Knowledge-Level Integration for JaCaMo

Artur Freitas, Daniela Schmidt, Alison Panisson
Rafael H. Bordini, Felipe Meneguzzi and Renata Vieira

Pontifical Catholic University of Rio Grande do Sul - PUCRS
Postgraduate Programme in Computer Science, School of Informatics (FACIN)
Porto Alegre - RS - Brazil
{artur.freitas,daniela.schmidt,alison.panisson}@acad.pucrs.br,
{rafael.bordini,felipe.meneguzzi,renata.vieira}@pucrs.br

Abstract. The specification of agent systems comprises different dimensions normally defined using distinct formalisms. This lack of uniform representations makes it harder to express how each level affects the others. To address this problem, we propose a semantic representation that integrates the formalisms that originally cover a single multi-agent system dimension, such as agency, environment, or social organisation. In doing this, we align current trends in semantic technologies and in knowledge representation for agents, environments, and organisations thus providing MAS designers with a unified approach at the knowledge-level for the development of complex systems based on collaborative agents. Our approach uses three ontologies to represent the multi-agent system dimensions, and we exemplify how separate platforms operating at each dimension can be integrated. We discuss the implications of such integrated view for designing agents, and highlight advantages of semantic representations for agent-based software development.

Keywords: ontology, knowledge representation, semantic reasoning, multi-agent system, agent organisation, agent environment

1 Introduction

The use of ontologies in the context of Multi-Agent Systems (MAS) is still an open issue, especially in relation to integrated frameworks that consider the co-specification of different dimensions of a Multi-Agent System (MAS). A MAS developed in the JaCaMo [1] programming platform comprises three distinct dimensions, namely: agent, organisation, and environment. However, these dimensions are not uniformly integrated into a single formalism: agents are programmed in Jason [2] using the AgentSpeak language; organisations are specified in Moise [3] in an XML-based document; and environments are coded in Java using the CArtAgO API [4]. This approach makes it difficult to keep track of problems because errors in one level can affect the other levels, and it also becomes cumbersome to explore interconnections between the different layers. To address these issues, we claim for a unified representation which covers these three agent programming dimensions and integrates the various formalisms. In the literature, there are ontologies to represent the agency [5], environment [6], and organisation [7] levels of MAS; there is also a platform integrating these three levels in agent programming (*i.e.*, JaCaMo [1]), but the ontological level is not considered in that platform. Hence, we discuss an integrated semantic model to represent these three dimensions based on existing ontologies that represent each MAS level. To demonstrate the need for and advantages of such an integrated view, we refer to a classical agent programming problem known as “Gold Miners”.

This paper is structured as follows. Section 2 refers to previous ontologies related to MAS aspects: agent activities, organisations, and environments. Section 3 shows the need for the integration of such aspects at the knowledge level on the light of an example. Section 4 concludes this paper and points to our next steps.

2 Ontologies for Multi-Agent Systems

Ontologies are knowledge representation structures composed of concepts, relationships, instances and axioms, which empower the execution of semantic reasoners that provide functionalities such as *consistency checking*, *concept satisfiability*, *classification*, and *realisation*. It is natural to think that there are advantages in using ontologies more expressively in agent development. There are many ways in which these areas are connected and explored in the literature. In fact, JaCaMo dimensions have been considered at the knowledge level in previous work, as done for instance in [7]. However a multi-dimensional unified view has not been proposed. Next we focus on some examples of ontologies proposed for MAS, specially considering the distinct dimensions which divide JaCaMo.

Task is a very central concept for agency. The OWL-T task ontology [5] was designed to allow a formal and semantic specification of tasks using a high-level knowledge abstraction. Its central concept is *Task*, which can be hierarchically decomposed into simple or complex tasks. A simple task is an atomic or composite task, and a complex task consists of at least one simple task. In OWL-T, tasks contain inputs, outputs, preconditions, postconditions, preferences and effects [5]. This is the traditional view in AI regarding planning and action. This would correspond in JaCaMo to the AgentSpeak programming component, due to its relation to plans specification.

