Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

TOWARDS AN ONTOLOGICAL FOUNDATION OF AGENT-BASED SIMULATION

Giancarlo Guizzardi

Federal University of Espírito Santo (UFES)
Computer Science Department
Av. Fernando Ferrari
s/n29060-970 Vitória, Espírito Santo, BRAZIL

Gerd Wagner

Brandenburg University of Technology
Institute of Informatics
P. O. Box 101344
03013 Cottbus, GERMANY

ABSTRACT

A simulation model is intended to capture a real-world system. Consequently, the modeling language used for making the simulation model should have a "real-world semantics" guaranteeing some kind of ontological faithfulness for the models made with it. In this paper, we propose to use ABDESO, a foundational ontology for agent-based discrete event simulation, for evaluating agent-based simulation languages.

1 INTRODUCTION

In recent years, there has been a growing interest in the application of *foundational ontologies* (also known as *upper level*, or *top-level* ontologies) for providing real-world semantics for conceptual modeling languages, and methodological guidelines for evaluating and improving the models made using these languages. While the value of an ontologically well-founded conceptual modeling language is widely acknowledged in the areas of information system and software system engineering, as indicated by the great number of recent publications in this area, the issue of investigating the ontological foundations of simulation languages did not yet receive much attention in the scientific literature. This paper on the ontological foundations of agent-based simulation (ABS) is a follow-up of (Guizzardi and Wagner 2010a), which was concerned with the ontological foundations of basic discrete event simulation (DES).

While there are several research papers on the ontological foundations of organization and business process modeling (see Section 2), there has been no attempt yet to demonstrate the value of an ontologically well-founded modeling language for ABS. The main benefit obtained from establishing the ontological foundations of the core concepts of agent-based modeling languages is a clarification of their real world semantics. An ontological semantics of a simulation modeling language leads to a higher overall quality of the simulation models expressed in that language with respect to comprehensibility, maintainability, interoperability and evolvability.

We argue in section 3 that it is natural to consider the ABS paradigm as an extension of the DES paradigm. We, therefore, also speak of *agent-based DES* or, in short, ABDES.

In a series of publications (Guizzardi and Wagner 2004, 2005, 2010b; Guizzardi, Falbo and Guizzardi 2008) we have reported about our project for developing a foundational ontology called "UFO" (for *Unified Foundational Ontology*) by employing theories from Formal Ontology, Cognitive Psychology, Linguistics, Philosophy of Language, and Philosophical Logics. The core of UFO has been established through the development of an ontology of endurants by the first author in Guizzardi (2005). This foundational ontology has been successfully applied in the analysis of several important conceptual modeling constructs such as Roles, Types, Part-Whole Relations, Attributes, and Data types, among others.

After analyzing the ontological foundations of basic DES in Guizzardi and Wagner (2010a), using the UFO layers A and B (about objects and events), we discuss the ontological foundations of *agent-based* DES, using the UFO layer C about agents, in this article.

The remaining of this article is organized as follows. In Section 2, we discuss some related work. In Section 3, we briefly discuss the question of how ABS can be viewed as an extension of DES. Section 4 contains a summary of UFO, tailored to the purposes of this article, and of DESO, the discrete event system/simulation ontology that we have proposed in Guizzardi and Wagner (2010a). Then, in Section 5, we present ABDESO, an extension of DESO by adding agent-related ontological categories. Finally, in section 6, we use ABDESO for evaluating ABS modeling languages.

2 RELATED WORK

In Christley, Xiang and Madey (2004), using the Web ontology language OWL, an ontology defining an agent-based simulation framework is presented and possibilities for using OWL's automated reasoning capabilities are discussed.

In Livet et al. (2010), it is proposed to use ontologies (in the sense of conceptual domain models) for making the scientists' conceptual models more coherent with the simulation program code. This amounts to making an explicit conceptual model (using UML and/or OWL) before starting to code a simulation. However, although the paper refers to philosophical work on ontologies, foundational ontologies are not considered.

There is a large body of work, in which foundational ontologies are used for evaluating business process modeling languages, e.g., Green and Rosemann (2005). As an example of more recent work on investigating the ontological foundations of multi-agent systems, see Bottazzi and Ferrario (2009), in which the ontological modeling of organizations is discussed.

So, while there have been several proposals about how to use ontology engineering technologies, such as OWL, in ABS engineering, we were not able to find any work on the ontological foundations of ABS modeling languages.

3 DES AND ABS

In the history of computer simulation, following the success of object-oriented programming languages in computer science, classical DES technologies, where the system state is modeled in terms of plain variables, have been moving towards object-oriented DES technologies, where the system state is modeled in terms of objects and their attribute-value slots. Agent-based DES can be viewed as the next step in the evolution of DES.

In Chan, Son and Macal (2010) it is stated that "whether an ABS model is a discrete-event model or a hybrid of discrete and continuous depends on its state variables." They define an ABS model as "a hybrid discrete-continuous simulation model with proactive, autonomous, and intelligent entities." While this characterization clearly classifies ABS as an extension of classical DES, it is not very clear with respect to the terms "proactive," "autonomous," and "intelligent," which are adopted from Artificial Intelligence, but which lack a precise meaning (both philosophical and operational).

