A Programming Approach to Monitoring Communication in an Organisational Environment

(Extended Abstract)

Mehdi Dastani
Utrecht University
The Netherlands
M.M.Dastani@uu.nl

Leendert van der Torre
University of Luxembourg
Luxembourg
leon.vandertorre@uni.lu

Neil Yorke-Smith
American University of Beirut
and SRI International, USA
nysmith@aub.edu.lb

ABSTRACT

Agreement technologies [1] achieve coordination among autonomous computational entities, by combining technologies for norms, semantics, organisations, argumentation, negotiation, and trust. We consider how an organisational programming language, such as 2OPL [2], can be extended to monitor communication. Such an extended programming language can be used to facilitate the development of electronic institutions, organisations, or marketplaces that aim at monitoring agent interaction (including both communication and non-communication actions), checking compliance with norms, and enforcing norms by means of sanctions. This abstract reports on specifying an operational semantics for agent interactions within such a setting, distinguishing constitutive norms for monitoring and sanction rules for enforcement of norms.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Theory

Keywords

Organisation, Environment, Communication, Commitments, Norms

1. INTRODUCTION

It has been established how to define an organisation where agent communication actions create and operate on social commitments (e.g., [5, 3]). Fornara *et al.* [3, 4] propose an Agent Communication Language (ACL) based on communication actions and define norms as "rules that manipulate commitments of the agents engaged in an interaction". Dastani *et al.* [2] specify norms to govern agent interaction; they provide rules to specify norms and sanctions, but do not focus on communication actions.

This abstract reports on extending the approach of [2] with communication actions by following the successful line of work using commitments. The extension is based on the assumption that communication should respect a set of generic norms which are inherent in communication actions.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1.1 Research questions

The following questions guide the design of a programming language (and its operational semantics) that facilitates the implementation of norm-based organisations:

1. How to define an operational basis for both communication and non-communication actions in norm-based organisations?
2. How to uniformly handle monitoring and enforcing of norms for both types of actions?
3. How to develop a full operational semantics with desirable properties—without focusing on the protocol or semantics concerns of a full ACL—by adopting a simple (but extendible) set of communication actions?

Compared to the foundational work of Singh and Colombetti and colleagues, our approach differs in that: (1) We aim to use *counts-as rules* explicitly as technical constructs while Fornara *et al.* treat counts-as relation primarily as linguistic conventions. (2) We want to provide an operational semantics for interactions (among which communication actions) within an organisational setting and analyze the properties of interaction. Fornara *et al.* provide semi-formal specification of organisations and consider only communication actions. (3) We aim to consider the effect of non-communicative actions as further the elapse of deadlines. (4) We want to allow for sanction rules whereas earlier works leave open the question of what should happens when norms or commitments are violated. We adopt a standard, contemporary lifecycle of commitments, following [6].

2. OVERVIEW OF THE APPROACH

An organisation monitors the agents' interactions (both communication and non-communication actions), determines the (state of) commitments and the violated norms, and ensures that the agents fulfil their commitments and norms—or otherwise imposes sanctions such as putting the violating agent on a blacklist. In our approach, the organisation, as an exogenous process, cannot *intervene* in the decision making of individual agents; in this setting agents are assumed to be autonomous in the sense that they decide their own actions. In the following subsections, we indicate how such organisations can be programmed.

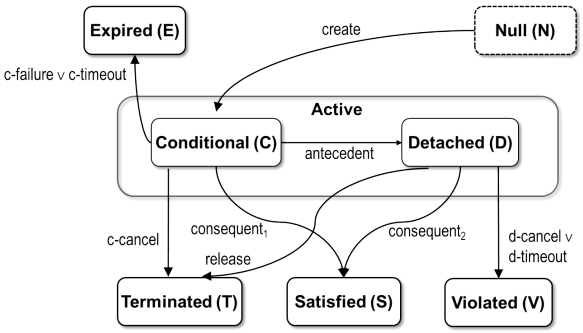

Figure 1: State transitions of commitment lifecycle.

2.1 Commitments

Social commitments represent a popular means of capturing relations between two agents with deontic force; they provide the basis for an ACL within an organisational setting [5, 3]. Formally, a *commitment* [6] is defined as a tuple $\mathsf{C}(x, y, p, q, d_1, d_2)$, which can be read as "Agent x (as debtor) commits to agent y (as creditor) that if proposition p (the antecedent) is brought about by deadline d_1 then x will bring about q (the consequent) by deadline d_2".

Fig. 1 shows the states of a simplified lifecycle of a commitment, adapted from [6] (we omit suspension and delegation). Boxes indicate states and arrows transitions. We write commitment state with superscript, i.e., $\mathsf{C}^{\text{state}}$.