Environments also play an essential role in MAS, and their semantic representation can improve the way agents reason about the objects with which they interact and the overall environment where they are situated. In [6] an environment ontology is proposed based on environment aspects of agent programming technologies that is integrated into a platform for developing cognitive multi-agent simulations. Thus, it can be used to specify environments and derive a project-level, complete, and executable definition of multi-agent environments. An environment description is a specification of its properties and behaviour, which includes concepts such as: *objects*, also referred to as resources of the environment; *agents*, or more precisely, their “physical” representation in the environment that is visible to other agents; *actions* that each type of agent can perform in the environment; *reactions* of the environment and objects when an agent’s actions affect them; *perception types* available to each type of agent; and *observable properties*, that is, the information about the simulation to which observers (*e.g.*, the agents) have access. This concepts would have correspondence to CArtAgO.

Agent organisations are required to provide the means for agents to query and reason about the structure of a society of agents [7]. Among the recent developments on Moise, there is already a semantic description of multi-agent organisations [7], using OWL to develop an ontology for organisational specifications of the Moise model (structural, functional, and normative levels). This approach may help agents in becoming aware, querying, and reasoning about their social and organisational context in a uniform way [7]. The semantic description of Moise [7] provides agent-side reasoning and querying features (*i.e.*, the agents are able to use this information). The benefits highlighted in [7] are increased modularisation, knowledge enriching with meta-data, reuse of specifications, and easier integration. With the semantic web effort aiming to represent the information in semantic formats, the MAS community can take advantage of new semantic technologies in MAS development tasks such as to integrate organisational models, to monitor organisations, and to analyse agent societies [7].

3 Unifying the Three Dimensions

Fore each of the three dimensions described above we have found proposals for the semantic representations of their particular type of abstractions. For each of them the advantages of semantic web technologies have been advocated, and they usually recognise the importance of the other dimensions. However, a global view of MAS is still missing. We aim to work towards the integration of these various dimension at the semantic level, since they are already being integrated at the programming level (for example in JaCaMo [1]) and each dimension has had proposals for a semantic account.

Agent programmers would benefit from an integration among these ontological levels with each programming dimension since the knowledge represented in one dimension could be reused in another, resulting in a greater interoperability of agent platforms. Also, a system designed with a higher degree of modularity is easier to maintain, given that it separates different concerns yet enables relations between them. For example, the characteristics of one dimension (*e.g.*, environment) could be used to define properties on other dimension (*e.g.*, organisational). In fact, it is often the case that the concepts of one level are related to another but current MAS platforms do not allow for such relations to be explicitly represented.

Figure 1 shows an overview of the concepts in the three sub-ontologies: the environment dimension (based on the ontology proposed in [6]); the agent dimension (based on the OWL-T ontology [5]; and the social organisation dimension (based on the Moise ontology [7]). To further illustrate this global picture, we discuss an example (the Gold Miners problem) in the light of these three dimensions but from an unified perspective. The Gold Miners was proposed in the 2006 editions of the Multiagent Contest[1] and it is a well-known challenge for teams of multiple agents working in a dynamic environment.

The miner agent has plans to achieve for example the *handle_gold* goal, which can be decomposed into the sub-goals *search_gold*, *search_deposit* and *achieve_position*. Each of these sub-goals must be further decomposed into direct environment actions,