While there is widespread agreement in the literature that agents are *interactive systems*, other characterizations are more controversial and less clear. As interactive systems, agents interact both with their animate environment (their fellow agents) via message-based communication, but also with their inanimate environment, that is, with objects in their environment, via perceptions and state-changing actions.

4 SUMMARY OF UFO AND DESO

Since the development of UFO is an ongoing project, we use a simplified version of it, called *Essential Unified Foundational Ontology (eUFO)*, which restricts both the breadth and the depth of UFO, and simplifies its philosophical terminology, harmonizing it with informatics terminology as much as possible.

In this section, for making the present paper self-contained, we briefly summarize the base layer of eUFO, called eUFO-0, as well as its layer eUFO-*A* about substance individuals and trope individuals, and its layer eUFO-*B* about events. These layers of eUFO have been more extensively discussed in Guizzardi and Wagner (2010a).

eUFO-0 defines a number of basic ontological categories, as depicted in Figure 1 below, making a fundamental distinction between ***individuals***, which are things that exist in time and space in “the real world” and have a unique identity, and ***universals***, which are feature-based classifiers that classify, at any moment in time, a set of individuals with common features.

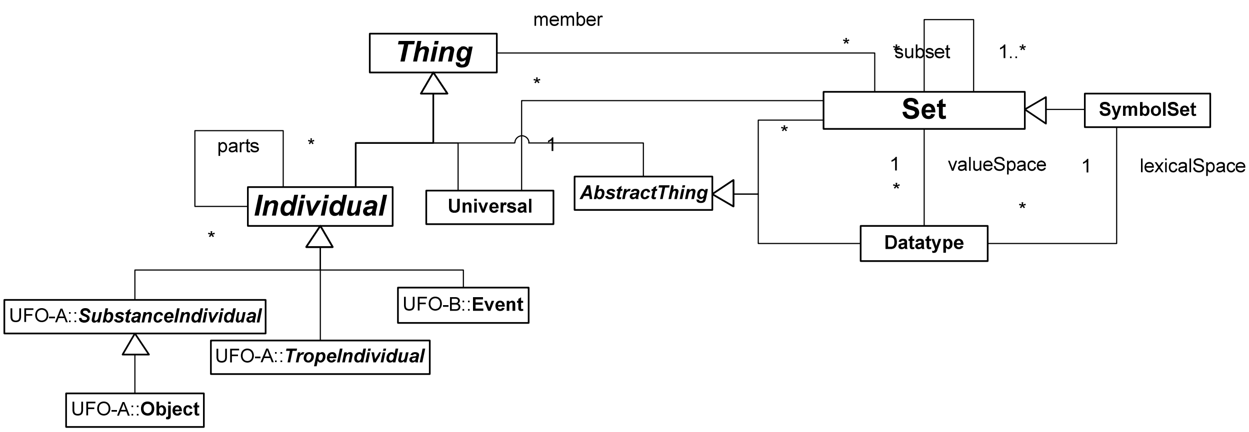

Figure 1: eUFO-0, the base layer of eUFO

We distinguish between three kinds of individuals: *substance individuals* (e.g., objects), *trope individuals* (e.g., attributions or beliefs) and *events*. As opposed to substance individuals, trope individuals can only exist in other individuals, i.e., they are *existentially dependent* on other individuals. The distinction between substance individuals and events can be understood in terms of their relationship to time. Substance individuals are wholly present whenever they are present, i.e., they *are in time*. Events *happen in time*, they may have temporal parts.

4.1 Substance Individuals, Attributions, Relationships and References

The ontology of substance individuals and trope individuals forms the UFO layer *A*. Examples of substance individuals are: the person with name “Gerd Wagner,” the moon, or an amount of sand. Examples of events are: today’s rise of the sun, my confirmation of an e-commerce purchase order through clicking the OK button, or the Second World War. Examples of trope individuals are: the redness of John’s T-shirt, Giancarlo’s employment with UFES, or my daughter’s belief in God.

There are two kinds of trope individuals: (a) *Intrinsic* trope individuals can be ***qualities*** such as an individualized color or a temperature, and ***modes*** such as a skill, a belief, or an intention; (b) *Relational* trope individuals or ***relators***: a medical treatment, a purchase order, or a social commitment. While qualities and modes depend on one single individual (their *bearer*), in which they *inhere*, relators depend on two or more individuals (their *relata*), which they *mediate*.

We distinguish between the color of a particular apple (as a *quality* of the apple) and the color *data value* that we associate with this quality in an ***attribution*** (with the help of an *attribute*). This data value is a member of the *value space* of the *data type* of the attribute. As an example, consider the attribute *hairColor*, which is applicable to persons, and associated to a data type with a lexical space consisting of color names. Then, the triple <john, hairColor, grey> represents an attribution that makes the sentence “The hair color of John is grey” true.

While a *formal relationship*, such as [Brandenburg is part of Germany], holds directly, for a ***material relationship***, such as [Paul is being treated in the medical unit M], to exist, something else, which *medi-*

ates the involved individuals (Paul and M), must exist. Such a mediating individual with the power of connecting individuals is called a *relator*. For example, a medical treatment connects a patient with a medical unit; an enrollment connects a student with an educational institution; a covalent bond connects two atoms. In general, relators are founded on events.