2.2 Agent interactions

Possible actions that agents perform to interact with each other or with their shared environment include pure communication actions (e.g., promise to pay), and non-communicative actions that change the actual state of their environment (e.g., make a payment). Our purpose is not to define an ACL or a communication protocol; instead, we select a representative set of actions influencing the generation and state of commitments. We take the following set to demonstrate our methodology for programming an organisational model and the management and enforcement of commitment-based norms. We use variables $x,\ y, \ldots$ to range over the agent names $i,\ j, \ldots$; propositional variables $p,\ q, \ldots$ to range over propositions $a,\ b, \ldots$; and finally $d,\ d', \ldots$ to range over deadlines $t_m,\ t_n, \ldots$, where $m, n \in \mathbb{N}$.

- *offer*(x, y, p, q, d_1, d_2) — x tells y that x will make q true in the environment by deadline d_2 if p becomes true in the environment by deadline d_1
- *tell*(x, y, p) — x tells y that p is true in the environment
- *cancel*(x, y, q) — x tells y that x will not make q true
- *release*(y, x, q) — y tells x that x needs not make q true
- *failure*(x, y, p) — x tells y that p cannot be made true in the environment
- *do*(x, p) — x performs an action to make proposition p true in the environment

We assume here that agents are trusted, i.e., their utterances are according to their beliefs. Note that an organisation may develop a list of trusted agents.

2.3 Organisation

An *organisation* is specified by facts, norms, and sanctions. Norms are states that an organisation aims at enforcing and can therefore be seen as the goals of the organisation. We distinguish *brute* and *institutional* facts. Brute facts denote the state of the shared environment (e.g., b_j denoting the fact that agent j has book b or $p_{(b,20)}$ denoting the fact that 20 euro is paid for book b), while institutional facts denote the normative state of an organisation (e.g., $\mathsf{C}^D(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5)$ denoting the fact that agent i is committed to pay 20 euro before t_5 if agent j sends book b before t_2, or $viol_{reg-b}$ denoting the fact that agent b has violated the registration norm).

We follow [2] and represent norms by means of the *counts-as* construct. The original version of the counts-as construct is of the form "ϕ counts as ψ in the context c". We program the monitoring component of an organisation by constructs of the form $\phi \wedge c \Longrightarrow_{cr} \psi$, where $\phi \wedge c$ can be either brute or institutional facts and ψ is an institutional fact. For example, the counts-as rule $\mathit{offer}(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5) \Longrightarrow_{cr} \mathsf{C}^C(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5)$ implements a norm that an offer by agent i to agent j to do a payment if j sends i a book *counts-as* a conditional commitment. Finally, we program sanctions by rules of the form $\phi \Longrightarrow_{sr} \psi$, where ϕ can comprise brute and institutional facts and ψ is a brute fact. In our running example, $\mathsf{C}^C(i, j, s_{(b,i)}, p_{(b,20)}, t_2, t_5) \Longrightarrow_{sr} \mathit{blacklist}(i)$ implements the sanctions to add agent i to a blacklist if agent i payment is not fulfilled after day 5. Following the above, an *organisation* is programmed by the tuple (F, cr, sr), where F is a set of initial brute facts, cr is a set of counts-as rules, and sr is a set of sanction rules. The institutional facts are generated during run-time.

2.4 Discussion

The details of the operational semantics consist of transition rules specifying how agent actions (communication and non-communication) create and modify institutional facts, including commitments, how the organisation determines norm violations by applying counts-as rules, and how the organisation respond to norm violations by applying sanction rules. Our ongoing work is to explore the properties of normative multi-agent system executions and to apply the programming approach to realistic scenarios.

3. REFERENCES

[1] Billhardt, H. et al. Organisational structures in next-generation distributed systems. *Multiagent and Grid Systems*, 7(2–3):109–125, 2011.

[2] M. Dastani, D. Grossi, J.-J. C. Meyer, and N. Tinnemeier. Normative multi-agent programs and their logics. In *Proc. Workshop on Knowledge Representation for Agents and Multi-Agent Systems*, LNCS 5605, pages 16–31, 2009.

[3] N. Fornara and M. Colombetti. A commitment-based approach to agent communication. *Applied Artificial Intelligence*, 18(9–10):853–866, 2004.

[4] N. Fornara, F. Viganò, and M. Colombetti. Agent communication and artificial institutions. *Autonomous Agents and Multi-agent Systems*, 14:121–142, 2007.

[5] A. U. Mallya, P. Yolum, and M. P. Singh. Resolving commitments among autonomous agents. In *Proc. Workshop on Agent Comm.*, pages 166–182, 2003.

[6] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models. *IEEE Trans. Services Computing*, PrePrint, 2011.