Adding Organizational Reasoning to Agent-Based Simulations in GAMA

John Bruntse Larsen

DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
jobla@dtu.dk

Abstract. The GAMA platform supports simulation with a bottom-up design from an agent perspective using a BDI framework. This paper proposes a design for implementing the AORTA framework for organizational reasoning in the GAMA platform to support combining a bottom-up BDI models with a top-down organizational model. The contribution is twofold: an operational semantics of the BDI framework in the GAMA platform, and an extension of it with operational semantics of AORTA.

1 Introduction

Social systems are systems which involve human interaction and decision making. Examples of social systems include private organizations, city regions and countries. Gaining insight into such systems is necessary for identifying workflows, bottlenecks and other important properties, but it is difficult because of the non-linearity of the systems. Agent-based simulation is an approach to gaining insight based on analysis of multiple runs of virtual simulation with agents that represent the real world actors in a social system. The advantage of the approach is that the designer of the simulation can focus on modeling the agents and have the system emerge as a result of their interaction, rather than having to model the system as an overall process. Agent-based simulation platforms, such as GAMA, provide general purpose tools to create environments and agents for any domain. In particular the BDI programming paradigm, which is also supported in GAMA, is a simple tool for modeling human reasoning in the agents. As argued in [1] however, the advances made in AI with frameworks and metamodels for agent environments and social systems could be further leveraged in agent-based simulation. In particular, the AORTA framework for adding organizational reasoning to agents can be useful for studying environments where humans enact roles and solve objectives of an organization. It enables BDI agents, modeled from a bottom-up perspective, to include organizational knowledge, modeled from a top-down perspective, in their reasoning and decision making. GAMA has useful features for setting up a simulation environment with geodata and supports BDI but has no development of organizational reasoning. We contribute to the development in two parts: we provide an operational semantics of the GAMA BDI agents and extend it with concepts and rules based on the operational semantics of the AORTA framework.

2 Background

First we present some background on the GAMA platform and AORTA.

2.1 BDI agents in GAMA

GAMA agents are programmed in the GAML language for programming reflexive agents [2,3]. The *simple-bdi* module extends agents with BDI-based behavior. The module is developed with efficiency and easy-of-use for simulation creators in mind. Core concepts of BDI agents in GAMA:

– Simulation environment - The agents are spatially situated in a simulation environment that controls time and synchronizes agent execution.
– Belief base - A set of predicates that define the agent's internal knowledge about the world or its own state.
– Desires - A set of predicates that define the things that the agent wants.
– Intentions - A set of predicates that define the things that the agent is actively trying to achieve.
– Perception statements - Statements that the agent uses to observe changes in the world and update its knowledge base accordingly.
– Rule statements - Statements that the agent uses to infer new knowledge.
– Plan statements - Statements that the agent uses to perform actions toward achieving specific intentions.
– Agent properties - An agent has properties similar to that of an object in OOP. An agent can update and check both its own properties and properties of other agents.

In each step of the simulation, every agent (i) perceives the environment and updates beliefs, (ii) continues its current plan if it is not finished, or (iii) selects a new plan and possibly new intention and executes that plan. Figure 1 shows a simplified diagram of the agent behavior, which is our outset for the operational semantics we present in Section 3. We refer to [2] for the full diagram.

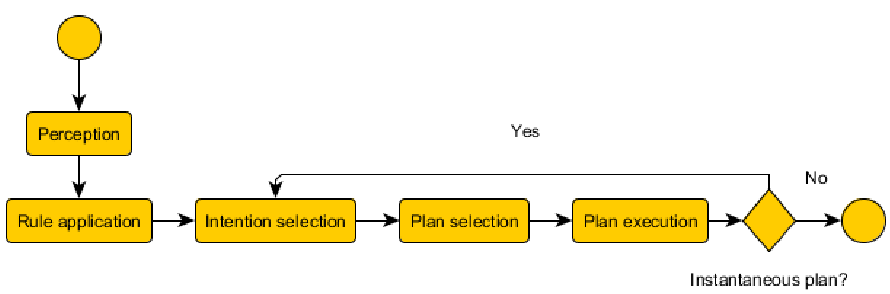

Fig. 1. Flowchart of agent behavior in GAMA.

