Prognostic normative reasoning in coalition planning*
(Extended Abstract)

Jean Oh
Felipe Meneguzzi
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
jeanoh@cs.cmu.edu
meneguzz@cs.cmu.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
katia@cs.cmu.edu

Timothy J. Norman
Dept. of Computing Science
University of Aberdeen
Aberdeen, UK
t.j.norman@abdn.ac.uk

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Languages

Keywords

Proactive assistant, Norms, Prognostic normative reasoning

INTRODUCTION

Human users planning for multiple objectives in coalition environments are subjected to high levels of cognitive workload, which can severely impair the quality of the plans created. The cognitive workload is significantly increased when a user must not only cope with a complex environment, but also with a set of unaccustomed rules that prescribe how the coalition planning process must be carried out. In this context, we develop a prognostic assistant agent that takes a proactive stance in assisting cognitively overloaded human users by providing timely support for *normative reasoning*–reasoning about prohibitions and obligations.

Existing work on automated norm management relies on a deterministic view of the planning model [1], where norms are specified in terms of classical logic; in this approach, violations are detected only after they have occurred, consequently assistance can only be provided after the user has already committed actions that caused the violation [3]. By contrast, our agent predicts potential future violations and proactively takes action to help prevent the user from violating the norms.

Here, we introduce the notion of *prognostic normative reasoning* so that the agent can reason about norm-compliant planning in advance. In order for that, we use probabilistic plan recognition to predict the user's future plan steps based on the user's current behavior and the changes in her environment. As the environment changes the agent's prediction is continuously updated, and thus its plan for remedial actions must be frequently revised during execution. In order to address this issue, our agent system supports a full cycle of autonomy including planning, execution, and replanning. This paper is specifically focused on the agent's prognostic normative reasoning.

*This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Cite as: Prognostic normative reasoning in coalition planning (Extended Abstract), J. Oh, F. Meneguzzi, K. Sycara, T. Norman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1233-1234.

PROGNOSTIC NORMATIVE REASONING

Our approach integrates plan recognition with normative reasoning. To illustrate our approach, we use a peacekeeping scenario, whereby military forces cooperate with various humanitarian coalition partners including the United Nations and Non-Governmental Organizations (NGOs). In this context, we consider the rules that regulate NGO operations in conflict areas, *e.g.*, an armed escort is required to transport relief supplies through certain routes.

Probabilistic plan recognition

From observing a user's current activities, the agent predicts the user's future activities as follows. We assume that a user's planning problem is given as a Markov Decision Process (MDP). Based on the assumption that a human user generally reasons about consequences and makes decisions to maximize her long-term rewards, we utilize an optimal stochastic policy of the MDP to predict a user's future activities.

The plan recognition algorithm is a two-step process. In the first step, the algorithm estimates a probability distribution over a set of possible goals. We use a Bayesian approach that assigns a probability mass to each goal according to how well a series of observed user actions is matched with the optimal plan toward the goal. We assume that the agent can observe a user's current state and action. Let $O_t = s_1, a_1, s_2, a_2, ..., s_t, a_t$ denote a sequence of observed states and actions from time steps 1 through t where s_t and a_t denote the user state and action, respectively, at time step t.

When a new observation is made, the agent updates, for each goal g, the conditional probability $p(g|O_t)$ that the user is pursuing goal g given the sequence of observations O_t. The conditional probability $p(g|O_t)$ can be rewritten using Bayes' rule as:

$$p(g|O_t) = \frac{p(s_1, a_1, ..., s_t, a_t|g)p(g)}{\sum_{g' \in G} p(s_1, a_1, ..., s_t, a_t|g')p(g')}. \qquad (1)$$

By applying the chain rule, we can write the conditional probability of observing the sequence of states and actions given a goal as:

$$\begin{aligned} p(s_1, a_1, ..., s_t, a_t|g) &= p(s_1|g)p(a_1|s_1, g)p(s_2|s_1, a_1, g) \\ &\dots \; p(s_t|s_{t-1}, a_{t-1}, ..., g). \end{aligned}$$

We replace the probability $p(a|s,g)$ with the user's stochastic policy $\pi_g(s,a)$ for selecting action a from state s given goal g. By the MDP problem definition, the state transition probability is independent of the goals. Due to the Markov assumption, the state transition probability depends only on the current state, and the user's action selection on the current state and the specific goal. By using these conditional independence relationships, we get:

$$\begin{aligned} p(s_1, a_1, ..., s_t, a_t|g) &= p(s_1)\pi_g(s_1, a_1)p(s_2|s_1, a_1) \\ &\quad ... \quad p(s_t|s_{t-1}, a_{t-1}), \end{aligned} \tag{2}$$

By combining Equations 1 and 2, the conditional probability of a goal given a series of observations can be obtained.

In the second step, we *sample* likely user actions in the current state according to a stochastic policy of each goal weighted by the conditional probability from the previous step. Subsequently, the next states after taking each action are sampled using the MDP's state transition function. From the sampled next states, user actions are recursively sampled, generating a tree of user actions known here as a *plan-tree*. The algorithm prunes the nodes with probabilities below some threshold. A node in a plan-tree can be represented in a tuple $\langle t, s, l\rangle$ representing the depth of the node (*i.e.*the number of time steps away from the current state), a predicted user state, and an estimated probability of the state visited by the user, respectively. Example 1 shows a segment of plan-tree indicating that the user is likely be in area 16 with probability .8 or in area 15 with probability .17 at time step t_1.

