (the first ten agents are the voters in the congress followed by the President), $Q = \{q_s, q_0, ..., q_{13}\}$, $Act = \{0, 1, wait\}$, $d_i(q_s) = \{0, 1\}$ for all $i \in \{1, ..., 10\}$, $d_{11}(q_s) = \{wait\}$, $d_i(q) = \{wait\}$ for all $i \in \{1, ..., 10\}$ and $q \in \{q_0, ..., q_{13}\}$, $d_{11}(r) = \{wait\}$ for $r \in (\{q_0, q_1, q_2, q_4, q_6\} \cup \{q_8, ..., q_{13}\})$, and $d_{11}(t) = \{0, 1\}$ for $t \in \{q_3, q_5, q_7\}$. The outcome function o is illustrated in Figure 3 where for example $o(q_s, (1, 0, 0, \star)) = q_4$ in which the war W will not take place because of the disapproval of the congress ($\star$ represents any available action). For notation convenience, actions of party members will be written collectively in the action profiles, e.g., we write $(0, 1, 0, \star)$ to denote the action profile $(0, 0, 1, 1, 1, 0, 0, 0, 0, 0, \star)$. Moreover, the simplifying assumption that all party members vote collectively is implemented by $o(q_s, \bar{\alpha}') = q_s$ for all possible action profiles $\bar{\alpha}'$ in which a party member acts independently.

The set of all (weakly) q_s-responsible coalitions $\mathbb{W}_{q_s}^W$ consists of two coalitions of GD and RD. These two and the coalition GRD, are the coalitions with the preclusive power over W in q_s. If an anti-war campaign wants to negotiate and invest its limited resources in order to avoid the war W, convincing any of coalitions in $\mathbb{W}_{q_s}^W$, as minimal coalitions with power to preclude the war, could avoid the war. However, we can see that convincing the President is also adequate. Although the President has no preclusive power in q_s over W, there exists some accessible states from q_s (i.e., q_3, q_5, and q_7), in which P is responsible for the state of affairs. This potential capacity of P, will be addressed, by means of the introduced notion of *functional degree of responsibility*. Two weakly q_s-responsible coalitions GD and RD, have the functional degree of q_s-responsibility

of 1 for W because they already have sufficient power to avoid W in source state of q_s. Coalitions $\emptyset$, G, R, D, GR and GRD are not (weakly) q_s-responsible for W and no power acquisition sequence exists for these coalitions. Accordingly, their functional degree of q_s-responsibility for W is 0. Coalitions PG, PR, PD, PGR, PGD, PRD and $PGRD$, have the potentiality of possessing the preclusive power in other states, i.e., q_3, q_5, and q_7, but none of them will be minimal coalition with preclusive power over W, as minimality is a requirement for being (weakly) responsible coalition [5]. Hence, the functional degree of q_s-responsibility for all these coalitions will be 0. The coalition which has a chance of becoming a (weakly) responsible coalition in states other than q_s (i.e., q_3, q_5, and q_7) is P. In fact, the President is the (unique) responsible coalition for W in states q_3, q_5, and q_7. As the minimum length of power acquisition sequence for P is 1, the functional degree of q_s-responsibility of P for W is 1/2. Although, P has no independent action profile to avoid W in q_s, there exists a *power acquisition sequence* for P through which P acquires the preclusive power over W.

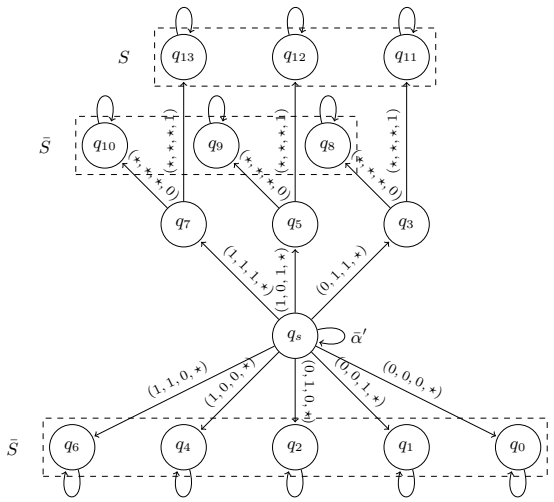


Fig. 3. War powers resolution

The next proposition illustrates that through a shortest power acquisition sequence, the potentiality that the coalition is responsible for the state of affairs, increases strictly. This potential reaches its highest possible value where the coalition "really" has the preclusive power over the state of affairs as a (weakly) responsible coalition. Note that there is a one-to-one correspondence between any power acquisition sequence $P = \langle \bar{\alpha_1}, ..., \bar{\alpha_n} \rangle$ in q for a coalition C for S and the sequence of states $\langle q_1 = q, ..., q_{n+1} \rangle$ due to the deterministic nature of the action profiles $\bar{\alpha_i}$ for $1 \leq i \leq n$, i.e., $o(q_i, \bar{\alpha_i}) = q_{i+1}$ and $q = q_1$ and $q' = q_{n+1}$ and C is (weakly) q'-responsible for S. Hence, in the following, we write $P = \langle q_1, ..., q_{n+1} \rangle$ and interchangeably use it instead of $P = \langle \bar{\alpha_1}, ..., \bar{\alpha_n} \rangle$. Therefore, we simply refer to any state q_i as a state "in" the power acquisition sequence P.