2.2 AORTA

AORTA extends BDI agents with organizational reasoning capabilities according to the OperA meta-model [4], which gives a way of including a top-down model in a multi-agent system. Agents keep the model in the form of an organizational knowledge base that they maintain separately from their internal agent knowledge base. We highlight the parts of the operational semantics of AORTA that we extend GAMA agents with and refer to [5] for the full definitions.

Agent Configuration The mental state of an agent is based on knowledge bases $\mathrm{MS}_{AORTA} = \langle \Sigma_a, \Gamma_a, \Sigma_o, \Gamma_o \rangle$ where Σ_a and Γ_a are its beliefs and intentions, Σ_o is its organizational state and Γ_o are its organizational options. The mental state thus ensures that an agent can separate organizational and personal knowledge, and it is possible for agents to have different beliefs about an organization.

An agent configuration is defined as $A = \langle \alpha, \mathrm{MS}_{AORTA}, AR, F, C, \mu \rangle$ where α is the name of the agent, MS_{AORTA} is its mental state, AR are its reasoning rules, F is a set of transitioning functions, C are the capabilities of the agent, and $\mu = \langle \mu_{in}, \mu_{out} \rangle$ is its mailbox. Intuitively the agent configuration defines the state of the agent.

Transition System The semantics of AORTA is defined in terms of a transition system that transforms the agent configuration in a sequence of phases. In the obligation execution phase, the agent performs an obligation check where it adds activated obligations or obligation violations to Σ_o, and retracts satisfied obligations from Σ_o.

$$\text{Obl} ::= \text{Obl-Activated}^*; \text{Obl-Violated}^*; \text{Obl-Satisfied}^*$$

In the option execution phase, the agent generates organizational options and adds them to Γ_o. It can enact or deact a role, perform an objective, delegate objectives to other agents it depends on or inform others that depend on it about an objective.

$$\text{Opt} ::= \text{Enact}^*; \text{Deact}^*; \text{Objective}^*; \text{Delegate}^*; \text{Inform}^*$$

In the action execution phase, the agent considers its options, decides on a matching action reasoning rule to execute, and then executes the associated action. Executing an action updates both Σ_o and Γ_o.

$$\text{Act} ::= (\text{Act-Exec} | \text{Act-Send} | \text{No-Op})$$

External changes are handled in the (Ext) rule, and incoming messages are handled in the (Check) rule.

$$(\text{Ext}) : \frac{}{\mathrm{MS}_{AORTA} \rightarrow \mathrm{MS}'_{AORTA}}$$

$$(\text{Check}) : \frac{\text{msg}(sender, msg) \in \mu_{in} \quad \mathscr{M}(sender, msg, \mathrm{MS}_{AORTA}) = \mathrm{MS}'_{AORTA}}{\mu_{in} \rightarrow \mu_{in} \setminus \{\text{msg}(sender, msg)\} \quad \mathrm{MS}_{AORTA} \rightarrow \mathrm{MS}'_{AORTA}}$$

Bringing it all together, the organizational cycle execution is defined as follows.

$$\text{Org} ::= \text{Check}^*; \text{Ext}; \text{Obl}; \text{Opt}; \text{Act}$$

3 Operational Semantics for AORTA Agents in GAMA

We give an operational semantics for AORTA in GAMA, which comprises a design for implementing AORTA in GAMA.

3.1 GAMA BDI operational semantics

Given the list of concepts for GAMA BDI agents and the diagram in Figure 1, we write an operational semantics that we then extend with AORTA semantics.

