From Human Regulations to Regulated Software Agents' Behaviour

Javier Vázquez-Salceda

Knowledge Engineering and Machine Learning Group
Technical University of Catalonia, Barcelona, Spain
jvazquez@lsi.upc.edu

1 Introduction

Internet, as an extension of the real world, is affected by the regulations of one or several countries on activities carried out through the web. For instance, Electronic Commerce activities between two parties are regulated by the law of the parties' countries plus international commerce treaties. In the HealthCare field, highly regulated, the citizens' rights are precisely defined and regulated by national and international laws. How to make sure that such norms and regulations are met on the activities and information exchanges through Internet? How to create mechanisms to enforce norm compliance, and therefore, increase trust between individuals and companies?

In most software and agent methodologies, these external regulations, along with the internal norms and regulations of the organization to be modelled, are seen only as extra requirements in the analysis phase of the system. If either the external or the internal regulations change (as they usually do from time to time), it becomes very hard to track all the changes to be done in the implementation, as there is no explicit representation of the norms and regulations, but a chain of design decisions that were guided by the norms' requirements (i.e., if norms are embedded in the agents' design and code, all the design steps have to be checked again and all the code verified to ensure compliance with new regulations). The alternative is to have an explicit representation of the norms.

Research on Distributed Artificial Intelligence has created the concept of Electronic Institutions. As their human counterparts, an Electronic Institution is an entity defining a set of norms over the behaviour of individuals inside the institution. Last years research in electronic institutions has focused on the use of Software Agent technology. An Agent-Mediated Electronic Institution (e-institution for short) belongs to a new and promising field where interactions between a group of (software) agents are regulated by means of a corpora of explicit norms, expressed in a computational language that agents can interpret. An e-institution [6] [7] is a safe environment mediating in the interaction of agents. The expected behaviour of agents in such an environment is described by means of an explicit specification of norms, which is a) expressive enough, b) readable by agents, and c) easy to maintain.

1.1 Problem: Abstractness of Human regulations

Human regulations are usually written in a quite abstract language and are open to interpretation. The main reason for this is to cover with the same legal text the major number of cases and therefore be stable for longer periods of time. For instance, Spanish regulations on Human Organ donation and transplantation state:

- *"a living donor should consent to the donation of an organ"*. In this case *"to consent"* is an abstract action not properly defined (on purpose), so the number of accepted ways to

give "consent" can be extended (e.g., to include electronic signed documents) without changing the law.
- *"the National Transplant Organization should provide for an appropriate distribution of organs"*. The term *"appropriate"* is vague (has no precise meaning) and open to the interpretation of lawyers and policy makers.

This abstraction and capability of multiple interpretations that are positive for humans pose a problem when trying to implement them on computers, where meanings should be precise and unambiguous. The main problem when designers try to include the norms in the design process of an e-Institution is that the level on which the norms are specified in human regulations is more abstract and/or general than the level on which the processes and structure of the organization are specified. Therefore we need to *translate* the norms specifically to a operational description where the impact of the norms on the behaviour of the organization can be described and modelled.

Currently in the agent systems field, there are two approaches to *Normative Systems*: those that represent norms explicitly at a very low, operational level (policies and procedures) and those that represent norms at a very high, declarative, abstract level, formally specifying norms in, e.g., deontic logic. The low level approaches allow an easy implementation as they are already *operational* in nature, but the problem arises when the correctness of the procedures and policies should be checked against the original regulations. Because of this problem, it is hard to design organizations that fully meet very complex normative specifications, and almost impossible to mantain the implemented systems if there are changes in the regulations. High level approaches are closer to the way regulations are made, so verification is easier to be done. Such approaches work on normative systems' formalization which are declarative in nature, and focus on the expressiveness of the norms, the definition of formal semantics and the verification of consistency of a given set. However, high level approaches usually use one or several computationally hard logics like deontic logic ([8, 12]). Although it is possible to capture the norms in this way and even give them a certain kind of semantics to reason about the consequences of the norms, this kind of formalization does not yet indicate how the norm should be interpreted within a certain organization. For instance, we can formalize a norm in the Human Organ Transplant domain like *"it is forbidden to discriminate on the basis of age"* (when determining the best possible recipient for an organ) in deontic logic as $F(\mathit{discriminate}(x,y,\mathit{age}))$ (stating that it is forbidden to discriminate between x and y on the basis of age). However, the semantics of this formula will get down to something like that the action `discriminate(x,y,age)` should not occur. It is very unlikely that the agents operating within the organization will explicitly have such an action available.

In summary, to apply norms in software agents setups, not only the *declarative* aspects of norms are important, but also an *operational* meaning.

