

# Towards Multi-Level Semantics for Multi-Agent Systems

Alison R. Panisson<sup>1</sup>

*Postgraduate Programme in Computer Science – School of Informatics (FACIN)  
Pontifical Catholic University of Rio Grande do Sul – PUCRS  
Porto Alegre – RS – Brazil*

Rafael H. Bordini<sup>2</sup>

*Postgraduate Programme in Computer Science – School of Informatics (FACIN)  
Pontifical Catholic University of Rio Grande do Sul – PUCRS  
Porto Alegre – RS – Brazil*

Antônio Carlos da Rocha Costa<sup>3</sup>

*Graduate Program in Computers in Education – UFRGS, Porto Alegre, Brazil  
Graduate Program in Computer Science – FURG, Rio Grande, Brazil*

---

## Abstract

In multi-agent systems, changes in an agent's mental state often reflect changes at the social level as well. Furthermore, most work on agent languages use structural operational semantics to formalise multi-agent systems. As multi-agent systems are complex systems, it is difficult to formalise all relevant aspects of such system as a single transition system, and even transition systems for a single agent can easily become cumbersome. In this paper, we propose a particular style of semantic rules that make it visually clearer how changes at one level of a multi-agent system require simultaneous changes in other levels of the system (where each component of each level is modelled as a separate transition system).

*Keywords:* Multi-Level Semantics, Multi-Agent Systems, Agent-Based Communication.

---

## 1 Introduction

Communication is one of the key issues in building multi-agent systems, where the agents need to communicate in order to resolve differences of opinion or conflicts of interest, to work coordinately, to resolve dilemmas, and to reach agreements.

---

<sup>1</sup> Email: [alison.panisson@acad.pucrs.br](mailto:alison.panisson@acad.pucrs.br)

<sup>2</sup> Email: [rafael.bordini@pucrs.br](mailto:rafael.bordini@pucrs.br)

<sup>3</sup> Email: [ac.rocha.costa@gmail.com](mailto:ac.rocha.costa@gmail.com)

Many of these communication requirements cannot be fulfilled by the exchange of single messages. They require the exchange of sequences of messages upon related statements [18]. Therefore, agents need the ability to engage in multi-agent dialogues [1,22].

Formalisation of communication in multi-agent systems in the context of agent languages have often used structural operational semantics. In many of those formalisations, the focus is on receiving and sending of speech-acts at the agent-level [28,3]. However, more sophisticated communication methods such as argumentation-based dialogues [21,17] have other requirements, where the exchange of speech-acts has a social implication, i.e., it has an implication at the social as well as at the individual level. Furthermore, there are more than the obvious individual/social levels in multi-agent systems. Examples of a conception of various different levels in multi-agent systems can be found in [8,13,5].

In this paper, we propose a semantics representation with simultaneous multi-level transitions. In our approach, each transition represents a level (or a component of a level) with explicit reference to multiple levels of a multi-agent system. We argue that this representation facilitates the reading of the changes of each level individually as well as their relations, so that the representation of the various transitions matches the abstractions we typically have in multi-agent systems.

The remainder of this document is organised as follows. In the next section, we discuss communication in multi-agent systems. Next we put forward the idea of multi-level semantics, the main contribution of this paper. After that, we discuss some of the characteristics of this semantic style. We then discuss some related work and, in the final section, we conclude the work with some final remarks and pointing out possible directions for our work.

## 2 Social Communication in Multi-Agent Systems

In general, systems programmed using an agent-oriented programming language are not composed of a single agent; typically a set of agents share the same environment. This characteristic makes evident the need for communications (interaction) between agents in the system, hence requiring some sort of *social ability* for *cooperation* and *coordination* [6].

The communication between agents is typically via message passing. This communication is often based on speech act theory, where speech acts are treated as actions<sup>4</sup>. In general, a speech act can be seen as having two main components: a **performative** (representing the sender's intention, e.g., achieve, tell, etc.) and a (propositional) **content**. For example, in the Jason [6] multi-agent programming language, communication is done through the pre-defined internal action '*.send*' that has the following format:

*.send(receiver, illocutionary\_force, propositional\_content)*

<sup>4</sup> Utterances have the characteristic of *actions*, in the sense that they change the state of the world in a way analogous to physical actions [2].

where *receiver* is the name of an agent (each agent has a unique individual name in the multi-agent system), or a list of agent names, to whom the message is being sent. The *propositional\_content* is a term in AgentSpeak (a literal, triggering event, plan, or a list of literals or plans). The *illocutionary\_force* denotes the intention of the sender (often called *performative*), as in speech-act theory. The formal semantics of receiving such messages is given in [28], and a complete list of all the illocutionary forces available can be found in [6].

