Graphically Explaining Norms

(Extended Abstract)

Madalina Croitoru

LIRMM
CNRS and Univ. Montpellier II
France
croitoru@lirmm.fr

Nir Oren Simon Miles Michael Luck

King's College London
Dept. of Computer Science
London, UK
nir.oren, simon.miles,
michael.luck@kcl.ac.uk

ABSTRACT

While much work has focused on the creation of norm aware agents, much less has been concerned with aiding a system's designers in understanding the effects of norms on a system. However, since norms are generally pre-determined by designers, providing such support can be critical in enabling norm refinement for more effective or efficient system regulation. In this paper, we address just this problem by providing *explanations* as to why some norm is applicable, violated, or in some other state. We make use of conceptual graph based semantics to provide an easily interpretable graphical representation of the norms within a system. Such a representation allows for visual explanation of the state of norms, showing for example *why* they may have been activated or violated. These explanations then enables easy understanding of the system operation without needing to follow the system's underlying logic.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Design, Languages

Keywords

Norms, Conceptual Graphs

1. INTRODUCTION

Norm-aware agents make use of concepts such as obligations, permissions, and prohibitions to represent and reason about socially imposed goals and capabilities. Such agents are able to decide whether to act in a manner consistent with norms, or whether to ignore them. Norms typically increase the overall utility of a system at the cost of individual utility [4].

While a norm-aware agent can reason about the norms that are applicable to it, the problem of *explaining* why a norm is applicable or violated, or associated with some other similar *status*, has not been investigated in depth. The ability to provide such an explanation has multiple benefits. For example, a designer would be better able to understand the interactions between different norms, allowing them to avoid creating redundant norms, and to specify norms more precisely. A user would be able to get a more intuitive understanding of a system by understanding the reason why certain norms were instantiated, violated, or expired in response to system events.

Our goal in this paper is to provide a graphical explanation of norms, based on conceptual graphs[1]. A graphical representation is more easily understandable to a non-expert, and the conceptual graph approach allows us to assign a logical semantics to our representation. This allows us to reason over the graph structure, and operations over norms can be performed over the graphs.

Cite as: Graphically Explaining Norms (Extended Abstract), M. Croitoru, N. Oren, S. Miles, M. Luck, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1405-1406

2. NORM REPRESENTATION

Norms are typically specified within some knowledge based system using a logic which, for non-technical users, is often difficult to understand. For example, [2] represented a norm using a 5-tuple of the form

$$\langle NormType, NormActivation, NormCondition, NormExpiration, NormTarget\rangle$$

where $NormType$ states whether the norm is an obligation or permission, and the remaining parameters are logical formulae identifying when a norm comes into force (*NormActivation*), whether it is violated once it is in force (*NormCondition*), when it ceases to be in force (*NormExpiration*), and which agents are affected by the norm (*NormTarget*). A norm that is not in force is referred to as *abstract*, and has unground variables in its *NormActivation* parameter, while an *instantiated norm* has a ground activation condition, and may be complied with or violated. The process of *instantiating* a norm generates an instantiated norm from an abstract norm, and binds its variables to specific constant values. Thus, for example, the following abstract norm represents the idea that a repair shop must repair a car within seven days of its arrival at the shop:

$$\begin{aligned}&\langle obligation,\\ &arrivesAtRepairShop(X, Car, T_1),\\ &repaired(Car) \vee (currentTime(CurrentTime) \wedge\\ &before(CurrentTime, T_1 + 7days)),\\ &repaired(Car),\\ &repairShop(X)\rangle\end{aligned}$$

An instantiated version of this norm would have constants substituted for X, Car and T_1.

3. CONCEPTUAL GRAPH REPRESENTATION

A basic graph (BG) is a bipartite graph containing two classes of nodes. The first class, called *concept* nodes, represents entities while the second, called *relation* nodes, represents entity properties as well as inter-entity relationships. Such nodes and relations are organised in a vocabulary that is thus composed of two partially ordered sets: a set of concepts and a set of relations of any arity (where the arity identifies the number of arguments of the relation). The partial order represents the *specialisation* relation. A conceptual graph is composed of a basic graph together with the graph's vocabulary. These graphical objects are provided with a semantics in first order logic, defined by a mapping classically denoted by Φ in the conceptual graphs literature [5]. The fundamental theorem states that given two BGs G and H, there is a homomorphism from G to H if and only if $\Phi(G)$ is a semantic consequence of $\Phi(H)$ and the logical translation of the vocabulary, i.e. $\Phi(\mathcal{V}), \Phi(H) \models \Phi(G)$.