In a correspondence theory of truth (such as Tarski's semantics of predicate logic), attributions, references and relationships are considered as "truth makers" ("facts") that make corresponding sentences true.

4.2 Events

Events are individuals that may be composed of temporal parts. They *happen in time* in the sense that they may extend in time accumulating temporal parts. An event cannot exhibit change in time like a substance individual.

Events can be atomic or complex, depending on their mereological structure. Events existentially depend on their *participants* in order to exist. For instance, in the event of Caesar being stabbed by Brutus, we have the participation of Caesar himself, of Brutus and of the knife. Each of these participations is itself an event (an *object participation event*), which existentially depends on a single object. Special cases of object participation events are *object creation*, *object change* and *object destruction* events.

Events may change the real world by changing the state of affairs from a *pre-state* situation to a *post-state* situation. Each situation is determined by a set of associated *object snapshots* and a set of associated material relationships holding between the involved objects, where an object snapshot is a set of attributions and references with respect to a particular object.

4.3 Universals

Universals *classify* individuals, which are said to be their *instances*. The set of all instances of a universal is called its *extension*. We consider seven kinds of universals: *event types*, *object types*, *quality universals*, *attributes*, *relator universals*, *reference properties* and *material relationship types*. There are other kinds of universals, but these seven are the most relevant for conceptual modeling.

While the notions of attribute, relationship type and reference property are well-known in computer science in the area of information and database modeling, their ontological foundation in connection with quality universals and relator universals is not well-known. An *attribute* is a universal that is based on a quality universal, and that is associated with an object type as its domain, and with a data type as its range. A *material relationship type* is based on a relator universal, and is associated with two or more entity (object or event) types and zero or more data types. A *material reference property* is a binary material relationship type.

4.3.1 Quality Universals and Attributes

A ***quality universal*** classifies individual qualities of the same type. A quality universal can be associated with one or more data types, such that any particular quality corresponds to a specific data value from the value space of the data type. The association between qualities from some quality universal and the corresponding data values from an associated data type is provided by an ***attribute***, which is a universal that classifies *attributions*. A quality universal can be captured by one or more corresponding attributes, each of them based on a different data type.

E.g., the quality universal "hair color" could be captured by an attribute with the range of RGB byte triples or by an attribute with the range of natural language color names. Consequently, we may have more than one attribution for a particular quality, one for each associated attribute.

4.3.2 Relator Universals, Material Relationship Types, and Material Reference Properties

A relator universal classifies individual relators of the same type. The ***material relationship type*** *R* induced by a relator universal ***R*** classifies all material relationships induced by relators from ***R***. Since each material relationship corresponds to a tuple, *R* also has a *tuple extension* (i.e., a *relation* in the sense of set theory). A material relationship type is a universal that classifies material relationships, which are 'truth makers' for material relationship statements.

A ***material reference property*** represents a binary material relationship type, corresponding to a relator universal whose instances mediate exactly two individuals. Its tuple extension is a subset of the Cartesian product of the extensions of the two involved types. The first type is called the ***domain***, and the second one the ***range*** of the reference property.

4.4 DESO - A Foundational Ontology for Basic DES

We summarize DESO, a foundational Discrete Event System Ontology on the basis of eUFO, proposed in Guizzardi and Wagner (2010a).

4.4.1 The Run-Time Ontology DESO-I

In DESO-I, for simplicity, we assume that there are only binary relationships, which are represented by ***references*** specifying an object as the value of a reference property. An ***atomic fact*** is either a reference or an attribution, as depicted in Figure 2.

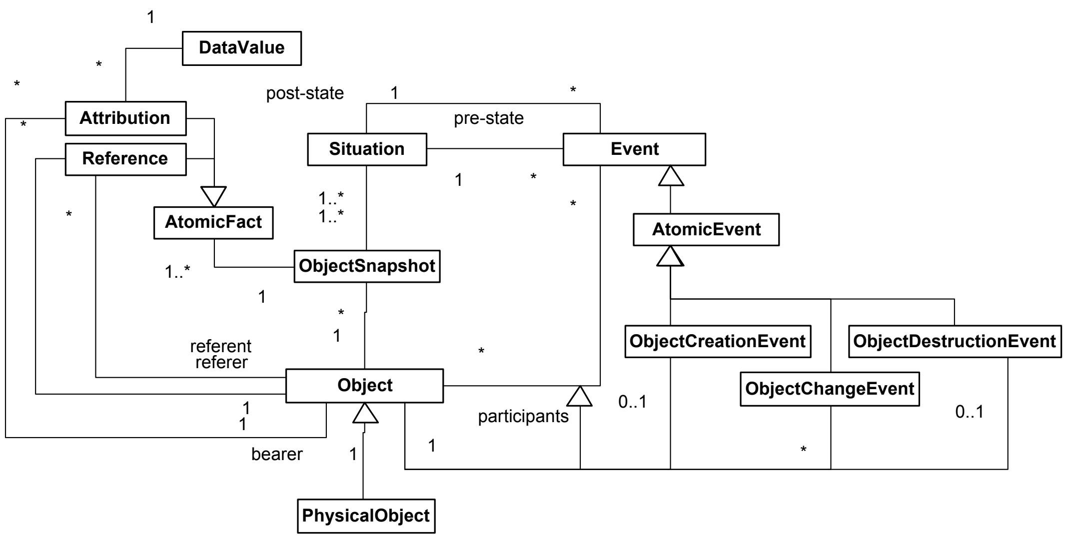

Figure 2: The categories of individuals, as defined by DESO-I.