Furthermore, communication in multi-agent systems often require the exchange of sequences of messages upon related statements, for example in communication protocols, and more recently also in argumentation-based dialogues for agents developed under the agent-oriented programming paradigm [18,20,19]. In argumentation-based dialogues [1,22], agents exchange a sequence of arguments to defend their positions in such dialogues. Furthermore, in a dialogue, agents are committed to their claims — they declare a social commitment<sup>5</sup>, not only to believe but also to defend what was claimed [26,27].

Considering the aspects mentioned above, the agents are supposed to have a so-called *commitment store* (CS) (other names can be found for *CS* such as *dialogue obligation store* in [16] and *dialogue store* in [24]). Such structures (the CS) maintain, at the social level, the assertions made by agents during the dialogue (all agents participating in a dialogue have access to those commitments). Clearly, in this context, communication also has consequences at the social level. The same occurs when an agent creates a new prohibition or obligation in normative systems [4] (supposing that this agent has the authority to do this), or further, when an agent commits to execute a task for another agent in the society [17].

### 3 Multi-level Semantics Representation

Multi-agent systems are distributed systems. In the process of engineering a multi-agent system, we need to specify in a rigorous manner the communication among the agents through interaction protocols [7]. Further, we need to specify the exact effect that this communication has over the agent state (e.g., the so-called *mental states* from BDI — *Beliefs-Desires-Intentions* — architecture). In some forms of agent interaction, communication affects not only the agent state, but also the multi-agent system as whole, i.e., there are effects on the *social level* as well (e.g., in social commitments, normative systems, argumentation-based dialogues, etc.). These aspects bring to light the need to specify these changes to maintain the rigorousness of multi-agent systems specification and to prove formal properties of those communication methods.

Considering these new representation needs, we propose a multi-level semantics representation, where we are interested in the effect that action (including communicative actions based on speech act) causes in the individual agent state as well as in the multi-agent system as a whole.

<sup>5</sup> Singh [26,27] argues that agents are social entities, therefore, when involved in social interactions, they are committed to what they say.

We propose the definition of a multi-level semantics using operational semantics, a widely used method for giving semantics to programming languages [23]. The operational semantics is given by a set of inference rules that define a transition relation between configurations of the multi-agent system — the *social-level* — and the agent configuration — the *agent-level*.

The main idea behind our approach for multi-level semantics is to facilitate giving semantics to systems with multiple but inter-related levels (e.g., the levels of abstraction in multi-agent systems) specially where individual levels of the system have already been given semantics through independent transition systems. Our approach allows constraints to be added to the existing semantic rules giving semantics to the inter-relations among the levels, that is, making clear how events occurring at one level of the system affects another level specifically or various other levels simultaneously. The specification of independent transition systems for each level of abstraction allows a series of benefits, among them: modularity, visually more elegant and understandable semantic rules, as well as allowing different granularity of specification for each level. We discuss these features of our approach in Section 4.

In order to demonstrate our multi-level semantics approach, in the next sections we describe a simplified multi-agent system configuration and an example system formalised using the semantic style we proposed.

### 3.1 Multi-Agent System Configuration

To exemplify our approach, we will use abstract components based on the BDI architecture. For the propose of this work, we are only interested in showing the multi-level semantics representation rather than specifying any multi-agent system language or technique. The BDI architecture is one of the most studied architectures for cognitive agents; it provides a particular structure for agents' internal states based on “mental attitudes”. The internal state of a BDI agent is composed of: (i) *Beliefs* that represent the information about the world (including itself and other agents) available to that agent; (ii) *Desires* representing the motivations of the agent, i.e., the states of the environment that the agent would like to reach; and (iii) *Intentions* which are desires that the agent is committed to achieve by following particular *plans* of action. Most of the BDI languages use the concept of an *applicable plan*, i.e., the plan has a context, and it is applicable if the context conditions are satisfied.