2 Programme objectives

We aim to find a connection between high-level, very expressive norm specifications and low-level e-institution implementations to be able to specify, design and implement organizations by means of agent-mediated e-institutions in highly-regulated domains. We have four main research lines:

1. to define a formal language to specify norms for agents. This language should be expressive enough for complex, highly regulated scenarios and machine-parseable.
2. to formally connect a specification of a set of norms (in the above-mentioned language) with an operational specification of the accepted behaviour inside an e-Institution. This connection is important for *traceability* both in the *top-down* and *bottom up* directions:

- *top-down*: the connection from norms to the final procedures guides the design process of the *e*-Institution, as it identifies the minimum set of restrictions that are defined by the normative framework, and eases the task of checking if the implemented procedures follow such restrictions. It also enables maintenance of the implemented system when regulations change, as the designer can trace down, for each norm to be changed, its effects in the operational specification and implementation.
- *bottom-up*: agents can trace the origin of a given protocol or procedure and reason in terms of the norms the protocol/procedure implements. This allows the definition of *Normative Agents*, which are able to handle those unexpected situations that a given protocol has not considered, by reasoning in terms of the related norms and adapting their behaviour appropriately.

3. to extend e-Institutions platforms (such as AMELI) with the enforcement mechanisms needed to check compliance of norms by the agents interacting in the platform.
4. to refine the OMNI framework to create a methodology and the tools to support designers in the specification, analysis, design and implementation of e-Institutions for highly-regulated environments.

3 Assumptions

1. Norms can sometimes be violated by agents in order to keep their autonomy, which can also be functional for the system as a whole as argued in [3]. The violation of norms is handled from the organizational point of view by violation and sanction mechanisms.
2. From the institutional perspective the internal state of the external agents is neither observable nor controlable (external agents as black boxes). Therefore, we cannot avoid a forbidden action to be in the goals and intentions of an agent, or impose an obligatory action on an agent to be in their intentions.
3. Implementing norms is not implementing a theorem prover that, using the norms semantics, checks whether a given interaction protocol complies with the norms. The implementation of norms should consider a) how the agents' behaviour is affected by norms, and b) how the institution should ensure the compliance with norms. The former is related to the *implementation of norms from the agent perspective*, by analyzing the impact of norms in the agents' reasoning cycle (work on this perspective can be found in [1] [2] [4]). The latter is related with the *implementation of norms from the institutional perspective*, by implementing a safe environment (including the enforcing mechanisms) to ensure trust among parties.
4. In the analysis and design phases of an e-institution, developers should cover 3 dimensions: a) the **Normative Dimension** of the organization, which specifies the mechanisms of social order, in terms of common norms and rules, that members are expected to adhere to; b) the **Organizational Dimension** of the organization, which describes the structure of an organization in terms of roles and interaction structures; and c) the **Ontological Dimension**, which defines environment and contextual relations and communication aspects in organizations. In those domains with none or small normative components, design is guided by the Organizational Dimension, while in highly regulated domains the Normative Dimension is more prominent and therefore guides the design.

4 Approach

1. **Type of norms**: In the legal domain, norms are descriptions of how a person (or agent) should behave in order to comply with legal standards. If we take a look at human regulations, we can observe three main types of norms:

- **Norms defining (refining) the meaning of abstract terms** (e.g. *"The criminal register administrator can be the Regional Police Force Commander, the Dutch National Police Force commander, the Royal Military Police Commander, the College of the Procurator-General or an official appointed by The Minister of Justice"*).
- **Norms defining (refining) an abstract action by means of sub-actions (a plan) or procedures** (e.g. *"A request for examination [of personal data] [...] is sustainable after receipt of the payment of EUR 4,50 on account [...] of the force mentioning 'privacy request'"*)
- **Norms defining obligations/permissions/prohibitions**. These can be subdivided in restrictive norms (norms permitting/forbidding actions or situations) or impositive (norms forcing an entity to do an action or to reach a state)

The first and second type of norms are important in order to define the vocabulary to be used in a given regulation, but pose no real problems for its implementation. In an agent-mediated system, these norms would be implemented in the ontology of the system and/or in the refinement process of the actions on the system. The last type of norms are the ones that define the acceptable behaviour of entities, and are the ones that pose problems. These are the type of norms we are focused on.

2. **Norm language** In order to express complex norms we use a language consisting of deontic concepts (OBLIGED, PERMITTED, FORBIDDEN) which can be conditional (IF) and can include temporal operators (BEFORE, AFTER) to be used with deadlines. Deadlines can be either *absolute* (e.g. 23:59:00 09/05/2004) or *relative* (e.g. $time(done(bid)) + 5min$). some examples of norms in this language are:

$$\begin{array}{l}\mathsf{OBLIGED}((buyer\ \mathsf{DO}\ bid(product, price))\\ \ \mathsf{BEFORE}\ (buyer\ \mathsf{DO}\ exit(auction\_house)))\end{array}$$

$$\begin{array}{l}\mathsf{PERMITTED}((user\ \mathsf{DO}\ appoint(regular\_user))\\ \ \mathsf{IF}\ (access\_level(user, register, `full\_control')))\end{array}$$