Notice that in conceptual modeling, and in simulation modeling, we are not really interested to consider all the things that constitute a real-world system. We call those things, which we want to include in our models, *entities*, subsuming physical objects, events and certain material relationships. This choice implies that we do not want to include more abstract concepts, like relators or qualities, in a simulation model.

Entities are classified with the help of *entity types*, as explained below in Subsection 4.4.2. Entity types allow describing the entities classified by them with the help of *attributes* and *reference properties*. Since we also want to be able to describe certain material relationships in this way, it is natural to subsume them under the concept of entities.

4.4.2 The Design-Time Ontology DESO-U

At design-time, we describe a discrete event system by defining the various entity types, the instances of which are part of the running system. In addition to the base concept *data type* from eUFO-0, also the concepts of (physical) object types, attributes and reference properties from eUFO-U are needed in DESO for allowing to represent abstractions of discrete event systems. In particular, the concepts of *object types* and *event types* are needed. Being special kinds of entity types, both object types and event types have *attributes* and *reference properties*, as depicted in Figure 3.

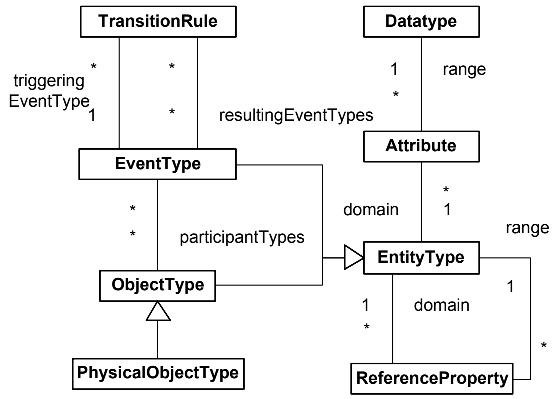

Figure 3: The categories of types, as defined by DESO-U

5 ABDESO - A FOUNDATIONAL ONTOLOGY FOR ABS

We now define ABDESO, our *agent-based discrete event system ontology*, which extends DESO by adding the concept of *agents* and other concepts related to *agency*. Clearly, agents are special objects. E.g., we may want to consider not only human beings, but also all kinds of living beings, including insects and bacteria, as agents. We may want to be even more inclusive, and not limit the applicability of our agent concept to biological systems, but possibly also allow certain artificial systems (such as robots) or social systems (such as organizations) to be considered as agents. On the other hand, we want to exclude all kinds of passive objects, such as chairs, apples and mountains, from our concept of an agent. So, what is common to living beings, robots and social systems? We claim that all these objects are *interactive systems* that are able to *interact* with passive objects in their environment or with each other in a *purposeful* way. The question what constitutes *interaction* is closely related to the question of *what is an action*.

In philosophy, this question has been approached by asking how to distinguish "*the things that merely happen to people − the events they undergo − and the various things they genuinely do,*" as Wilson (2007) has put it. While there has been a strong tradition in philosophy to require that a 'genuine' action is based on an *intention*, Wilson (2007), by referring to Frankfurt (1978), states that "*the purposeful behavior of animals constitutes a low-level type of 'active' doing. When a spider walks across the table, the spider directly controls the movements of his legs, and they are directed at taking him from one location to another. Those very movements have an aim or purpose for the spider, and hence they are subject to a kind of teleological explanation.*" He concludes that "*this behavioral activity is 'action' in some fairly weak sense.*"

Thus, we will define *actions* to be those events that are the direct result of the *purposeful behavior* of an *interactive system*. Notice that this definition does not exclude higher-level action concepts such as *intentional actions*, which we just consider to be a special case of our general action concept. So, we do not require an agent to have a mental state with beliefs and intentions, as it is common in philosophical theories of humans as cognitive subjects, and also in many Artificial Intelligence approaches to multi-agent systems, in particular in the popular *Belief-Desire-Intention* (BDI) approach, which was originally a modal logic approach, but has later been popularized to stand for any kind of mentalistic agent architecture. Simple agents may have beliefs, but they don't have intentions to carry out certain actions for achieving some goal.

Without going into any deeper discussion of the philosophical issues involved, we would like to point out that for the purpose of establishing the foundations of agent-based simulation, it may be also helpful to consider agents as *intentional systems* in the weak sense of Dennett's theory of the *intentional stance* (Dennett 1996), in which one uses mental attitudes for describing and predicting the behavior of interactive systems without assuming that they really possess these attitudes.

5.1 An Ontology of Simple Agents

The goal of our ontological theory of agents is to characterize interactive systems as special objects that are distinct from passive (physical) objects, no matter if an interactive system can be considered intentional or not. It is obvious that we have to include the concepts of *perception* and *action* in our account of interactive systems. We include both of them as special kinds of events, so we speak of ***perception events*** and ***action events***, which are categories of eUFO's layer C1 about simple agents, depicted in Figure 4.