We define the multi-agent system configuration as the tuple  $\langle AG, CL, CO \rangle$ , where the *AG* component is a set of BDI agents in the society (i.e., in the multi-agent system), where each agent is identified by a unique identifier *id*. The *CL* component contains the specification of communication link between agents, where two agent  $ag_1$  and  $ag_2$  are able to communicate only if there exist  $link_{com}(ag_1, ag_2)$  or  $link_{com}(ag_2, ag_1) \in CL$ , meaning that agents  $ag_1$  and  $ag_2$  have a communication link. Such specification is commonly found in organisational model as MOISE [15].

The component  $CO$  represents the agents' social commitments<sup>6</sup>.  $CO$  is a set of tuples in the format  $\langle ag_1, ag_2, commit \rangle$ , meaning that agent  $ag_1$  has the commitment  $commit$  with the agent  $ag_2$ , where  $commit$  could be a commitment towards a payment, to perform an action, etc.

The agent configuration is given by a tuple  $\langle ag, B, I, A, M, S \rangle$ , where:

- $ag$  is the agent identifier.
- $B$  is the set of beliefs.
- $I$  is a set of *intentions*  $\{i, i', \dots\}$ . Each intention  $i$  is a stack of partially instantiated plans.
- $A$  is a set of *actions* available to be performed in the environment.
- $M$  is a tuple  $\langle In, Out \rangle$  whose components characterise the following aspects of communicating agents.
  - $In$  is the mail inbox: the multi-agent system is assumed to have a runtime infrastructure which includes all messages addressed to this agent in this structure. Elements of set  $In$  have the form  $\langle mid, id, ilf, cnt \rangle$ , where  $mid$  is a message identifier,  $id$  identifies the sender of the message,  $ilf$  is the illocutionary force of the message, and  $cnt$  its content.
  - $Out$  is where the agent posts messages it wishes to send; it is assumed that some underlying communication infrastructure handles the delivery of such messages. Messages in this set have exactly the same format as above, except that here  $id$  refers to the agent to which the message is to be sent.
- $S$  is the current state of the agent reasoning cycle, which will be updated when transitions occur, but we will not mention exactly what the state (as we do not use a concrete example here).

In the interests of readability, we adopt the following notational conventions in our semantics rules:

- If  $M$  is the message component, we write  $M_{In}$  to make reference to the  $In$  component of  $M$ , and similarly for other components of the multi-agent system and of the configuration of each agent.
- We write  $B^{ag}$  to identify the beliefs (the component  $B$ ) of an agent identified by  $ag$ . We use this whenever it is necessary to identify the components of a specific agent, e.g., when we are treating two agents in the same semantic rule.
- We use two transitions to represent the state change of the multi-agent system (the main contribution of this work), where the transition  $\longrightarrow_A$  correspond to the transition of the configuration of an individual agent. The transition  $\longrightarrow_{MAS}$  correspond the transition of the multi-agent system. Each transition in the agent configuration is treated independently, where actions executed by such agents may cause a transition in the multi-agent system, e.g., an agent may change the multi-agent configuration, an agent may make a new commitment, etc.

<sup>6</sup> We use only components that are needed to demonstrate the semantics, but we emphasise the existence of other components such as roles, norms, etc.

- We use three functions in our semantics rules, defined as follows: (i) *ExecutableAct()* function, which returns all action that given agent is able to execute; (ii) the *Current()* function, which receives as parameter the set of agent intentions and returns the intention that has been selected for update/execution in that reasoning cycle, and; (iii) the *SelMsg* function, which receives as parameter the set of messages received and returns the message that has been selected to be treated in that reasoning cycle.

### 3.2 Example

In order to exemplify our approach, we use the following example: an agent, named  $ag_1$ , at a certain moment of the system execution, needs to commit itself to execute an action  $a_1$  for another agent, named  $ag_2$ . The only way to do that is to send a message to  $ag_2$  with the specific speech-act *commit*. However, there is no communication link between agents  $ag_1$  and  $ag_2$  in the organisational specification, which makes that the organisational infrastructure to block such communication. In order to communicate to  $ag_2$ ,  $ag_1$  needs to change the organisational specification executing the action  $.add\_com\_link(ag_1, ag_2)$ , which has the following semantics:

$$\begin{array}{c}
 i[head \leftarrow .add\_com\_link(ag_1, ag_2); h] = Current(I^{ag_1}) \\
 .add\_com\_link(ag_1, ag_2) \in ExecutableAct(A^{ag_1}) \\
 ag_1, ag_2 \in AG \\
 \hline
 (a) \quad \langle AG, CL, CO \rangle \xrightarrow{MAS} \langle AG, CL', CO \rangle \\
 (b) \quad \left[ \begin{array}{l} \langle ag_1, B, I, A, M, S \rangle \\ \langle ag_2, B, I, A, M, S \rangle \end{array} \right] \xrightarrow{A} \left[ \begin{array}{l} \langle ag_1, B', I', A, M, S_{t+1}, \rangle \\ \langle ag_2, B', I, A, M, S_{t+2} \rangle \end{array} \right]
 \end{array} \quad (AddComLink)$$

where:

$$\begin{array}{ll}
 (a) \quad CL' &= CL \cup link_{com}(ag_1, ag_2) \\
 (b) \quad I^{ag_1'} &= (I^{ag_1} \setminus \{Current(I^{ag_1})\}) \cup \{i[head \leftarrow h]\} \\
 B^{ag_1, ag_2'} &= B^{ag_1, ag_2} + com\_link(ag_1, ag_2)
 \end{array}$$

In this semantic rule, the agent represented by  $ag_1$  is following a plan associated with a particular intention — the *current* intention within its reasoning cycle. The next plan step is to execute the action  $.add\_com\_link(ag_1, ag_2)$ , which adds a communication link between the agents  $ag_1$  and  $ag_2$ . When the agent executes this action it updates the organisational specification. From that moment, agents  $ag_1$  and  $ag_2$  are able to communicate to each other<sup>7</sup>. Furthermore, both agents become aware of such communication link, updating their belief bases.

<sup>7</sup> Note that we consider the communication link a symmetric relation.

After agent  $ag_1$  adds the communication link in the organisational specification, the agent is able to communicate to agent  $ag_2$ . Therefore, the agent is able to send a message committing itself to execute action  $a_1$ .

$$\begin{array}{c}
 i[head \leftarrow .send(ag_2, commit, execute(a_1)); h] = Current(I) \\
 a_1 \in ExecutableAct(A) \quad link_{com}(ag_1, ag_2) \in CL \\
 \hline
 ag_1, ag_2 \in AG \quad (SND COMMIT) \\
 (a) \quad \langle AG, CL, CO \rangle \longrightarrow_{MAS} \langle AG, CL, CO' \rangle \\
 (b) \quad \langle ag_1, B, I, A, M, S \rangle \longrightarrow_A \langle ag_1, B, I', A, M', S_{t+1} \rangle \\
 \text{where:} \\
 (a) \quad CO' = CO \cup \{ \langle ag_1, ag_2, execute(a_1) \rangle \} \\
 (b) \quad M'_{Out} = M_{Out} \cup \{ \langle mid, ag_2, commit, execute(a_1) \rangle \} \\
 I' = (I \setminus \{Current(I)\}) \cup \{i[head \leftarrow h]\}
 \end{array}$$

In this semantic rule, the agent represented by  $ag_1$  is following a plan associated with a particular intention — the *current* intention on its reasoning cycle. The next plan step is to execute the action *.send* with the performative *commit*, where the agent is committing itself with the receiver of the message (identified by  $ag_2$  in the above example) to execute action  $a_1$ . When the agent executes this action it updates the commitment base, representing the social commitments in the multi-agent system. Besides, the agent sends the respective message, committing itself to execute the action, and updates its intention considering the action of sending the message as executed.

The semantic rule for receiving such message is as follows:

$$\begin{array}{c}
 SelMsg(M_{In}) = \langle mid, ag_1, commit, execute(a_1) \rangle \\
 \hline
 (a) \quad \langle AG, CL, CO \rangle \longrightarrow_{MAS} \langle AG, CL, CO \rangle \\
 (b) \quad \langle ag_2, B, I, A, M, S \rangle \longrightarrow_A \langle ag_2, B', I, A, M', S_{t+1} \rangle \\
 \text{where:} \\
 (b) \quad M'_{In} = M_{In} \setminus \{ \langle mid, ag_1, commit, execute(a_1) \rangle \} \\
 B' = B + execute(a_1)[source(ag_1)]
 \end{array}
 \quad (RECEIVING COMMIT)$$

In this semantic rule, the agent is treating the received message (identified by *mid*) from the agent  $ag_1$ , which has committed to it (indicated by the performative *commit*) to execute the action  $a_1$  (indicated by the content of the message – *execute(a<sub>1</sub>)*). The message is removed from the component  $M_{In}$  (the mail inbox of the agent), and the agent which received the message updates its belief base, being aware of such commitment.

