On-line reasoning for institutionally-situated BDI agents (Extended Abstract)

Tina Balke
Inf. Sys. Management
University of Bayreuth
tina.balke@uni-bayreuth.de

Marina De Vos, Julian Padget
Dept. of Computer Science
University of Bath
{mdv,jap}@cs.bath.ac.uk

Dimitris Traskas
CACI Ltd.
Andover, UK
dtraskas@gmail.com

ABSTRACT

Institutions offer the promise of a means to govern open systems, in particular, open multi-agent systems. Research in logics and their derived tools now support the specification, verification and enactment of institutions (or organizations, depending on the terminology of the tool). Most effort to date has tended to focus on the static properties of institutions, such as whether a particular state of affairs is reachable or not from a given set of initial conditions. Such models are useful in forcing the designer to state their intentions precisely, and for testing (static) properties. We call this off-line reasoning. We identify two problems in the direct utilization of off-line models in the governance of live systems: (i) static model artefacts that are typically aspects of agent behaviour in the dynamic model (ii) over-specification of constraints on actions, leading to undue limitation of agent autonomy. Agents need to be able to query an institution for (dynamic) properties. We call this on-line reasoning. In this paper we present a methodology to extract the on-line specification from an off-line one and use it to support BDI agents to realize a norm-governed multi-agent system.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: Applications; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents

General Terms

Theory,Verification,Algorithms

Keywords

institutions, simulation, belief-desire-intention, agents

Introduction The motivation for this work derives from the construction of a simulation to evaluate a possible future development for mobile phone networks, in which mobiles dynamically construct ad-hoc wireless grids with the objective of achieving (i) faster download times by splitting content into parts, downloading a subset using 3G and acquiring the rest from nearby phones using wifi (ii) reducing power consumption by trading off high-cost 3G communication for low-cost wifi communication [3]. In planning the simulation, rather than using the conventional marionette approach of agent-based simulation, we chose to explore the idea of using a social institution to guide and inform agent actions. Given the event-based nature of the simulation, we adopted the formal approach to institutional modelling described in [2]. Using its domain specific modelling language Inst*AL* , and its a complementary computational model, realized through Answer Set Programming (ASP), agents are provided with information about the institutional state. At the same time, we also needed a suitable agent architecture, with a programming model that would fit the requirements for both being able to process institutional events and taking a goal-driven approach to the tasks to be fulfilled in the simulation. We chose the BDI architecture as implemented in Jason [1].

We address the institutional modelling task in two phases: (i) Off-line: where we built an institutional model of the wireless grid concept to evaluate whether it makes sense to pursue the idea at all. This model hard-codes simplifications of the environment in which the agents interact. (ii) On-line: created by stripping the off-line model of everything except normative information and domain facts. It provides the BDI agents in the simulation with a kind of oracle, that can respond to queries both about the current state and the normative consequences of actions.

The experience gained during the development of this simulation has lead to the main contribution of this paper: a methodology for developing off- and on-line institutional models—that is, models that play a key part in developing and running either an application or, as in our case, a simulation, in expressing the rules of governance for an open system. In that respect, the simulation and its results are tangential to the present focus, which is normative design and making such models accessible to agents.

Norm Governed Systems We have two motivations in choosing a norm-governed approach: (i) *flexibility:* by changing the institutional model, it is possible to influence agent behaviour, without modifying individuals—assuming a suitable goal-driven agent implementation (ii) *realism:* in this scenario, as in those foreseen for multi-agent systems, we cannot either predict or control with total certainty the behaviour of agents, but it is hoped that social institutions can provide functions similar to those found in the physical world, thus it is important to be able to test the potential impact of institutional control on suitably adapted agents.

Off- vs. On-line Most research to date on institutional modelling and reasoning focusses on the static properties of institutions. A model is used, for example, to determine whether a particular state of affairs is reachable or not from a given set of initial conditions. As such it can be used to design and verify properties of protocols and the effectiveness of sanctions. In our grid scenario, the off-line model was used as a prototype to demonstrate that normative reasoning can be beneficial to the individual agents.

The off-line model is an abstraction of a possible running system and cannot take into account participants' reasoning capabilities as some of the participants might not be norm-aware or even be irra-

Cite as: On-line reasoning for institutionally-situated BDI agents, T. Balke, M. De Vos, J. Padget and D. Traskas, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1109-1110.

tional. In the off-line model, it should be possible for participants to download the same chunk over and over again, while in reality this would be a waste of battery power. The model also does not have access to the information available in a running system so might have to manufacture some such information for itself. In the grid example this means that the off-line model has to keep track of which channels are in use at any given time in order to prevent simultaneous downloads on the channel. This also implies it has to monitor the duration of the download. The same is true for the sending and receiving of the chunks. In a running system this is taken care of by the system and its components (such as the base-stations) or the physical limitations of the devices.

The modelling of such extra details in the off-line model forces the designer to be very precise about his or her intentions, ultimately leading to better normative specifications.

