# NORM-ORIENTED PROGRAMMING OF ELECTRONIC INSTITUTIONS[1]

(Extended Abstract)


Andrés García Camino [a] Juan Antonio Rodríguez Aguilar [a]

Carles Sierra [a] Wamberto Vasconcelos [b]

[a] *IIIA-CSIC, Campus UAB, 08913 Bellaterra, Spain*

[b] *Dept. of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom*



**Abstract**

Norms constitute a powerful coordination mechanism among heterogeneous agents. We propose means to specify and explicitly manage the normative positions of agents (permissions, prohibitions and obligations), with which distinct deontic notions and their relationships can be captured. Our rule-based formalism includes constraints for more expressiveness and precision and allows the norm-oriented programming of electronic institutions: normative aspects are given a precise computational interpretation. Our formalism has been conceived as a machine language to which other higher-level normative languages can be mapped, allowing their execution.


A major challenge in multi-agent system (MAS) research is the design and implementation of *open* multi-agent systems in which coordination must be achieved among agents defined with different languages by several designers who may not trust each other. Norms can be used for this purpose as a means to regulate the observable behaviour of agents as they interact in pursuit of their goals [6]. There is a wealth of socio-philosophical and logic-theoretical literature on the subject of norms (*e.g.*, [5]), and, more recently, much attention is being paid to more pragmatic and implementational aspects of norms, that is, how norms can be given a computational interpretation and how norms can be factored in in the design and execution of MASs (e.g. [1, 3]).

A normative position is the "social burden" associated with individual agents, that is, their obligations, permissions and prohibitions. Depending on what agents do, their normative positions may change – for instance, permissions/prohibitions can be revoked or obligations, once fulfilled, may be removed. Ideally, norms, once captured via some suitable formalism, should be directly executed, thus realising a computational, normative environment wherein agents interact. This is what we mean by *norm-oriented programming*. We try to make headway along this direction by introducing an executable language to specify agents' *normative positions* and manage their changes as agents interact via speech acts [4].

In this paper we present a language that acts as a "machine language" for norms on top of which different, higher-level normative languages can be accommodated. This language can represent distinct flavours of deontic notions and relationships. Although our language is rule-based, we achieve greater flexibility, expressiveness and precision than production systems by allowing constraints to be part of our rules and states of affairs. In this way, normative positions can be further refined. For instance, picture a selling agent that is obliged to deliver a good satisfying some quality requirements before a deadline. Notice that both the quality requirements and the delivery deadline can be regarded as constraints that must be considered as part of the obligations. Thus, when the agent delivers the good satisfying all the constraints, we should regard the obligation as fulfilled. Notice too that since the deadline might eventually be changed, we also require the capability of

[1] Previously published as [3]

modifying constraints at run-time. Hence, constraints are considered as first-class citizens in our language.

Although in this paper we restrict to a particular class of MASs, namely electronic institutions [2], our work sets the foundations to specify and implement open regulated MASs via norms.

Our main goal is to produce a language that supports the specification of coordination mechanisms in multi-agent systems by means of norms. For this purpose, we identify below the desirable features we expect in candidate languages.

**Explicit management of normative positions** We take the stance that we cannot refer to agents' mentalistic notions, but only to their observable actions and their normative positions. Notice that as a result of agents' observable, social interactions, their normative positions change. Hence, the first requirement of our language is to support the *explicit management* of agents' normative positions.

**General purpose** Turning our attention to theoretical models of norms, we notice that there is a plethora of deontic logics with different axioms to establish relationships among deontic notions. Thus, we require that our language captures different deontic notions along with their relationships. In other words, the language must be of *general purpose* so that it helps MAS designers to encode any axiomatisation, and thus specify the widest range of normative systems possible.

**Pragmatic** In a sense, we pursue a "machine language" for norms on top of which higher-level languages may be accommodated. Along this direction, and from a language designer's point of view, it is fundamental to identify the *norm patterns* (*e.g.*, conditional obligations, time-based permissions and prohibitions, continuous obligations, and so on) in the literature to ensure that the language supports their encoding. In this way, not only shall we be guaranteeing the expressiveness of our language, but also addressing pragmatic concerns by providing *design patterns* to guide and ease MAS design.

**Declarative** In order to ease MAS programming, we shall also require our language to be *declarative*, with an implicit execution mechanism to reduce the number of issues designers ought to concentrate on. As an additional benefit, we expect its declarative nature to facilitate verification of properties of the specifications.

**Acknowledgements** – This work was partially funded by the Spanish Education and Science Ministry and Spanish Council for Scientific Research (CSIC) as part of the projects TIN2006-15662-C02-01 and 2006-5-0I-099. García-Camino enjoys an I3P grant from CSIC.

## References

[1] Alexander Artikis, Lloyd Kamara, Jeremy Pitt, and Marek Sergot. A Protocol for Resource Sharing in Norm-Governed Ad Hoc Networks. In *Declarative Agent Languages and Technologies II*, volume 3476 of *LNCS*. Springer-Verlag, 2005.

[2] Marc Esteva. *Electronic Institutions: from specification to development.* PhD thesis, Universitat Politecnica de Catalunya, 2003. Number 19 in IIIA Monograph Series.

[3] Andrés García-Camino, Juan-Antonio Rodríguez-Aguilar, Carles Sierra, and Wamberto Vasconcelos. Norm-Oriented Programming of Electronic Institutions: A Rule-based Approach. In *Coordination, Organization, Institutions and Norms in agent systems (COIN'06) in Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. (AAMAS'06)*, May 2006.

[4] John Searle. *Speech Acts, An Essay in the Philosophy of Language.* Cambridge University Press, 1969.

[5] Yoav Shoham and Moshe Tennenholtz. On Social Laws for Artificial Agent Societies: Off-line Design. *Artificial Intelligence*, 73(1-2):231–252, 1995.

[6] Michael Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons, Chichester, UK, February 2002.