An important point to note in our examples is that the components which correspond to the multi-agent system are not updated in all semantics rules. This occurs because the agents and multi-agent system configurations are treated by different transition systems which bring to light the interesting properties discussed in the next section.

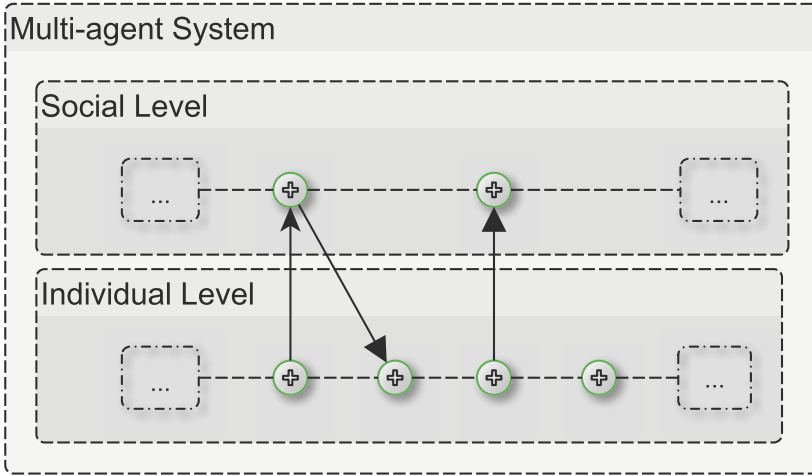


Fig. 1. Level interactions in multi-agent systems, considering an agent level and a generic social level.

We can observe that there are different interactions between the agent (individual) and the social level of a multi-agent system, as represented in Figure 1. For example, considering Figure 1, first, on the left, we have an event occurring at the agent level causing an event at the social level. Such event at the social level, on the other hand, causes an event at the agent level. This situation can be observed in the semantic rule *AddComLink*, where an agent executes an action to change the organisational specification (at the social level) and this change reflects on the agent level, where all agent involved become aware of such changes. Second, we have an event at the agent level causing an event at the social level. This situation is exemplified in our semantic rule *SndCommit*, where an agent executes an action for sending a message with the performative `commit` and such action registers a social commitment at the social level. The last situation is when an event at agent level has no effect on the social level, for example when an agent receives a message, represented by the semantic rule *ReceivingCommit*.

## 4 Characteristics of the Multi-level Semantics

In this section we discuss, informally, some of the characteristics of the multi-level semantics representation. One of the most important characteristics of our multi-level semantics is *modularity*, where the treatment of each level for separated transition system allows us to clarify the modularity of the multi-agent system. In case a layer of the system needs to be changed, for example, the organisation, it is only necessary to modify the corresponding transition system in the semantic rules; the independent transition systems for the other levels remain unchanged.



Besides its modularity, we argue that multi-level semantics is *visually more elegant* and *understandable* for complex systems like multi-agent systems, and accordingly programming such semantic rules in frameworks like JaCaMo [5] is easier. This is a consequence of the fact that the multi-agent programming framework itself is based on such modularity, where the different levels of abstraction (agents, environment, and organisation), also called dimensions, are programmed by different platforms, following the different levels of the multi-level semantics we presented.

Another interesting characteristic is that our style for defining multi-level semantics allows representation of the components of the different levels of abstraction with *different granularity*, so it is possible to concentrate the semantic specification of the desired level, leaving the other levels more abstract, but still considering such other levels, thus maintaining the coherence and system representation as a whole.

## 5 Related Work

Dennis *et al.* [12] propose an *Agent Infrastructure Layer* (AIL) for BDI style programming languages. The aims in [12] are: i) to provide a common semantics basis for a number of BDI languages, and; ii) to support formal verification by developing a *model-checker* optimised for agent programs. The authors propose the design of AIL using an extensive operational semantics presented in [11]. The authors argue that AIL captures all major features of common BDI languages. The main goal of AIF is to provide a common (operational) semantics for a large fragment of BDI-style languages allowing the development of common verification tools and technologies.

Regarding communication aspects, Dennis *et al.* [12] describe that the semantics proposed by them covers the effects of a plan executing *send* actions and the *received* events. These aspects are similar to [28], where the action *send* is treated as the other actions and the receiving messages is treated through a specific event. Further, the authors of [12] state that the use of constraints allows one to filter out certain messages, handling notions such as the social acceptability.