We define an agent as $Agent = \langle P, \text{MS}_{GAMA}, Q, R, \Pi \rangle$ where P is a set of properties, $\text{MS}_{GAMA} = \langle B, D, I \rangle$ (with B, D and I being sets of predicates), Q is a set of **Perception** statements, R is a set of **Rule** statements and Π is a set of **Plan** statements of the form $t : c \rightarrow S$, where t is a trigger intention, c is a condition that must be *true* for the plan to be applicable, and S is a sequence of action statements. In GAMA, agents can read meta-data from the simulation environment such as the step counter or time between steps. For the purpose of extending with AORTA semantics however, we simplify the simulation environment Env to be the set of all agents in the simulation $Env : Agent\ set$.

Given the above definition of an agent and the environment, we can define the BDI reasoning in GAMA in terms of functions on its mental state, applying statements relevant to that step. Perception and Rule application include the simulation environment and agent properties as an agent can perceive not only other agents in the environment but also its own properties.

$$\begin{aligned}
&\text{Perception} ::= Q, Env, \text{MS}_{GAMA}, P \rightarrow \text{MS}'_{GAMA} \\
&\text{Rule application} ::= R, Env, \text{MS}_{GAMA}, P \rightarrow \text{MS}'_{GAMA} \\
&\text{Intention selection} ::= \text{MS}_{GAMA}, I_{cur} \rightarrow I'_{cur} \\
&\text{Plan selection} ::= \Pi, I_{cur}, \text{MS}_{GAMA} \rightarrow \Pi_{sel}
\end{aligned}$$

Having selected a plan to execute, the agent executes it which yields a new environment (and hence updated agents).

$$\text{Plan execution} ::= \Pi_{sel}, Env \rightarrow Env'$$

Given the above definitions, the activity semantics of a GAMA agent can be defined as the following sequence. If the selected plan is instantaneous, the agent may execute multiple plans for multiple intentions within one step.

$$\begin{aligned}
\text{Act} ::= \;&\text{Perception}; \text{Rule application}; \\
&(\text{Intention selection}; \text{Plan selection}; \text{Plan execution})^*
\end{aligned}$$

3.2 Extending with AORTA semantics

Having defined an operational semantics of GAMA BDI agents, we proceed by defining the AORTA agent semantics in terms of the GAMA BDI semantics. Doing so comprises a design for how the semantics can be implemented in GAMA.

First we define the mental state and the agent configuration. We use a naming scheme to separate organizational beliefs and goals from regular beliefs and intentions.

$$(\Sigma_a): \frac{b \in B \quad \mathit{prefix}(\mathit{pred}(b)) \neq \mathbf{O\_}}{b \in \Sigma_a} \qquad (\Gamma_a): \frac{i \in I \quad \mathit{prefix}(\mathit{pred}(i)) \neq \mathbf{O\_}}{i \in \Gamma_a}$$

$$(\Sigma_o): \frac{b \in B \quad \mathit{prefix}(\mathit{pred}(b)) = \mathbf{O\_}}{b \in \Sigma_o} \qquad (\Gamma_o): \frac{i \in I \quad \mathit{prefix}(\mathit{pred}(i)) = \mathbf{O\_}}{i \in \Gamma_a}$$

Next we define the name of an agent as simply the **name** property of the agents.

$$\alpha = \mathbf{agent.name}$$

The action reasoning rules AR are used in the Act phase to select an option, among those found in the Opt phase, and execute the action associated with that option. For example if the action is $\text{enact}(\rho)$, the agent adds $\text{rea}(\alpha, \rho)$ to Σ_o, and adds $\text{send}(\top, \mathit{tell}, \text{rea}(\alpha, \rho))$ to Γ_o. We define the reasoning rules in GAMA as a subset of instantaneous **Plan** statements that add intentions to Γ_a matching the action reasoning rules. We also use instantaneous **Plan** statements to define the set of transition functions of AORTA.

$$AR \subseteq \Pi \qquad\qquad F \subseteq \Pi$$

The capabilities of an agent are defined as the triggers of the plans in its plan library Π. Note that this is only a subset of the beliefs that the agent can make true, as carrying out a plan typically has side effects, but for simplicity we do not include beliefs from side effects in Π.

$$(C): \frac{t : c \rightarrow S \in \Pi}{t \in C}$$

As with the mental state, the mailbox is defined using a naming scheme that separates mailbox beliefs from regular beliefs.

$$(\mu_{in}): \frac{b \in B \quad \mathit{prefix}(\mathit{pred}(b)) = \mathbf{muIn\_}}{b \in \mu_{in}}$$

$$(\mu_{out}): \frac{b \in B \quad \mathit{prefix}(\mathit{pred}(b)) = \mathbf{muOut\_}}{b \in \mu_{out}}$$

Next we extend with the AORTA transition system.