EXAMPLE 1. $\langle\langle t_1, (area = 16), .8\rangle, \langle t_1, (area = 15), .17\rangle\rangle$

Normative reasoning

After predicting a user's plan, the agent evaluates the predicted plan according to a set of normative regulations to prevent any potential violations. Norms generally define constraints that should be followed by the members in a society at particular points in time in order for them to be compliant with societal regulations. Formally,

DEF. 1 (NORM). *A norm is a tuple $\langle\nu, \alpha, \mu\rangle$, where the deontic modality $\nu \in \{\mathbf{O}, \mathbf{F}\}$ and $\mathbf{O}$ and $\mathbf{F}$ denote* obligations *and* prohibitions*, respectively; α is a formula specifying when the norm is relevant to a state (*context condition*); and μ, a formula specifying the constraints imposed on an agent when the norm is relevant (*normative condition*).*

EXAMPLE 2. *An intelligence message notifies that regions 3, 16 and 21 are unsafe. The norm, denoted by ι_{escort}, that an NGO is obliged to have an armed escort can be expressed as:*

$$\iota_{escort} = \langle\mathbf{O}, area \in \{3, 16, 21\}, escort = granted\rangle.$$

DEF. 2 (SATISFIABILITY). *A context condition α or a normative condition μ containing variables $\{\varphi_k \dots \varphi_m\} \subseteq \vec{\varphi}$ with specified domains $d_{\varphi_k}, \dots d_{\varphi_m}$ is* satisfiable *in state s (so that $s \models \alpha$) if the value assigned to the variables in state s is within the domain specified for the variables in condition α, so that $\forall\varphi_j \in \alpha.(\varphi_j = v) \wedge (v \in d_{\varphi_j})$.*

When a state is relevant to a norm – *i.e.*, the norm's context condition is satisfied in the state – a normative condition is evaluated to determine the state's compliance, which depends on the deontic modality of the norm. Specifically, an obligation is violated if the normative condition μ is not supported by state s; *i.e.*, $s \not\models \mu$. For instance, considering norm ι_{escort} in Example 2, given state $s = \{(area = 16), (escort = init)\}$ the violation detection function $violation(s, \iota_{escort})$ would return 1, denoting that norm ι_{escort} is violated in state s.

Given a predicted user plan in a plan-tree, the norm reasoner traverses each node in the plan-tree and evaluates the associated user state for any norm violations. For each state that violates a norm the agent needs to find a state that is *compliant* with all norms; *i.e.*, for each state s where $violating(s, \cdot) = 1$, the agent is to find the nearest state g that satisfies $violating(g, *) = 0$. Here, the distance between two states is measured by the number of variables whose values are different.

Since norm violations occur as the result of certain variables in the state space being in particular configurations, finding compliant states can be intuitively described as a search process for alternative value assignments for the variables in the normative condition such that norms are no longer violated, which is analogous to search in constraint satisfaction problems. When a norm-violating state is detected, the norm reasoner searches the nearby state space by trying out different value assignment combinations for the agent-variables. For each altered state, the norm reasoner evaluates the state for norm compliance. The current algorithm is not exhaustive, and only continues the search until a certain number of compliant states, say m, are found.

When compliant state g is found for violating state s, state g becomes a new goal state for the agent, generating a planning problem for the agent such that the agent needs to find a series of actions to move from initial state s to goal state g. The goals that fully comply with norms are assigned with *compliance level* 1. When a search for compliant states fails, the agent must proactively decide on remedial actions aimed at either preventing the user from going to a violating state, or mitigating the effects of a violation. In the norm literature these are called *contrary-to-duty obligations* [2]. For instance, a contrary-to-duty obligation in the escort scenario can be defined such that if a user is about to enter a conflict area without an escort, the agent must *alert* the user of the escort requirement. For such partial compliance cases, we assign compliance level 2.

EXAMPLE 3. *Let the domain of variable escort be: $\{init$, requested, granted, denied, alerted$\}$. Given a predicted plan-tree in Example 1, if variable escort for area 16 has value init indicating an escort has not been arranged, the agent detects a norm violation and thus searches for a compliant state as follows. By alternating values, we get two compliant states, where state $(granted)$ is fully compliant while state $(alerted)$ is partially compliant – as it complies with the contrary-to-duty obligation. As a result, a newly generated planning problem is passed to the planner module as follows: $\langle init, \{(granted, 1), (alerted, 2)\}\rangle$.*

CONCLUSION

The main contributions of this paper are the following. We developed a proactive assistant agent architecture where the agent autonomously identifies and performs new tasks in a principled way by integrating probabilistic plan recognition with reasoning about norm compliance. We introduced the notion of *prognostic norm reasoning* to predict the user's likely normative violations, allowing the agent to plan and take remedial actions before the violations actually occur. To the best of our knowledge, our approach is the first that manages norms in a proactive and autonomous manner.

REFERENCES

[1] S. Modgil, N. Faci, F. Meneguzzi, N. Oren, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *Proc. of AAMAS*, pages 153–160, 2009.

[2] H. Prakken and M. J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[3] K. Sycara, T. Norman, J. Giampapa, M. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. Strub. Agent support for policy-driven collaborative mission planning. *The Computer Journal*, 53(5):528–540, 2010.