In [9], the authors propose an operational semantics introducing the concept of modules. The authors argue that modularisation facilitates the implementation of agents, agents roles and agents profiles being an essential principle in structured programming and in agent programming in particular. Operational semantics is given to creating, executing, testing, updating, and realising module instances, i.e., module-related actions.

In [14], an operational semantics is given to cover the details of operations that may be applied to goals (dropping, aborting, suspending, and resuming goals). The authors argue that the semantics clarifies how an agent can manage its goals, based on the decisions that it chooses to make. The semantics, according to the authors, further provides a foundation for correctness verification of agent behaviour. The authors emphasise that the work contributes to the development of a rich and detailed specification of the appropriate operational behaviour when a goal is being pursued, has succeeded or failed, or has been aborted, suspended, or resumed by

the agent.

In [8], the author proposes an operational semantic framework for legal systems that are (structurally and operationally) situated in *agent societies*. The work uses operational semantics for modelling the *structure* and *dynamics* of legal systems.

The work presented in [8] is concerned with actions associated with legal systems (internal legal acts, external legal acts, social acts, etc.), i.e., under an *action-based dynamics* where the configuration transitions are determined by the performance of actions by the *legal organs* and *legal subjects* of the studied legal system. This is a clear example of work that already uses operational semantics to formalise multi-level systems.

Vieira *et al.* [28] have given operational semantics for speech-act based communication. In that paper, semantics is given for basic performatives that allow the communication between agents through simple message exchanges. In [28], communication is treated as single message exchanges and not as a sequence of interactions (i.e., a dialogue), as well as communication affects only the agent-level, i.e., the messages do not affect the social-level of the multi-agent system.

## 6 Final Remarks

In this paper we introduced a multi-level semantic representation of state transitions in multi-agent systems. Our approach takes in account the multi-level abstractions of multi-agent systems, so each transition represents changes at one level of such abstractions — the *agent-level* and *social-level* typically, although arbitrarily many levels can be formalised. We argue that this representation allows us to visualise more easily effects at multiple levels, for example effects of communication over not only individual agents but the system as a whole. This is useful given the current trend in agent techniques that affect the individual but with social implications, such as argumentation-based dialogues, normative systems, constitutive rules, etc.

Our multi-level semantics can be used to represent various forms of combined transitions occurring in such systems. For example, imagine that the multi-agent system receives externally a new norm which prohibits the execution of a specific action for the agents of a specific role in the system. Our style of formal semantics can be used to describe this transition at the *social level* (i.e., adding a norm in the normative system) which will cause a transition at the *agent level* in all agents of that specific role simultaneously (i.e., adding the belief indicating that specific action as prohibited).

As future work, we intend to use the multi-level transitions to ensure *semantics integrity* in multi-level systems, ensuring the coherence among the transitions with *integrity constraints* such as the “count-as” rules introduced by Searle in [25] and recently used in the multi-agent systems context for example in [10]. These ideas appear also in [5] where the authors declare the interest to explore the O-E (organisation–environment) connection in order to implement more advanced institution mechanisms such as “count-as” rules by Searle [25]. Further, we intend use this semantic style to specify formally our work in the multi-agent communication

field, where we are looking at implementing sophisticated dialogue frameworks in particular using argumentation-based techniques.