Obligation execution We integrate obligation execution in the rule application step in GAMA, using the above definition of Σ_o and α. For simplicity, we only make **Rule** statements with grounded predicates, meaning that we need a statement for each grounded premises for both (Obl-Activated), (Obl-Satisfied) and (Obl-violated).

Option execution We also integrate option generation in the rule application step, with **Rule** statements that add new predicates to Σ_o, Γ_o and α.

Action execution We integrate action execution in the looping part of the activity semantics as instantaneous **Plan** statements. By making them instantaneous, the agent can perform an organizational action, such as enacting a role, updating its mental state and possibly sending a message to other agents, and still carry out an action as usual.

(Ext) and (Check) Same as in AORTA, with the mental state as defined above.

As a result we have defined AORTA semantics in terms of GAMA BDI operational semantics, which comprises a design for implementing AORTA in GAMA.

4 Evaluation

To demonstrate the usage of the operational semantics defined in the previous section, we show an example of the organizational reasoning that the agents perform. For the example we use an organizational meta-model based on the one in [6] (see Table 1), which defines a simplified organizational meta-model for patient treatment in a hospital emergency room. Due to space limitations, we leave out details of the example.

In the example we assume two agents p and n who initially have the following mental states:

- Σ_o (for both agents): as specified in Table 1, plus the following predicates: "**O_rea(patient, p)**" and "**O_rea(nurse, n)**". The condition in Σ_o states that the nurse should perform triage before a patient is treated.
- Σ_a (for both agents): contains "**patient(p)**".
- Γ_o, Γ_a (for both agents): empty.

We describe the updates that occur in the first loop of the GAMA activity semantics with AORTA.

Perception We assume none of the agents perform any **Perception** statements.

Rule application Both obligation execution and option execution takes place in this step. The **Rule** statements concerning obligation execution adds the predicate "**O_obl(n, nurse, triage(p), treatment(p))**" to Σ_o. The **Rule** statements concerning option execution then adds "**O_obj(triage(p))**" to Γ_o.

Intention selection Having "**O_obj(triage(p))**" in Γ_o, and thus in I, it is selected as the current intention for agent n.

Table 1. Initially Σ_o for all agents contains these predicates.

$\mathrm{role}(\mathit{patient}, \{\mathit{treatment}(\mathit{Patient})\})$
$\mathrm{role}(\mathit{nurse}, \{\mathit{triage}(\mathit{Patient})\})$
$\mathrm{obj}(\mathit{triage}(\mathit{Patient}), \{\})$
$\mathrm{dep}(\mathit{patient}, \mathit{nurse}, \mathit{triage}(\mathit{Patient}))$
$\mathrm{cond}(\mathit{nurse}, \mathit{triage}(\mathit{Patient}), \mathit{treatment}(\mathit{Patient}), \mathit{patient}(\mathit{Patient}))$

Plan selection and execution Having a matching action reasoning rule for the intention "**O_obj(triage(p))**", agent n then commits to **triage(p)** and subsequently adds it to Γ_a. As the plans for the action reasoning rules and the action transition function are instantaneous, agent n can then select a plan with "**triage**" as trigger and begin execution of that plan.

The example shows how a nurse agent can use a clearly defined organizational meta-model made from a top-down perspective to decide its course of action in patient treatment. To get similar behavior using only the existing BDI framework in GAMA, it would be necessary to design agents with a bottom-up method, which would make the organization less clear. By adjusting the action reasoning rules, we can also adjust how a nurse agent handles organizational obligations separately from how it handles its own intentions.