[1] http://multiagentcontest.org/2006

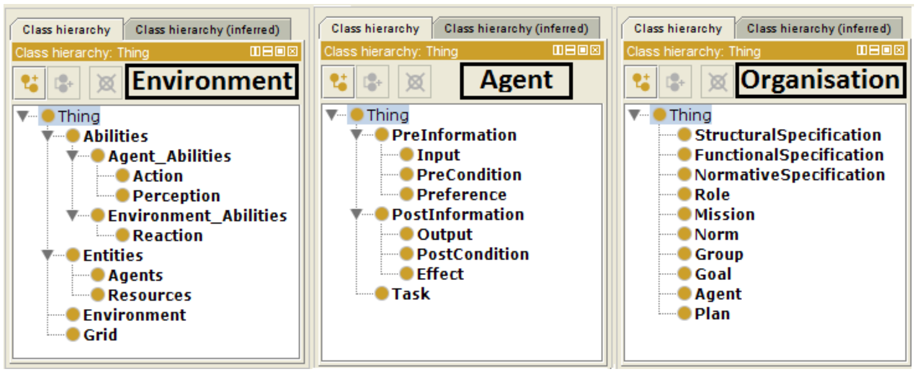

Fig. 1. Ontologies used to model MAS dimensions.

such as *pick*, *drop*, and *move*. These actions are instances of *Atomic Task*, *i.e.*, they can be directly executed by the agent in the environment. The *move* action causes the agent to change its position, while the *pick* action results in removing the gold from its location so that the miner agent can carry it, and, finally, the *drop* action causes the opposite (adding the piece of gold to a particular location in the environment).

In this Gold Miners problem, *pick* is a task instance. The *pick Task* has as *input* the *miner location* and the *gold location*. Its precondition is that the *miner location* is equal to the *gold location*. The postconditions of the *pick* task is the gold being carried by the agent that picked it and the gold no longer being available in the environment.

High level representation of tasks with which agents are involved is important in many aspects in MAS development. For instance, this global view of the an agent's course of action is helpful for the communication in a team of developers. However, current MAS development platforms do not have this relevant semantic layer, which can also become the bridge linking different development approaches. The similarities between elements of different paradigms for the agent dimension are the following: an *Action* in Jason corresponds to an *Atomic Task* in OWL-T; and a *Plan* in Jason (*i.e.*, a sequence of *Actions*) can be matched to the definition of a *Composite Task* in OWL-T. The alignment of these similarities is an important step towards the knowledge-level integration of agent dimensions.

Although actions have a strong connection with the environment, OWL-T [5] only addresses the abstraction of actions. Also, this level does not addresses questions such as which type of agent may perform different actions, *e.g.*, in Gold Miner the *leader* is not the type of agent that will be collecting gold in the field. To describe our scenario in more detail regarding the aspects of other dimensions, other ontologies have to be considered, as done next.

The actions modelled at the agent level can produce effects on the environment, and, therefore, they must be properly related with how actions are handled. Enhancing action representation to relate them to the environment and existing objects would be advantageous to achieve a better semantically integrated model of MAS to improve agent awareness and reasoning about the environment.

To model the Gold Miners problem in terms of the important aspects of the environment and artifacts, two types of *Agent* were considered: *Miner* and *Leader*; and three types of *Resources* were added: *Gold*, *Deposit*, and *Obstacle*. The object and data properties of the individuals of these concepts must be used to define their characteristics. Three types of *Action* concepts were included to represent the agent possible actions: *Drop*, *Pick*, and *Move*. Also, the *Environment* and *Grid* concepts were instantiated with one individual each for this MAS context. To create a semantic representation of this situation, new concepts must be instantiated according to the execution context in order to represent each agent, artifact, action, and so forth.

Although references are made to actions in this model, in the previous dimension, agency is taken into a more detailed account. But of course they are important for the environment representation since they are the elements responsible for the changes that occur in it. If we consider, as proposed in the action ontology, pre- and post-conditions might be directly related to the effects and conditions upon the environment. Future directions for CArtAgO have been presented in [4], which considers the use of ontologies for defining artifact operations, descriptions, and observable states. However, the semantic layers across different dimensions still have to be aligned.

