On Programming Organization-Aware Agents

Andreas Schmidt JENSEN[1]
Technical University of Denmark, Kgs. Lyngby, Denmark

Abstract. Since it is difficult (or even impossible) to assume anything about the agents' behavior and goals in an open multi-agent system, it is often suggested that an *organization* is imposed upon the agents, whichhich, by abstracting away from the agents, specifies boundaries and objectives that the agents, by enacting roles, are expected to adhere to. In practice, this is usually done by creating a middleware, which acts as a bridge between an organizational specification and the agents, often taking away too much of the agents' autonomy. This project investigates how to make agents *organization-aware*, thus removing the middleware and letting the agents directly reason about the organization. In this paper, we discuss the results so far, and describe the future goals and research direction for the project.

Keywords. multi-agent systems, organizations, logic, programming

1. Introduction

In artificial intelligence, intelligent agents are entities that are situated in an environment in which they can *act* and *sense*. Intelligent agents are usually characterized by their ability be proactive, reactive, autonomous, and social. *Proactivity* enables agents to exhibit goal-directed behavior, *reactiveness* allows agents to adapt to changes in the environment, *autonomous* agents can choose by themselves how to achieve their goals, and, finally, *social* agents can communicate and cooperate with other agents. Intelligent agents are normally modeled using the beliefs-desires-intentions (BDI) model, which characterizes the agent's mental state by beliefs (about the environment and other agents), desires (what the agent would like to achieve) and intentions (desires that the agent has committed to achieve). A multi-agent system is a system of intelligent agents in an environment, and it is used to take advantage of the abilities of each intelligent agent in order to complete tasks that are difficult (or impossible) for the individual agents to complete.

When designing a multi-agent system in a fixed setting with a controlled number of agents and globally desirable states, it is often possible to implement the agents such that their own desirable states coincide with the globally desirable states. However, in open societies, agents from many different sources can enter and interact with each other. In the simplest case, a society is an environment that is accessible by anyone and that has no control over the agents entering it. Without control over the agents, it is not possible

[1] Corresponding Author: Andreas Schmidt Jensen, DTU Compute, Matematiktorvet, Building 303B, 2800 Kgs. Lyngby, Denmark; E-mail: ascje@dtu.dk.

for the society to ensure that global objectives are achieved and it is furthermore very hard to assume any kind of behavior of the agents in the system. One suggestion is to impose an organization on the agents, which can influence the actions of the agents toward the desires of the organization. Organizations consists of roles and groups, regulated by social rules and are created by actors to achieve common goals. Agents can enact roles in a society and interact with others as a means to accomplish their own goals.

Several organizational models have been proposed (MOISE^{+} [1]; OperA [2], EIDE [3]) as a way of distinguishing the aim of the system from the agents in terms of objectives, roles, groups and norms. Work has been done toward building agents that can enact roles in such organizations (e.g. S-MOISE^{+} [4] for the MOISE^{+} model, or AMELI in the EIDE framework), but the approach is usually to create a *middleware* responsible for understanding the organizational specification and ensuring that the agents do not violate it. While this effectively separates the agents and the organization, it also severely limits the possibilities of how to achieve the organizational (and the agents' own) objectives. Since agents are individual entities, a middleware is not able to take into account the *mental state* of an agent, meaning that organizational reasoning will be superficial at best. A middleware might very well put agents in undesirable situations, which could have been avoided if the agent's desires and beliefs were taken into account, thus by using a middleware, we risk taking away too much of the agents' autonomy.

This main goal of this project is to give the control back to the agents by letting them become *organization-aware*. Organization-aware agents are able to reason about organizational specifications, thus enabling them to reason about joining or leaving an organization, and about how to achieve organizational objectives while taking their own desires and intentions into account [5].

2. Aim & Approach

The project has both a theoretical and practical aspect. A theoretical foundation is required to make sure that the resulting organization-aware agents are sound and not implemented in an ad-hoc fashion. Work has been done in both areas, though the focus has primarily been on theory. For instance, the *Logic of Agent Organizations* (LAO) [6] aims to formalize the organizational structure and the agents acting within the organization making it possible to verify that the organization is, for example, well-defined and successful. As mentioned above, several organizational models have also been developed, making it quite possible to create organizational specifications. Furthermore, several practical tools are developed, in terms of both agent programming languages (e.g., GOAL, *Jason*, Jadex) and tools for specifying organizations (e.g., OperettA for OperA, S-MOISE^{+} includes an editor, and ISLANDER in the EIDE framework). Some integration has been made (e.g., S-MOISE^{+} has been integrated with *Jason*), but this work is, as mentioned above, done using middleware agents.

The primary objective is to enable organization-oriented programming for multi-agent systems. That is, agents should have access to an organizational specification and be able to understand and use it in their reasoning. Figure 1 illustrates how we aim to make agents organization-aware. Given the agent's desires and beliefs, the *organizational reasoning* component uses an organizational model to decide which objectives to commit to (thus becoming intentions) and how to cooperate and coordinate its organi-

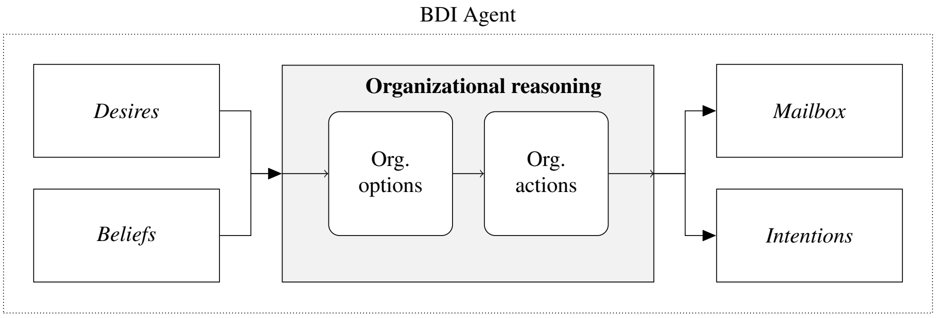

Figure 1. A BDI agent with an *organizational reasoning* component, which makes the agent organization-aware. Note that the usual BDI reasoning has been omitted for clarity.