We represent norms by means of a *norm tree*, whose nodes contain individual conceptual graphs. The tree's root represents the entire norm, the nodes in the second level of the norm tree are associated with the activation condition, the nodes in the third level are associated with the normative condition, and in the forth level with the expiration condition. Figure 1 depicts a norm tree. Each of the nodes in the norm tree has associated a Conceptual Graph representation of their content. Edges in the norm tree allow one to identify the subgraphs which affect the norm's status as the system executes.

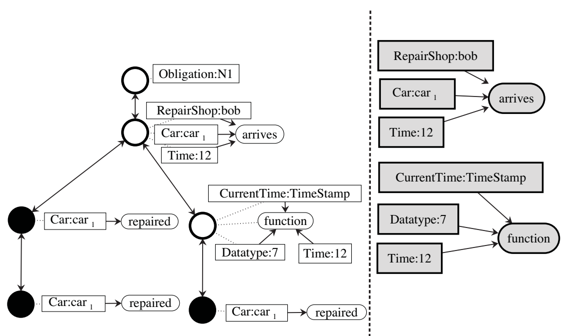

Figure 1: A norm tree (on the left) whose nodes are evaluated according to the knowledge base shown on the right.

Since CGs cannot easily represent disjunctive formulae, we represent the norm using the disjunctive normal form of its elements, i.e. a norm n_1 is written as

$$\langle Type, \bigvee_{i=1,a} AC_i, \bigvee_{j=1,c} NC_j, \bigvee_{k=1,e} EC_k, NT \rangle$$

Where $Type$, AC_i,NC_j, EC_k and NT are all conjunctive positive existential first order logic formulae. Each such formula is then associated with a node in the tree, as illustrated by Figure 1. Thus, for example, the CGs associated with the nodes in the normative condition level, i.e. the third level of the graph, state that a car should either be repaired or that 12 time units have not yet passed.

4. EXPLAINING NORMS

Very little work appears to deal with the explanation of norms, instead assuming that systems are fully automated (thus requiring no explanation), or that the user is able to understand the norm's representation. Even in the latter case, a more intuitive, graphical explanation may be advantageous when trying to reason about complex interactions between large groups of norms.

By shading nodes in the graphical norm representation according to their truth value, and allowing a user to then visualise the associated CGs in a similar manner, it is possible to explain *why* a norm is violated, instantiated, or expired. Norms in the real world may have a complex internal structure (as seen by the model proposed above), and not be easily accessible to a non-expert. The graphical model proposed here allows a non-technical user to drill down into the norm and obtain an explanation regarding its status. We can envisage applications of our model not only in reasoning about norms, but also in monitoring norms. The latter strand of work is particularly applicable in the domain of automated contracting [2]. The ability to provide explanations for a norm's status in such domains is particularly useful. For example, a complex contract dispute may require that some rewards or penalties be assigned by a human mediator. However, in order to understand what rewards or penalties should be assigned, the mediator must first understand which norms were violated, and which were complied with. Norm explanation is also important at the system design stage. A designer may model the system, but must then understand what norms could be violated at different points in time.

5. CONCLUSIONS

In this paper we described how a rich model for tracking and determining the status norms may be represented graphically. As a norm's status changes, so does its graphical representation. This allows the normative system to be understood visually.

Work such as [3] has attempted to explain the causes of a norm violation by making use of a causal graph. This is then fed into a policy engine which attempts to determine whether there are mitigating circumstances for the violation. If so, then enforcement of penalties against the violator can be ignored, or reduced.

While other studies have shown that graphical representations are more easily understood by non-experts than logic-based ones [1], we have not yet evaluated our model in this way, but intend to do so in the short term. We also intend to leverage the formal power of our model, by investigating the use of graph theoretical operations to identify redundant norms, and identify and resolve normative conflict.

6. REFERENCES

[1] M. Chein and M. Mugnier. *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer, 2009.

[2] N. Faci, S. Modgil, N. Oren, F. Meneguzzi, S. Miles, and M. Luck. Towards a monitoring framework for agent-based contract systems. In *CIA '08: Proceedings of the 12th international workshop on Cooperative Information Agents XII*, pages 292–305, 2008. Springer

[3] S. Miles, P. Groth, and M. Luck. Handling mitigating circumstances for electronic contracts. In *AISB 2008 Symposium on Behaviour Regulation in Multi-agent Systems*, pages 37–42, 2008.

[4] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1–2):231–252, 1995.

[5] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.