Proposition 4 (Strictly increasing functionality). *Let $P = \langle q_1, ..., q_{n+1}\rangle$ $(n \geq 1)$ be a power acquisition sequence in $q = q_1$ for a coalition C for S. Then, for any tuple of states (q_i, q_{i+1}), $1 \leq i \leq n$, $\mathcal{FDR}^S_{q_i}(C) < \mathcal{FDR}^S_{q_{i+1}}(C)$ iff P is a shortest power acquisition sequence in q for C for S.*

Proof. "⇒": Suppose the claim is false. Then, although the functional degree of responsibility of C for S is strictly increasing from q_1 to q_{n+1} in P, there exists a shorter power acquisition sequence $P' = \langle q'_1, ..., q'_{m+1}\rangle$ $(n > m \geq 0)$ in $q = q'_1$ for C for S. Note that as degrees are strictly increasing, for any states q_a and q_b in P $(q_a \neq q_b)$ we have that $\mathcal{FDR}^S_{q_a}(C) \neq \mathcal{FDR}^S_{q_b}(C)$. Both P and P' end in a state in which C is (weakly) responsible for S. Thus, for states q_{n+1} and q'_{m+1} we have that $\mathcal{FDR}^S_{q_{n+1}\in P}(C) = \mathcal{FDR}^S_{q'_{m+1}\in P'}(C) = 1$. If we trace back step by step through both sequences, the functional degree of responsibility of C for S is equal in corresponding states in P and P'. For example, for the states q_n and q'_m, we have that $\mathcal{FDR}^S_{q_n}(C) = \mathcal{FDR}^S_{q'_m}(C) = 1/2$ $(m \geq 1)$. By continuing the stepwise process of matching all states in P' with corresponding states in P, as number of states in P' is strictly less than P and both sequences start in same state of $q = q_1 = q'_1$, we reach the corresponding states q_{n+1-k} and q'_{m+1-k} for $0 \leq k \leq m$ where $\mathcal{FDR}^S_{q_{n+1-k}}(C) = \mathcal{FDR}^S_{q'_{m+1-k}}(C)$ and $q_{n+1-k} \neq q'_{m+1-k}$ and both states of q_{n+1-k} and q'_{m+1-k} are in P. This contradicts with the assumption that for any states q_a and q_b in P, if $q_a \neq q_b$, we have that $\mathcal{FDR}^S_{q_a}(C) \neq \mathcal{FDR}^S_{q_b}(C)$.

"⇐": Suppose the sequence P is a shortest power acquisition sequence in q for C for S. According to Definitions 1 and 3, the functional degree of q_i-responsibility of C for S must be formulated based on the sequence $P_i = \langle \bar{\alpha_i}, ..., \bar{\alpha_n}\rangle$ as a sub-sequence of P. Accordingly, length of P_i is equal to $\ell_i = n-i+1$. Hence, in each state q_{i+1}, the length of a shortest power acquisition sequence for C for S, ℓ_{i+1}, will be one unit shorter than ℓ_i. Finally, as $\ell \geq 0$, the functional degree of responsibility of C for S in each state q_{i+1} in P is strictly larger than in the state q_i in P. □

The following propositions focus on the cases in which a coalition has partial degrees of functional and structural responsibility in a specific state. In former, we can reason about the degree of responsibility of the coalition in some states other than the current sate while in the latter, we can reason about the degree of responsibility of some other coalitions in the current state.

Proposition 5 (Global signalling of partial functional degree). *Let C be a coalition with functional degree of q-responsibility $1/k$ for S where k is a natural number. Then, it is guaranteed that there exists at least $k-1$ states $\hat{q}$ such that $\mathcal{FDR}^S_{\hat{q}}(C) > \mathcal{FDR}^S_q(C)$ and at least one state q' such that $\mathcal{FDR}^S_{q'}(C)$ $= \mathcal{SDR}^S_{q'}(C) = 1$.*

Proof. According to Proposition 4, the functional degree of responsibility of C for S is strictly increasing during a shortest power acquisition sequence in q for C for S. This sequence passes $k-2$ states and reaches a state q'. Hence, the existence of at least $k-1$ states in which C has functional degree of responsibility

larger than $1/k$ for S, and one state in which the functional and structural degree of responsibility of C is equal to 1 for S is guaranteed. □

Note that based on Definition 3, the functional degree of responsibility could always be written in form of $1/k$ ($k \in \mathbb{N}$) unless it is equal to 0.