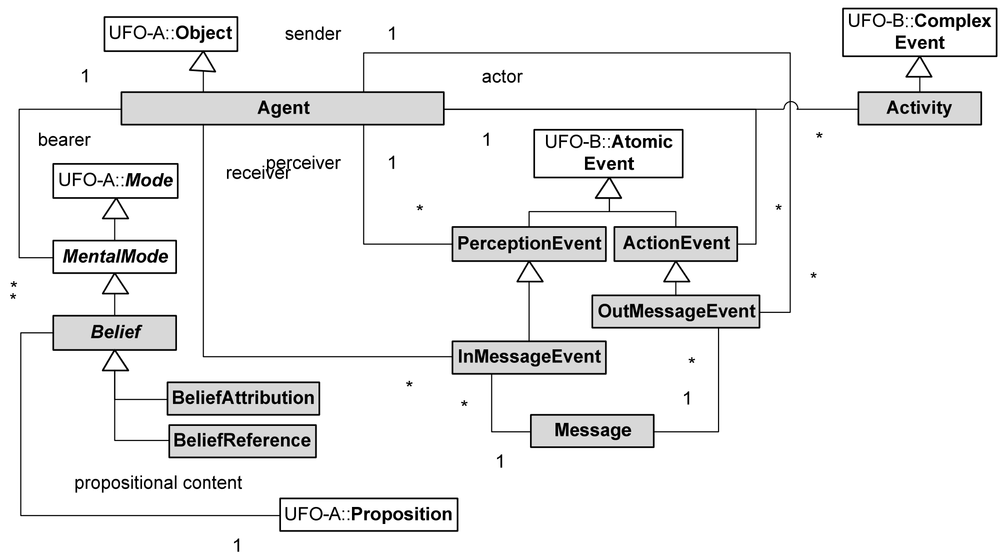

Figure 4: The categories of eUFO-C1, with ABDESO concepts in grey

For being able to model communication as a special kind of interaction between agents, we introduce the concepts of ***messages*** and ***communication events***, as depicted in Figure 5. A *communication event* is a complex event, associated with a *sender* and one or more *receivers*. It binds an ***out-message event*** to one or more ***in-message events*** (one for each receiver), sharing a common outgoing and incoming message. As shown in Figure 4, out-message events are action events, while in-message events are perception events.

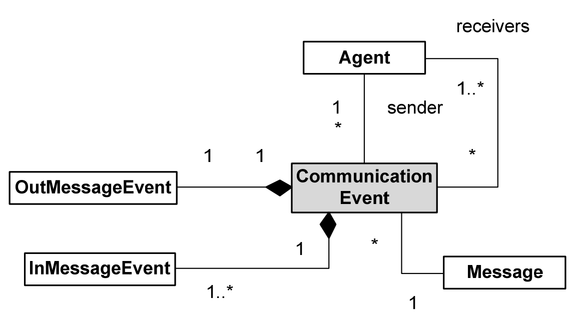

Figure 5: Communication events are complex events

The influence of actions, and other events, on the perceptions of an agent is given by the causal laws of the agent's environment, taking the form of *transition rules* (see DESO-U in Figure 3), which determine the caused perception events.

The influence of perceptions on the actions of an agent is given by its reactive behavior, which is based on behavior patterns in the form of ***reaction rules***, shown in Figure 6. A perception event may lead, via triggering a reaction rule, to a resulting action of the agent in response to the event, and/or to an update of the agent's information state, typically in the form of a ***belief*** change specified by the post-condition of the rule. A belief has a *proposition* as its ***propositional content***. In ABDESO, a belief is either a ***belief attribution*** or a ***belief reference***, the propositional contents of which are *attributions* or *references* (i.e., atomic facts). While beliefs represent propositional information, the information state of an agent may also include various forms of non-propositional information, such as pictorial information.

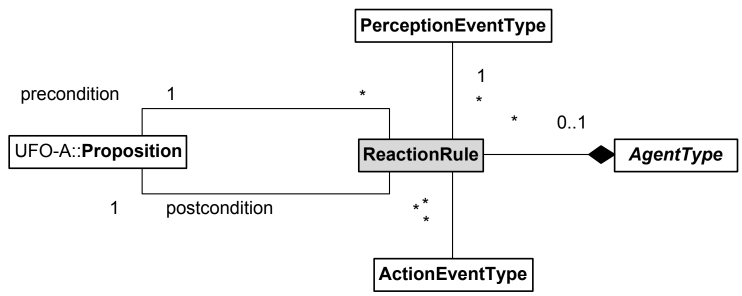

Figure 6: The concept of reaction rules

We assume that beliefs are expressed as *belief statements* in the agent's belief representation language (e.g., in the form of *belief attribution triples* expressed with the help of *belief attribute* names and *belief reference triples* expressed with the help of *belief reference property* names). In general, however, as discussed in Schwitzgebel (2010), there may be some form of beliefs even in simple agents without language.