## References

- [1] Amgoud, L., N. Maudet and S. Parsons, *Modeling dialogues using argumentation.*, in: *ICMAS* (2000), pp. 31–38.
- [2] Austin, J. L., “How to do things with words,” Harvard University Press, 1962.
- [3] Bedi, P. and P. Vashisth, *Extending speech-act based communication to enable argumentation in cognitive agents*, in: *Advances in Computing, Communication and Control*, Berlin: Springer, 2011 pp. 25–40.
- [4] Boella, G., L. Van Der Torre and H. Verhagen, *Introduction to normative multiagent systems*, Computational & Mathematical Organization Theory **12** (2006), pp. 71–79.
- [5] Boissier, O., R. H. Bordini, J. Hübner, A. Ricci and A. Santi, *Multi-agent oriented programming with JaCaMo*, Science of Comp. Prog. **78** (2013), pp. 747–761.
- [6] Bordini, R. H., J. F. Hübner and M. Wooldridge, “Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology),” John Wiley & Sons, 2007.
- [7] Chopra, A. K. and M. P. Singh, *Agent communication*, in: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Second Edition, chapter 3*, MIT Press, Gerhard Weiss (ed), 2013 pp. 101–141.
- [8] da Rocha Costa, A. C., *Situated legal systems and their operational semantics*, Artificial Intelligence and Law **23** (2015), pp. 43–102.
- [9] Dastani, M. and B. R. Steunebrink, *Operational semantics for bdi modules in multi-agent programming*, in: *Computational Logic in Multi-Agent Systems*, Springer, 2010 pp. 83–101.
- [10] de Brito, M., J. F. Hübner and R. H. Bordini, *Programming institutional facts in multi-agent systems*, in: *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII*, Springer, 2013 pp. 158–173.
- [11] Dennis, L. A., *Agent infrastructure layer (ail): Design and operational semantics v1*, Technical report, 0. Technical Report ULCS-07-001, Department of Computer Science, University of Liverpool, 2007 (2007).
- [12] Dennis, L. A., B. Farwer, R. H. Bordini, M. Fisher and M. Wooldridge, *A common semantic basis for bdi languages*, in: *Programming Multi-Agent Systems*, Springer, 2008 pp. 124–139.
- [13] Dignum, V., *The role of organization in agent systems*, Handbook of Research on MAS: Semantics and Dynamics of Organizational Models (2009), pp. 1–16.
- [14] Harland, J., D. N. Morley, J. Thangarajah and N. Yorke-Smith, *An operational semantics for the goal life-cycle in bdi agents*, Autonomous agents and multi-agent systems **28** (2014), pp. 682–719.
- [15] Hubner, J. F., J. S. Sichman and O. Boissier, *Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels*, International Journal of Agent-Oriented Software Engineering **1** (2007), pp. 370–395.
- [16] McBurney, P. and S. Parsons, *Locutions for argumentation in agent interaction protocols.*, in: R. M. van Eijk, M.-P. Huget and F. Dignum, editors, *AC*, Lecture Notes in Computer Science **3396** (2004), pp. 209–225.
- [17] Panisson, A. R., A. Freitas, D. Schmidt, L. Hilgert, F. Meneguzzi, R. Vieira and R. H. Bordini, *Arguing about task reallocation using ontological information in multi-agent systems*, in: *12th International Workshop on Argumentation in Multiagent Systems (ArgMAS)*, 2015.
- [18] Panisson, A. R., F. Meneguzzi, M. Fagundes, R. Vieira and R. H. Bordini, *Formal semantics of speech acts for argumentative dialogues*, in: *13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2014, pp. 1437–1438.
- [19] Panisson, A. R., F. Meneguzzi, R. Vieira and R. H. Bordini, *Towards practical argumentation-based dialogues in multi-agent systems*, in: *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 2015.

- [20] Panisson, A. R., F. Meneguzzi, R. Vieira and R. H. Bordini, *Towards practical argumentation in multi-agent systems*, in: *2015 Brazilian Conference on Intelligent Systems, BRACIS 2015*, 2015.
- [21] Parsons, S. and P. McBurney, *Argumentation-based dialogues for agent co-ordination*, *Group Decision and Negotiation* **12** (2003), pp. 415–439.
- [22] Parsons, S., M. Wooldridge and L. Amgoud, *An analysis of formal inter-agent dialogues*, in: *1st Int. Conference on Autonomous Agents and Multi-Agent Systems* (2002), pp. 394–401.
- [23] Plotkin, G. D., *A structural approach to operational semantics* (1981).
- [24] Sadri, F., F. Toni and P. Torroni, *Logic agents, dialogues and negotiation: An abductive approach*, in: *In Proceedings AISB'01 Convention* (2001).
- [25] Searle, J. R., “The construction of social reality,” Simon and Schuster, 1995.
- [26] Singh, M. P., *Agent communication languages: Rethinking the principles*, *Computer* **31** (1998), pp. 40–47.
- [27] Singh, M. P., *A social semantics for agent communication languages*, in: *Issues in agent communication*, Springer, 2000 pp. 31–45.
- [28] Vieira, R., A. Moreira, M. Wooldridge and R. H. Bordini, *On the formal semantics of speech-act based communication in an agent-oriented programming language*, *J. Artif. Int. Res.* **29** (2007), pp. 221–267.