5 Related Work

We compare this work with other models and frameworks for social simulation. Network-oriented modeling has been applied for social system simulation to study the effects of various social parameters for the agent behavior on the outcome of the system [7–9]. Compared to the network model, AORTA is based on the BDI paradigm and logic. MOISE+ is an organization meta-model which has been implemented in the Jason agent programming platform [10,11]. In contrast we use AORTA, which has also been implemented in Jason [12], and GAMA, which is an agent-based simulation platform. There are also other methods to include normative reasoning in agents which do not incorporate an organization meta-model [13–15].

6 Conclusion

We have given an operational semantics for BDI agents in the GAMA platform for agent-based simulation, and extended them with concepts and rules that add organizational reasoning according to the AORTA framework. The extended semantics comprises a design for implementing AORTA in the GAMA platform. We have also shown the execution of the semantics with a small example. Future work includes more details in the semantics and implementing them in the GAMA platform with a larger example.

Acknowledgements

This work is part of the Industrial PhD project *Hospital Staff Planning with Multi-Agent Goals* between PDC A/S and Technical University of Denmark.

References

1. Larsen, J.B.: Agent programming languages and logics in agent-based simulation. In: Modern Approaches for Intelligent Information and Database Systems, Springer (2018) 512–526
2. Taillandier, P., Bourgais, M., Caillou, P., Adam, C., Gaudou, B.: A BDI agent architecture for the GAMA modeling and simulation platform. In: Multi-Agent Based Simulation XVII, Springer (2017) 3–23
3. Caillou, P., Gaudou, B., Grignard, A., Truong, C.Q., Taillandier, P.: A Simple-to-Use BDI Architecture for Agent-Based Modeling and Simulation. Advances in Intelligent Systems and Computing **528** (2017) 15–28
4. Dignum, V.: A Model for Organizational Interaction: based on Agents, founded in Logic. SIKS Dissertation Series 2004-1. Utrecht University (2004) PhD Thesis.
5. Jensen, A.S., Dignum, V., Villadsen, J.: A framework for organization-aware agents. Autonomous Agents and Multi-Agent Systems **31**(3) (2017) 387–422
6. Larsen, J.B., Villadsen, J.: An approach for hospital planning with multi-agent organizations. In: Rough Sets: International Joint Conference, IJCRS 2017, Part II, Springer (2017) 454–465
7. Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Understanding Complex Systems. Springer (2016)
8. Van der Wal, C.N., Formolo, D., Robinson, M.A., Minkov, M., Bosse, T.: Simulating crowd evacuation with socio-cultural , cognitive , and emotional elements. In: Transactions on Computational Collective Intelligence XXVII. Volume 10480., Springer (2017) 139–177
9. Formolo, D., Van Ments, L., Treur, J.: A computational model to simulate development and recovery of traumatised patients. Biologically Inspired Cognitive Architectures **21** (2017) 26–36
10. Hübner, J.F., Sichman, J.S., Boissier, O.: Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. International Journal of Agent-Oriented Software Engineering **1**(3/4) (2007) 370–395
11. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous Agents and Multi-Agent Systems **20**(3) (2010) 369–400
12. Jensen, A.S., Dignum, V., Villadsen, J.: The AORTA architecture: Integrating organizational reasoning in Jason. Lecture Notes in Computer Science **8758**(3) (2014) 127–145
13. Deljoo, A., van Engers, T., van Doesburg, R., Gommans, L., de Laat, C.: A normative agent-based model for sharing data in secure trustworthy digital market places. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART, SciTePress (2018) 290–296
14. Jager, W., Janssen, M.: An updated conceptual framework for integrated modeling of human decision making: The Consumat II. ECCS 2012 (2012)
15. Ghorbani, A., Bots, P., Dignum, V., Dijkema, G.: MAIA: A framework for developing agent-based social simulations. Jasss **16**(2) (2013) 1–15