In many agent-based simulation models, there is no need to handle true beliefs, which are independent of the corresponding facts. Instead of maintaining a storage-inefficient beliefs and facts duplication structure, the simulator may give agents direct access to those facts they need to know. This means that beliefs are identified with, or short-circuited to, corresponding facts. We call the agents of a simulation model with this feature *perfect information agents*.

We claim that these eight concepts depicted in Figures 4, 5 and 6: *beliefs*, *perception events* and *action events*, *messages* and *communication events*, *in-message events* and *out-message events*, and *reaction rules*, form the foundation of an ontological account of agents as interactive systems, no matter if agents are intentional or not. Simple agents have just these components, or even a subset thereof (e.g., they may do without beliefs).

5.2 An Ontology of Cognitive Agents

Cognition is based on perception and beliefs, but it may include further components, such as goals and intentions, as depicted in Figure 7 about the intentional concepts of ABDESO.

Both beliefs and goals have a propositional content. An intention is associated with a goal and a plan for achieving that goal, where a plan consists of a partial order of action rules, each having a pre-condition, a post-condition, and zero or more action event types. A plan is executable, if the pre-conditions of the first group of action rules hold in the current belief state of the agent. The sequence of action rules of a plan is constructed in such a way that the post-conditions of the preceding rules imply the pre-conditions of the succeeding rules. When executing a plan, an agent may notice that the plan is no longer executable, if the pre-condition of some rule to be executed does not hold. In that case, the agent has to make a new plan for the same goal.

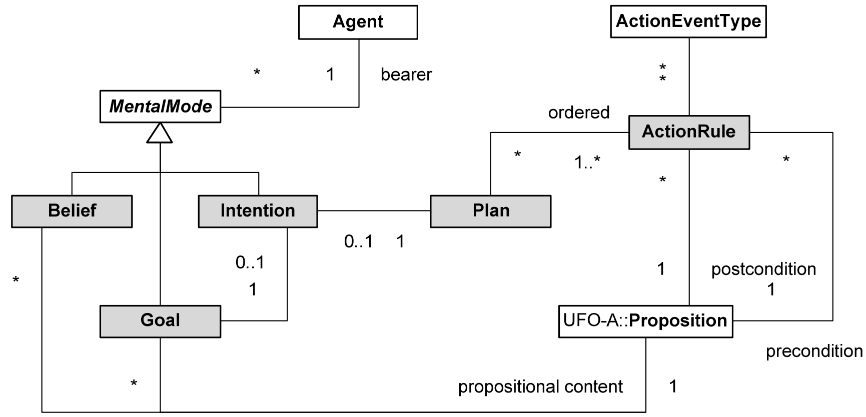

Figure 7: The constructs of eUFO-C2, including the intentional concepts of ABDESO in grey

5.3 An Ontology of Institutional Agents

An ***institutional agent***, as depicted in Figure 8 about the eUFO layer C3, may define ***roles*** to be played by its subagents, while an ***organization***, as a special kind of institutional agent, may also define ***positions***, which aggregate roles, for its human subagents. These roles and positions, as agent types, define additional properties and additional behaviors for their instances. The overall reactive behavior of a subagent of an organization results from merging all reaction rules associated with the agent's base type, the positions held by it and the roles played by it. Since positions and roles can be assumed and dropped at runtime, this leads to a dynamic behavior definition for agents.

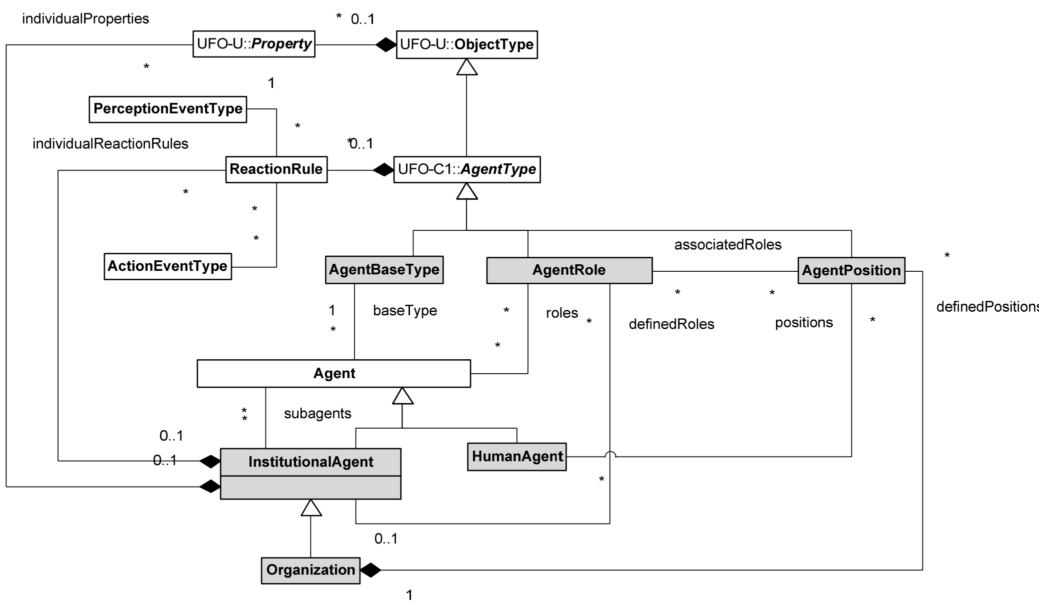

Figure 8: The constructs of eUFO-C3, including the institutional concepts of ABDESO in grey