Proposition 6 (Local signalling of partial structural degree). *Let C be a coalition with structural degree of q-responsibility of k for S such that $0 < k < 1$. Then, there exists at least a coalition $\hat{C}$ with structural and functional degree of q-responsibility of 1 for S.*

Proof. Based on Definition 2, k is assigned to C based on its contribution in a (weakly) q-responsible coalition which has the structural and functional degree of q-responsibility of 1 for S. □

In general, the existence of a coalition $\hat{C}$ with the structural and the functional degree of q-responsibility of 1, could not guarantee the existence of a coalition with structural degree of q-responsibility of k such that $0 < k < 1$. As explained in Theorem 1, cases in which we have a singleton q-responsible coalition for S are counterexamples for such a claim.

6 Conclusion

In this paper, we proposed a forward-looking approach to measure the degree of group responsibility. The proposed notions can be used as a tool for analysing the potential responsibility of agent coalitions towards a state of affairs. In our approach, full structural and functional degrees of responsibility towards a state of affairs are assigned to agent coalitions, if they can preclude the state of affairs. All other coalitions that may contribute to such responsible coalitions receive a partial structural degree of responsibility. Also, all other coalitions for which there exists a path to a state in which they possesses the preclusive power receives a partial functional degree of responsibility. The structural degree of responsibility captures the responsibility of a coalition based on accumulated preclusive power of included agents, while the functional degree of responsibility captures the responsibility of a coalition due to the potentiality of reaching a state in which it has the preclusive power.

These notions follow the responsibility notions in [5] and are in coherence with the concept of preclusive power in [9]. Our notion of *functional degree of responsibility* of an agent group is based on the minimum length of a sequence from a source state towards a state in which the agent group has power over a given state of affairs. This stepwise formulation was put forward by [1] in a quantified degree of responsibility as a backward-looking approach. However, [1] traces the steps in a causal network and studies the degree of causality, whereas we define our notions in strategic settings by means of a similar formulation. The other connection is to the [4] in which the notion of *avoidance potential* is central. There are two main differences between our approach and [4]. First, our notion of preclusion of a state of affairs is a property of a coalition, whereas in [4] the avoidance potential for a state of affairs is a property of a strategy of an individual agent. Second, the notion of preclusion in our case considers the

power of a coalition while avoidance potential in [4] considers the probability of other agents to choose a strategy such that the strategy of the agent in question has no contribution to the establishment of the state of affairs.

We plan to apply our presented methodology for analysing forward-looking responsibility to backward-looking responsibility. We believe that integrating the responsibility notions as proposed in [1, 4] with our methodology could lead to a graded notion for backward-looking responsibility in strategic settings. In such extension, one could reason from a realized outcome state and assign a degree of blameworthiness to coalitions in liability determination principles from legal domain such as *contributory negligence*. We also plan to relate our notions to existing power indices such as Banzhaf index from cooperative game theory (see [15]). Finally, we aim at extending our framework with logical characterizations of the proposed notions based on the coalitional logic with quantification [16, 5, 17].

References

1. Chockler, H., Halpern, J.Y.: Responsibility and blame: A structural-model approach. J. Artif. Intell. Res.(JAIR) **22** (2004) 93–115
2. Grossi, D., Royakkers, L., Dignum, F.: Organizational structure and responsibility. Artificial Intelligence and Law **15**(3) (2007) 223–249
3. Miller, S.: Collective moral responsibility: An individualist account. Midwest studies in philosophy **30**(1) (2006) 176–193
4. Braham, M., Van Hees, M.: An anatomy of moral responsibility. Mind **121**(483) (2012) 601–634
5. Bulling, N., Dastani, M.: Coalitional responsibility in strategic settings. In: Computational Logic in Multi-Agent Systems. Springer (2013) 172–189
6. Van de Poel, I.: The relation between forward-looking and backward-looking responsibility. In: Moral responsibility. Springer (2011) 37–52
7. Sulzer, J.L.: Attribution of responsibility as a function of the structure, quality, and intensity of the event. PhD thesis (1965)
8. Shaver, K.: The attribution of blame: Causality, responsibility, and blameworthiness. Springer Science & Business Media (2012)
9. Miller, N.R.: Power in game forms. Homo Oeconomicus **16** (1999) 219–243
10. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM (JACM) **49**(5) (2002) 672–713
11. Pauly, M.: Logic for social software. (2001)
12. Kripke, S.A.: Semantical considerations on modal logic. (1963)
13. Turnovec, F.: Monotonicity of power indices. In: Trends in Multicriteria Decision Making. Springer (1998) 199–214
14. Holler, M.J., Napel, S.: Monotonicity of power and power measures. Theory and Decision **56**(1-2) (2004) 93–111
15. Ågotnes, T., van der Hoek, W., Tennenholtz, M., Wooldridge, M.: Power in normative systems. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems (2009) 145–152
16. Ågotnes, T., Van Der Hoek, W., Wooldridge, M.: Quantified coalition logic. Synthese **165**(2) (2008) 269–294
17. Pauly, M.: A modal logic for coalitional power in games. Journal of logic and computation **12**(1) (2002) 149–166