In a group of agents acting collaboratively, it is of course very relevant to specify the group structure. At the organisational level, a MAS is defined by the social context of multiple agents, the roles that agents play, missions that each role must achieve, how groups are structured, types of communications available among agents, which norms are enforced to them, and so on. Based on the Moise ontology [7], and regarding the miners example we use here, the *Role* concept was instantiated with the *miner* and *leader* definitions. Unlike the previous dimension, where the actions that agents could perform in the environment were represented, this specification includes only the goals that agents playing specific roles must achieve. Therefore, the *FunctionalSpecification* states that *miners* must attempt to *find_gold*, *achieve_position*, and *handle_gold*, while *leaders* must have plans for achieving *assign_quadrant* and *allocate_miner*.

However, the organisational specification does not detail how agents will achieve these goals (that is defined at the agent level) and which actions they may or must perform. On the other hand, the model requires the definition of an *authority link* from *leaders* to *miners*, and a *communication link* between *miners*. The links represent what are the "socially acceptable" exchanges of messages among agent roles. The *Norms* for these roles were defined in such a way that *leaders* have an *obligation* to commit to the mission of managing *miners*, and the agents playing the role of *miners* are *prohibited* from executing *leader*'s missions. Moreover, following the organisational Moise ontology [7], the following concepts have to be instantiated with exactly one individual: *StructuralSpecification*, *FunctionalSpecification*, and *NormativeSpecification*. The concepts related to the plans and missions (in the *FunctionalSpecification*) must also be instantiated in this semantic representation.

With this well-known example (the Gold Miners), we intended to show the importance of the interrelation of these distinct levels, how complementary previous approaches are, and how crucial a unified view at the knowledge-level is for MAS development, which is being proposed by means of ontologies. Thus, we are exploring ways of integrating semantic technologies within MAS.

4 Final Remarks

The development of services that are based on collaborative agents requires a comprehensive view of a complex problem. This includes knowledge about the environment, the relations the agents establish among each other, and the common tasks that they have to deal with in a collaborative way. Unified MAS platforms such as JaCaMo [1] are being developed with the purpose of helping developers to build such complex solutions. However, such unification must happen ideally during the system design and at the knowledge level. Therefore, this paper presented an initial investigation towards the integration of agent-oriented programming platforms and ontologies. More specifically, we described the use of ontologies to streamline MAS development in JaCaMo.

The inclusion of ontology technologies in MAS is expected to bring together the power of knowledge-rich approaches and complex distributed systems. In terms of MAS design, such an integrated approach allows the design of a global conceptual view, and semantic tools make it possible to verify model consistency, perform inferences using semantic reasoners, and query instantiated models, all of which can contribute to a more principled way to develop multi-agent systems.

Acknowledgements

Part of the results presented in this paper were obtained through research on a project titled “Semantic and Multi-Agent Technologies for Group Interaction”, sponsored by Samsung Eletrônica da Amazônia Ltda. under the terms of Brazilian federal law No. 8.248/91.

References

1. Boissier, O., Bordini, R.H., Hübner, J., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Science of Computer Programming **78**(6) (2013) 747–761
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason. John Wiley & Sons (2007)
3. 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
4. Ricci, A., Viroli, M., Omicini, A.: CArtAgO: an infrastructure for engineering computational environments in MAS. In Weyns, D., Parunak, H.V.D., Michel, F., eds.: 3rd International Workshop “Environments for Multi-Agent Systems” (E4MAS). (2006) 102–119
5. Tran, V.X., Tsuji, H.: OWL-T: A task ontology language for automatic service composition. In: IEEE International Conference on Web Services (ICWS), IEEE Computer Society (2007) 1164–1167
6. Okuyama, F.Y., Vieira, R., Bordini, R.H., da Rocha Costa, A.C.: An ontology for defining environments within multi-agent simulations. In: Workshop on Ontologies and Metamodeling in Software and Data Engineering. (2006)
7. Zarafin, A.M.: Semantic description of multi-agent organizations. Master’s thesis, Automatic Control and Computers Faculty, Computer Science and Engineering Department – University “Politehnica” of Bucharest (2012)