6 EVALUATING AGENT-BASED SIMULATION LANGUAGES

ABS languages can be evaluated by comparing a representation of their concepts, typically provided by a metamodel of the language, to an ontology of agent-based discrete event systems (more precisely, to an ontology that represents a shared conceptualization of the domain of agent-based discrete event systems). The stronger the match between them, the easier it is to communicate and reason with models made in that language. In order for an ABS model *M* to faithfully represent an agent-based discrete event system abstraction *A*, the simulation language *L* used to make *M* should faithfully represent the conceptualization of agent-based discrete event systems used to conceive the abstraction *A*.

Since ABDESO represents a general conceptualization of agent-based discrete event systems, it can serve as a reference ontology for evaluating the simulation languages of ABS frameworks. For any given ABS language *L*, we may consider (1) a *representation* mapping from the concepts of ABDESO to the elements of *L* and (2) an *interpretation* mapping from the elements of *L* to the concepts of ABDESO. If these mappings are far from being isomorphisms, this may indicate deficiencies (soundness and completeness problems) of *L*. In Guizzardi and Wagner (2010a) we have defined the following measures: *soundness*, *completeness*, *lucidity* and *laconicity*, each with a degree. The lower these degrees are for a given ABS language, the more problems may be expected from using a model expressed in it, e.g., by communicating incorrect information and inducing the user to make incorrect inferences about the semantics of the domain.

We now apply the described method for providing a brief evaluation of the soundness and lucidity of the ABS system *Brahms*. In an extended version of the paper, we plan to include also an evaluation of its completeness and laconicity, and to apply the method also to the ABS systems *RePast* and *NetLogo*.

6.1 Evaluating the Soundness and Lucidity of Brahms

Brahms (Sierhuis 2001) is an agent-based modeling and simulation environment a) for developing simulations of people, organizations, and objects such as tools, documents and systems and b) for designing, simulating and implementing multi-agent software systems. The following paragraphs about Brahms summarize relevant parts of the Brahms online tutorial (Acquisti et al 2007).

Brahms makes a difference between "animate-intentional-objects" referred to as *agents* and "inanimate-unintentional-objects" referred to as *objects*. An agent "represents a person or, more inclusively, an interactive system." Objects are supposed to be able to represent various kinds of entities including artifacts (such as computers and other kinds of machines), which are considered as "unintentional action-oriented systems." The Brahms tutorial states that "There might be occasions when the intentional stance is appropriate for objects. When this is the case, we might decide to represent a machine as an agent. For example, […] ATM machines and Bank computers might be modeled as agents."

The key features of objects are 'class instantiation,' 'facts,' 'activities,' and 'workframes,' which together represent the state and behavior of objects. Some objects may have internal states, such as information in a computer, that are modeled as 'beliefs.' Classes in Brahms represent object types. They define the attributes, relations, activities and workframes, initial-facts, and initial-beliefs for the instances (objects) of that class.

A fact is "a first-order predicate statement about the world." Any agent can detect a fact in the world and turn it into a belief. A fact is global, and can be "acted on (in the case of objects) or detected (in the case of agents)." Activities represent "real-life actions." They are executed by 'workframes,' which are "situation-action rules." Primitive activities are atomic actions, and a small number of primitive activities are defined to have built-in semantics that is implemented in the Brahms engine. These predefined primitive activities exist to communicate beliefs, create runtime objects, and travel to a location.

The key features of agents are 'group membership,' 'beliefs,' 'workframes,' 'thoughtframes,' and 'location.' A group can "represent one or more agents, either as direct members or as members of subgroups." Typically, 'activities' are associated with groups, so that a group "represents a group of individuals playing a particular role in an organization." A group defines "attributes, relations, initial-beliefs, initial-facts, activities, workframes and thoughtframes" for its members. A belief is "a first-order predicate statement about the world, which is local to an agent, i.e., only the agent can access its beliefs, and no other agent can." "Agents act based on their beliefs." A belief held by an agent may differ from the corresponding fact or a belief that another agent has about the same fact. Thoughtframes, which are production rules, are used as “the agent's inference rules.”

Table 1 lists the elements of the language of Brahms together with their interpretation in ABDESO.

Table 1: The ABDESO interpretation of Brahms elements

Brahms element	***ABDESO interpretation***	***Brahms element***	***ABDESO interpretation***
Object	Agent, object	Activity	Activity
Class	Object type	Workframe	Reaction rule
Attribute	Attribute	Agent	Agent
Relation	Reference property	Group	Agent type
Fact	Fact triple	Thoughtframe	Reaction rule
Belief	Belief triple		
Soundness (Lucidity)			***100% (90%)***

As we can see from this table, the language of Brahms is 100% sound w.r.t. ABDESO, since all elements of it have an ABDESO interpretation. It is 90% lucid w.r.t. ABDESO, since one out of 11 elements (namely *Object*) have more than one DESO interpretation.

