

## Graphically explaining norms

Madalina Croitoru, Oren Nir, Miles Simon, Luck Michael

► **To cite this version:**

Madalina Croitoru, Oren Nir, Miles Simon, Luck Michael. Graphically explaining norms. Wiebe van der Hoek; Gal A. Kaminka; Yves Lespérance; Michael Luck; Sandip Sen. AAMAS: Autonomous Agents and Multiagent Systems, 2010, Toronto, Canada. IFAAMAS 2010, 9th International Conference on Autonomous Agents and Multiagent Systems, 1, pp.1405-1406, 2010, AAMAS'10: 9th International Conference on Autonomous Agents and Multiagent Systems. lirmm-00538391

**HAL Id: lirmm-00538391**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00538391>**

Submitted on 22 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 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 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. Such an explanation 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 is able to reason about what norms are applicable to it, or some other agent in the system given some situation, the problem of *explaining* why a norm is applicable, or violated, or in some other similar state, has not been investigated in depth. The ability to provide such an explanation has multiple

**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. XXX-XXX.

Copyright © 2010, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

benefits. For example, a designer would be better able to understand the interactions between different norms, allowing them to avoid creating redundant norms, and specify their norms more precisely. A user would be able to get a more intuitive understanding of the system by understanding the reason why certain norms were assigned a certain status 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 based 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.

## 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, the CONTRACT project 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 or not 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 which is not in force is referred to as *abstract*, and has ungrounded 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:

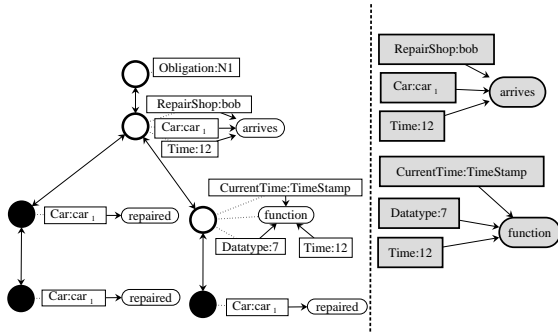
$$\langle obligation, \\ arrivesAtRepairShop(X, Car, T_1), \\ repaired(Car) \vee (currentTime(CurrentTime) \wedge \\ before(CurrentTime, T_1 + 7days)), \\ repaired(Car), \\ repairShop(X) \rangle$$

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

## 3. A CONCEPTUAL GRAPH BASED REPRESENTATION

A basic graph (BG) is a bipartite graph: one class of nodes, called the *concept* nodes, represent entities and the other, called the *relation* nodes, represent relationships between these entities or their properties. The nodes and the relations are organised in the vocabulary. The vocabulary is thus composed of two partially ordered sets: a set of concepts and a set of relations of any arity (the arity is the number of arguments of the relation). The partial order represents the specialisation relation. These graphical objects are provided with a semantics in first order logic, defined by a mapping classically denoted by  $\Phi$  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 using conceptual graphs by the means of a tree where the 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 fourth 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.



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

In order to map a norm into a norm tree, represent the norm using the disjunctive normal form of its elements, i.e. a norm  $n_1$  can be 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. It should be noted that by making use of negation as failure, we may remove the positivity requirement from this definition.

## 4. DISCUSSION

Much of the existing work on norms and normative reasoning originated from the philosophical domain. Norms in this domain were comparatively simple entities, typically having an antecedent and consequent, and were thus (relatively) easy to understand. Work on such norm emphasised problems such as identifying what state of affairs should hold, or how to resolve normative conflict. However, norms in the real world may have a complex internal structure (as seen by the model proposed above), some technique for explaining why a norm is in a certain state, such as violated or complied

with, is required, and we proposed a visual model for explaining the status of a norm. 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 of particular use. For example, a complex contract dispute may require that some rewards or penalties be assigned by a human mediator. In order to understand what rewards or penalties should be assigned, the mediator must first understand which norms were violated, and which were obeyed.

Norm explanation is also important at the system design stage. A designer may model the system, and must then understand what norms could be violated at different points in time.

Very little work appears to deal with the explanation of norms, instead assuming that the system will be 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.

Work such as [3] attempted to explain the causes of a norm violation by making use of a causal graph. This explanation was then fed into a policy engine which attempted to determine whether some sort of mitigating circumstances for the violated existed. If such circumstances were present, enforcement of penalties against the violator could be ignored, or reduced.

## 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.

The use of conceptual graphs to provide the formal underpinnings of our representation will allow us to extend this work in a number of interesting directions.

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, and 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, Berlin, Heidelberg, 2008. Springer-Verlag.
- [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, April 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.