zational tasks (by communicating via its mailbox). The organizational reasoning is split into two parts: *option consideration* and *action deliberation*. Options can, for example, be role responsibilities, while actions can be the commitment toward an objective, or delegation of a task to a sub-role. Note that the agent might still have private goals that do not coincide with the organizational objectives, which means that existing reasoning mechanisms (committing to desires, making them intentions) should still be working. This requires the agents to be able to *decide* which kinds of goals to commit to.

We thus need an organizational model and an agent programming language in order to program organization-aware agents. This project takes an existing organizational model and investigates how agents can be made organization-aware in an existing agent programming language. This is done for two reasons: first, since mature tools exist in both areas, it seems unnecessary to spend too much time "reinventing the wheel". Second, by choosing well-known tools, the integration will be readily useful for people who are familiar with each of the tools.

We have chosen to work with the OperA model (see [2] for a thorough introduction) and the GOAL agent programming language (see [7] for a description of the language), though we aim to extract *programming patterns* that will be useful in any integration of an organizational model and an agent programming language.

3. Results & Future Work

We have compared multi-agent systems with and without an organization and, although not decisive, the results suggest that in complex systems with many agents, organization-centered systems makes cooperation and coordination easier, while also making the code more clear and easy to maintain [8].

When agents have organizational objectives and private goals, deciding what to commit to achieving is not always straightforward, and even less so if the organizational objectives are in conflict with the agent's goals. In [9,10] we investigated how agents can make such decision based on *expected consequences*. The approach is based on a logic for qualitative decision theory, and we have made the first steps toward an implementation in Prolog, providing a way for agents to choose which objectives to (attempt to) achieve in different situations.

We have taken the first steps toward making agents organization-aware in [11], where we investigate how GOAL agents can reason about the OperA organizational model, specifically about organizational objectives. We have identified programming patterns for coordinating and delegating objectives, which is done by reasoning about responsibilities and role relations. Finally, we are working on a formalization of a Danish theatrical performance, which employ techniques that are similar to the OperA organizational model [12]. This allows us to formalize models, such that they can be used in, for example, GOAL, making it possible to simulate the theatrical performance.

We aim to continue investigating programming patterns for larger parts of the OperA organizational model, thus also identifying how easily each part can be integrated in GOAL. We will then use this when considering whether it would be beneficial to extend the GOAL agent programming language to incorporate organizational models in a more natural way. A long-term objective is to apply multi-agent organizations to a larger case with many heterogeneous agents, for example, a real-time strategy game, since such games require coordination, cooperation and the ability to handle many different kinds of agents simultaneously.

Finally, we aim to investigate how verification of organizational multi-agent systems can be done practically. As mentioned, organizations specified in LAO [6] can be shown to hold certain properties. However, such analysis is difficult by hand, and an implementation of the theory in, for example, Isabelle/HOL [13] would make it easier to verify that an organization is e.g. well-defined.

References

[1] J. F. Hübner, J. S. Sichman, and O. Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *SBIA '02 Proceedings*, pages 118–128, 2002.

[2] V. Dignum. *A Model for Organizational Interaction: Based on Agents, Founded in Logic*. PhD thesis, Utrecht University, 2004.

[3] J. L. Arcos, M. Esteva, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. An integrated development environment for electronic institutions. In *Software Agent-Based Applications, Platforms and Development Kits*, pages 121–142. Birkhäuser Basel, 2005.

[4] J. F. Hübner, J. S. Sichman, and O. Boissier. S-moise+: A middleware for developing organised multiagent systems. In *COIN I*, pages 64–78. Springer LNCS 3913, 2006.

[5] M. B. van Riemsdijk, K. Hindriks, and C. Jonker. Programming organization-aware agents. In *ESAW '09 Proceedings*, pages 98–112. Springer LNCS 5881, 2009.

[6] V. Dignum and F. Dignum. A logic of agent organizations. *Logic Journal of IGPL*, pages 283–316, 2011.

[7] K. V. Hindriks. Programming Rational Agents in GOAL. *Multi-Agent Programming: Languages, Tools and Applications*, 2:119–157, 2009.

[8] A. S. Jensen and J. Villadsen. A comparison of organization-centered and agent-centered multi-agent systems. *Artificial Intelligence Research*, 2:59–69, 2013.

[9] A. S. Jensen. Deciding between conflicting influences. In *Proceedings of the 1st International Workshop on Engineering Multi-Agent Systems*, pages 224–238, 2013.

[10] A. S. Jensen. Deciding between conflicting influences. In *Post-proceedings of the 1st International Workshop on Engineering Multi-Agent Systems*, 2013. To appear in Springer LNCS.

[11] A. S. Jensen, H. Aldewereld, and V. Dignum. Dimensions of organizational coordination. Submitted for publication, 2013.

[12] A. S. Jensen, J. Spurkeland, and J. Villadsen. Formalizing theatrical performances using multi-agent organizations. In *Proceedings of the 12th Scandinavian AI Conference*, 2013. To appear.

[13] T. Nipkow, M. Wenzel, and L. C. Paulson. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer LNCS 2283, 2002.