7 CONCLUSIONS

We have defined the *Agent-Based Discrete Event Simulation Ontology (ABDESO)*, extending the *Discrete Event Simulation Ontology (DESO),* and derived from the *Unified Foundational Ontology (UFO)*. ABDESO defines the basic concepts that need to be supported in any general-purpose ABS language. In future work, we plan to evaluate other established ABS frameworks (such as *RePast* and *NetLogo*) by analyzing the representation mapping from ABDESO to these languages, and the interpretation mapping from these languages to ABDESO.

REFERENCES

Acquisti, A., B. Clancey, R. J. J. van Hoof, M. Scott, and M. Sierhuis. 2007. Brahms Tutorial. Version 1.0. Accessed March 31, 2011. http://www.agentisolutions.com/documentation/tutorial/tt_title.htm.

Bottazzi, E., and R. Ferrario. 2009. “Preliminaries to a DOLCE Ontology of Organizations.” *International Journal of Business Process Integration and Management* 4(4):225–238.

Chan, W. K. V., Y.-J. Son, and C. M. Macal. 2010. “Agent-Based Simulation Tutorial – Simulation Of Emergent Behavior and Differences Between Agent-Based Simulation and Discrete-Event Simulation.” In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 135–150. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Christley, S., X. Xiang, and G. Madey. 2004. “An Ontology for Agent-Based Modeling and Simulation.” *Proceedings of the Agent 2004 Conference*.

Dennett, D. C. 1996. *The Intentional Stance.* 6th printing. Cambridge, Massachusetts: The MIT Press, (first published 1987).

Frankfurt, H. 1978. “The Problem of Action.” *American Philosophical Quarterly* 15:157-62.

Green, P., and M. Rosemann. 2005. *Business Systems Analysis with Ontologies*. Idea Group Publishing.

Guizzardi, G. 2005. “Ontological Foundations for Structural Conceptual Models.” PhD Thesis, University of Twente, The Netherlands.

Guizzardi, G., R. A. Falbo, and R. S. S. Guizzardi. 2008. “Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE Software Process Ontology.” In *XI Ibero-American Workshop on Requirements Engineering and Software Environments (IDEAS’2008)*, Recife.

Guizzardi, G., and G. Wagner. 2004 . "A Unified Foundational Ontology and some Applications of it in Business Modeling." In *Proceedings of the CAiSE'04 Workshops,* edited by J. Grundspenkis and M. Kirikova, 3:129–143. Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia. June 7–11, 2004.

Guizzardi, G., and G. Wagner. 2005. "Towards Ontological Foundations for Agent Modeling Concepts using UFO." In *Agent-Oriented Information Systems (AOIS)*, selected revised papers of the Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems 2005. Lecture Notes in Computer Science, 3508:110–124. Berlin/Heidelberg: Springer.

Guizzardi, G., and G. Wagner. 2010a. "Towards an Ontological Foundation of Discrete Event Simulation." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan and E. Yücesan, 652–664. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. Accessed April 1, 2011. http://www.informs-sim.org/wsc10papers/059.pdf.

Guizzardi, G., and G. Wagner. 2010b. "Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages." In *Theory and Application of Ontologies,* edited by R. Poli, 175-196. Berlin/Heidelberg: Springer.

Livet, P., J.-P. Müller, D. Phan, and L. Sanders. 2010. "Ontology, a Mediator for Agent-Based Modeling in Social Science." *Journal of Artificial Societies and Social Simulation* **13**:1. Accessed April 9, 2010 http://jasss.soc.surrey.ac.uk/13/1/3.html.

Sierhuis, M. 2001. "Modeling and Simulating Work Practice. BRAHMS: a multiagent modeling and simulation language for work system analysis and design." Ph.D. thesis, Social Science and Informatics (SWI), University of Amsterdam, SIKS Dissertation Series No. 2001-10, Amsterdam, The Netherlands, ISBN 90-6464-849-2.

Schwitzgebel, E. 2010. "Belief." In *The Stanford Encyclopedia of Philosophy,* edited by E. N. Zalta. Winter 2010 Edition.
<http://plato.stanford.edu/archives/win2010/entries/belief/>

Wilson, G. 2009. "Action." In *The Stanford Encyclopedia of Philosophy,* edited by E. N. Zalta. Fall 2009 Edition. Accessed April 9, 2011. http://plato.stanford.edu/archives/fall2009/entries/action/.

AUTHOR BIOGRAPHIES

GIANCARLO GUIZZARDI is Associate Professor at the Computer Science Department, Federal University of Espírito Santo (UFES), Brazil, and senior member of the Ontology and Conceptual Modeling Research Group (NEMO). He has been working for about fourteen years in the development of Domain and Foundational Ontologies and their application in computer science and, primarily, in the area of Conceptual Modeling and Organizational Modeling. His email address is gguizzardi@inf.ufes.br.

GERD WAGNER is Professor of Internet Technology at the Department of Informatics, Brandenburg University of Technology, Germany. His research interests include agent-oriented modeling and agent-based simulation, foundational ontologies, (business) rule technologies and the Semantic Web. In recent years, he has been focusing his research on the development of an agent-based discrete event simulation framework, called *ER/AOR Simulation* (see www.AOR-Simulation.org) He can be reached at http://www.informatik.tu-cottbus.de/~gwagner/.