For a given normative system, both the off-line and on-line model should have the same normative intentions, making the off-line model a good starting point for the development of the on-line one. A first step is to remove rules and conditions that deal with simulating a running system. The on-line model is only there to monitor normative behaviour not the system's behaviour. It only monitors the external events resulting from agent actions, however, it does not predetermine all agent behaviour.

The Off-line Model In neither model are we concerned with the technicalities of the negotiation phase—any off-the-shelf protocol could be employed—as long as the post-condition is satisfied: that each chunk is assigned to exactly one handset and that each handset is assigned the same number of chunks.

The results received from this off-line model verify that when agents follow the norms the entire community benefits, except when norms are breached at the end of the interaction as enforcement have no longer an effect. However, this might not cause problems when participants never meet again, penalties can always be applied at the next encounter. This information gives us sufficient reassurance to implement the protocol in our energy-saving simulation where handsets might engage in several sharing contracts over a period of time and past information can be used against them and propagated in the network.

The On-line Model When moving to an on-line model we no longer need to be concerned with modelling system data. In a running system, the sole purpose of the normative component is to monitor agents' actions and verify whether they were allowed or not from a normative perspective. Concretely for our example this means that our model should not concern itself with any restriction from a technical perspective, i.e. whether a mobile phone is technically capable to send or receive chunks.

In contrast to the off-line model, in which the chunk attribution to agents (i.e. the initial configuration of the agent/chunk/channel combinations) is pre-determined, in the on-line model this is decided by the agents themselves. So a dynamic normative specification consists of two parts: a static part that is independent of the participating handsets and contains the general norms for cooperation; and a dynamic part which is determined at run-time with handsets form sharing coalitions.

Monitoring On-line State For maintaining the institutional state in our running system we introduce a special type of agent or entity: the Governor. When created it is given the static part of the on-line model. When our agents agree to collaborate they create a contract specifying the agents involved and who is responsible for downloading with chunks from the base-station This information is then turned into a custom dynamic instantiation of the institution. Whenever an action takes place that affects the contract, the Governor is informed who then updates the normative state for that particular contract using the current state of contract as the initial state.

The agents involved with the contract can then pose queries to the Governor regarding the state and possible consequences of certain actions, such as (i) what norms affect my current situation, (ii) is a specific norm X true (i.e. valid) in the current situation, or more specifically, (iii) given the current situation, following the norms, am I allowed to execute action Y? In terms of the on-line reasoning model these questions query the properties of fluents at the current state.If this is the case, the action is permitted, otherwise, the agent does not have permission to perform the action, however it can choose to act in contravention of the norm.

Another class of questions are exemplified by "What is going to happen if I take action Y (e.g. download chunk x1 from channel 1)"? In terms of the normative framework this question is executed almost like the normal processing of an "exogenous event" (i.e. agent action) described earlier. Thus, the current state is used as initially part of the dynamic InstAL-specifications and Inst*AL* is run with the new query-event as input over one time-step.The answer set solver returns a trace containing the queried event which is passed to the agent that has asked the query. However, in contrast to the normal handling of exogenous events, the results of the query are not stored in the associated contract, i.e. no new state is created.

A third class of questions that can be answered concern the future, such as: (i) What would happen if a series of actions (e.g. actions A,B,C and D) take place?, or (ii) Is it possible to end in state Y (e.g. being cheated on) from here within n timesteps? If the result is an answer set, the query is true, otherwise it is false.

BDI Agents and Institutions For the implementation of the online reasoning we use the Jason platform [1], a Java-based interpreter for an extended version of AgentSpeak. We linked it to the institutional model and answer set solver using system calls. Agents can query the Governor about the current state of the institution (fluents), about existing norms as well as potential results of actions. This is done whenever the current step of the agent's reasoning cycle requires perceptions and as a result, an update of the agent's belief base takes place; i.e. the agent stores the percepts in its belief base and can use them for reasoning from that point onward. Based on its internal reasoning, an agent will perform actions in the MAS. These actions are registered in the environment and result in exogenous events, about which the governor is informed, and which may trigger institutional events in a direct reflection of the counts-as principle and thereby change the state of the institution.

Conclusions In this paper we demonstrated that social institutions can be used in running multi-agent systems. To do so, the traditional off-line model allowing for verifying static properties of the modelled system can be reduced to a more compact on-line model that just contains normative information and relevant domain fluents and permission related sanctions.

REFERENCES

[1] R. H. Bordini, M. Wooldridge, and J. F. Hübner. *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. John Wiley & Sons, 2007.

[2] O. Cliffe, M. De Vos, and J. Padget. Specifying and reasoning about multiple institutions. In *Coin*, volume 4386 of *LNAI*, pages 67–85. Springer Berlin / Heidelberg, 2007.

[3] F. H. P. Fitzek and M. D. Katz. Cellular controlled peer to peer communications: Overview and potentials. In *Cognitive Wireless Networks*, pages 31–59. Springer, 2007.