   More details about this language can be found in [10].
3. **Violations, sanctions and repairs**: In order to implement enforcement mechanisms that are well-found, one has to define some kind of operational semantics first. In general, an operational semantics for norms always comes down to either one of the following: a)defining constraints on unwanted behaviour, or b) detecting violations and reacting to these violations. The choice between these two approaches is highly dependent on the amount of control over the addressee of the norms. Prevention of unwanted behaviour can only be achieved if there is full control over the addressee; otherwise, one should define and handle violations. As one of our assumptions is that e-Institutions do not have full control over the agents and, therefore, there may be illegal actions and states which are outside the control of the enforcer, violations should be included in the normative framework. In order to manage violations, each violation should include a plan of action composed by a sanctioning mechanism (*sanctions*) and countermeasures to return the system to an acceptable state (*repairs*).
4. **Norm expressions and enforcement**: the implementation of enforcement is composed of three related processes a) the detection of when a norm is active, b) the detection of a violation on a norm, and c) the handling of the violations. An agent platform should include the detection mechanisms to ease norm enforcement, specially for those checks that are computationally hard and may overload the agents in the e-institution that have to enforce the norms.

5 Past, Present and Future

The roots of this research come from the HARMON*IA* framework, first presented by the author in his PhD thesis [9]. The HARMON*IA* framework only covered the normative di-

mension of e-Institutions and proposed a way to connect the high- and low-levels. With the collaboration of the Intelligent Systems group at Utrecht University, the OMNI framework has been created. OMNI [11] is an integrated framework for modelling a whole range of MAS, from closed systems with fixed participants and interaction protocols, to open, flexible systems that allow and adapt to the participation of heterogeneous agents with different agendas. This approach is rather unique, as most existing frameworks concentrate in a specific type of MAS. OMNI is composed by three dimensions: **Normative**, **Organizational** and **Ontological** that describe different characterizations of the environment. The model is a fusion of the OperA framework [5], and HARMON*IA* .

In parallel, in [10] some fundamental research has been done on norm modeling (by proposing a first language for norms thata is parseable by agents) and on implementation issues of norms (by proposing enforcement mechanisms and how to connect the enforcement mechanisms with the norms).

Currently there is a collaboration with the IIIA group to extend EIDE by introducing our norm model into ISLANDER, and adding enforcement mechanisms to the AMELI run-time platform.

Acknowledgements

This position paper compiles results of work done in collaboration with Frank Dignum, Virginia Dignum, John-Jules Ch. Meyer, Davide Grossi and Huib Aldewereld. It also includes ideas coming from valuable discussions with Ulises Cortés, Julian Padget and Owen Cliffe.

References

1. G. Boella and L. van der Torre. Fulfilling or violating norms in normative multiagent systems. In *Proceedings of IAT 2004*. IEEE, 2004.
2. G. Boella and L. van der Torre. Normative multiagent systems. In *Proceedings of Trust in Agent Societies Workshop at AAMAS'04*, New York, 2004.
3. C. Castelfranchi. Formalizing the informal?: Dynamic social order, bottom-up social control, and spontaneous normative relations. *Journal of Applied Logic*, 1(1-2):47–92, February 2003.
4. C. Castelfranchi, F. Dignum, C. Jonker, and J. Treur. Deliberative normative agents: Principles and architectures. In N. Jennings and Y. Lesperance, editors, *ATAL '99*, volume 1757 of *LNAI*, pages 364–378, Berlin Heidelberg, 2000. Springer Verlag.
5. V. Dignum. *A Model for Organizational Interaction: based on Agents, founded in Logic*. SIKS Dissertation Series 2004-1. SIKS, 2004. PhD Thesis.
6. V. Dignum and F. Dignum. Modelling agent societies: Coordination frameworks and institutions. In P. Brazdil and A. Jorge, editors, *Progress in Artificial Intelligence*, LNAI 2258, pages 191–204. Springer-Verlag, 2001.
7. M. Esteva, J. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J.-J.Ch. Meyer and M. Tambe, editors, *Intelligent Agents VIII*, volume 2333 of *LNAI*, pages 348–366. Springer Verlag, 2001.
8. J.-J. Ch. Meyer and R.J. Wieringa. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley and sons, 1991.
9. J. Vázquez-Salceda. *The Role of Norms and Electronic Institutions in Multi-Agent Systems. The HARMONIA framework*. Whitestein Series in Software Agent Technology. Birkhäuser Verlag, 2004.
10. J. Vázquez-Salceda, H. Aldewereld, and F. Dignum. Implementing norms in multiagent systems. In G. Lindemann, J. Denzinger, I.J. Timm, and R. Unland, editors, *Multiagent System Technologies*, LNAI 3187, pages 313–327. Springer-Verlag, 2004.
11. J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. Technical report, Institute of Information and Computing Sciences, Utrecht University, 2004.
12. G.H. von Wright. On the logic of norms and actions. *New Studies in Deontic Logic*, pages 3–35, 1981.