

# Applying the UFO Ontology to Design an Agent-oriented Engineering Language

Renata S.S. Guizzardi, Giancarlo Guizzardi

Ontology and Conceptual Modeling Research Group (NEMO)  
Federal University of Espírito Santo  
Av. Fernando Ferrari, S/N, 29060-970, Vitória/ES, Brazil  
[rguizzardi, gguizzardi]@inf.ufes.br

**Abstract.** The problem of designing suitable conceptual modeling languages for system engineering is far from being solved. In the past years, some works have proposed the use of foundational ontologies as analysis tools to enable semantic coherence when (re)designing such languages. In this paper, we exemplify this approach by applying a foundational ontology named UFO in the design of an agent-oriented modeling language for the ARKnowD methodology. Instead of proposing new concepts and constructs, ARKnowD relies on existing work, combining two different approaches, namely Tropos and AORML. Each work is applied in a different development activity, according to their natural propensity: Tropos in Requirements Analysis and AORML in System Design. Besides the ontological approach, we propose some mapping rules between the notations, inspired in the Model Driven Architecture (MDA) metamodel transformation method. This approach helps to guarantee a smooth transition from one activity to the other.

**Keywords:** conceptual modeling languages design; foundational ontologies, agent-oriented engineering language, model-driven architecture.

## 1 Introduction

The problem of designing suitable conceptual modeling languages for system engineering is far from being solved. This is a complex matter because, on the one hand, one expects a language which is expressive enough to capture the important aspects of the particular domain in which the system is to be inserted. And on the other hand, one also wants this language to be accessible and provide the right level of abstraction to enable effective analysis and communication with stakeholders. These requirements are often contradictory and hard to achieve.

In the past years, some works have proposed the use of foundational ontologies as analysis tools to enable semantic coherence when (re)designing such languages. Foundational ontologies may be understood as formal systems of domain-independent categories that can be used to characterize the different modes of *existence* and, thus,

can be used to characterize the most general aspects of concepts and entities that belong to different domains in reality.

In this paper, we exemplify this approach by applying a foundational ontology named UFO in the design of an agent-oriented modeling language for the ARKnowD (Agent-oriented Recipe for Knowledge Management System Development, read “Arnold”) methodology [1]. Given the current stage of research on the agent-oriented paradigm, and the vast availability of methodologies and languages for agent-oriented analysis and design, the methodology presented here is built over existing work. It is our belief that not one methodology possesses all the right characteristics to be applied in a particular domain and/or situation. Instead, these characteristics can often be attained by combining different approaches. ARKnowD explores the combination of Tropos [2] and AORML[3]. Each work is applied in a different development activity, according to their natural propensity: Tropos in Requirements Analysis and AORML in System Design.

The main idea behind the application of UFO regards interpreting the concepts of the languages applied in ARKnowD (i.e. the Tropos’s notation and AORML) in terms of the concepts of the ontology. Having understood that, we may simply assume that the concepts of Tropos and AORML which map to the same concept of UFO are equivalent. This has both theoretical and practical implications. For instance, it is common to promote some redesign in the languages due to the deeper analysis provided by the ontology. This analysis enables one to understand better how such languages should model specific domains, which often leads to introducing or suppressing concepts from the original language. In practice, this leads to distinctions in the designed models. Hopefully, these distinctions are not too many, so as to justify the use of this particular language. Such changes generally result in engineering models which are clearer to understand and communicate (thus addressing the requirements mentioned in paragraph one).

Besides the ontological approach, we propose some mapping rules between the notations, inspired in the Model Driven Architecture (MDA) metamodel transformation method [4]. This guarantees a smooth transition from Requirements Analysis to System Design, guiding the developer on the use of the methodology, and facilitating automatic model transformation from one activity to the other.

The focus of this particular paper is to illustrate our approach by: (i) describing the ontological interpretations of Tropos and AORML and (ii) presenting the mapping rules which enable guidance to the designer in producing a detailed design model, whose draft is automatically mapped from the system’s architectural model (section 4). Before these core sections, section 2 describes the ontological approach applied to design ARKnowD’s language and section 3 provides introductory information on Tropos and AORML, also discussing why these two approaches are appropriate to engineer KM systems. Complementarily, section 5 illustrates the transformation of a Tropos diagram into an AORML diagram, following the proposed mapping rules; finally, section 6 presents some final considerations.

## 2 Using Foundational Ontologies to Analyze, (re)Design and Combine Conceptual Modeling Languages

Ontologies are recognized as important conceptual tools in Computer Science since the end of the 60s, especially in the areas of conceptual modeling and artificial intelligence [5]. In the past years, we observed an explosion of works related to ontologies in several scientific communities. This is motivated by the potential of ontologies to solve semantic interoperability problems (e.g. application and database integration).

An important point to notice is the difference in meaning of the term “ontology” when used, on the one hand, by the conceptual modeling community and, on the other hand, by the artificial intelligence, software engineering and semantic web communities. In conceptual modeling, the term is used in accordance with its original definition in philosophy, i.e. as a formally and philosophically well-founded model of categories that can be used to articulate conceptualizations in specific engineering models and knowledge domains. Conversely, in the other areas mentioned above, the term ontology has been used to describe: (i) a concrete engineering artifact, designed to serve a specific function, without (or with minimum) concern to theoretical foundational aspects; or (ii) domain models (e.g. biology, finance, logistics etc.) expressed in a knowledge representation language (e.g. RDF, OWL, F-Logic or conceptual modeling language (e.g. UML, EER, ORM).

With respect to the analysis and (re)design of conceptual modeling languages (i.e. the focus of this particular paper), we must understand ontology as in conceptual modeling, i.e. as a theoretical body of knowledge or foundation (that is why we call it *foundational ontology*). Using this foundational ontology as a reference model enables the evaluation, comparison, and identification of correspondences between different modeling languages, in other words, UFO is employed here as a well-founded basis for (1) making explicit the ontological commitments of each modeling language; (2) defining (ontological) real-world semantics for their underlying concepts; (3) providing guidelines for the correct use of these concepts; (4) relating concepts defined in different languages via their ontological semantics. The adequacy of this ontology for our purposes lies on the fact that an important part of UFO (named UFO-C) includes the concepts that are relevant to engineer systems in this particular domain, i.e. knowledge management. More about how UFO-C has been developed may be found in [1].

Finally, we would like to highlight the fact that Ontology-Based approach for combining modeling languages such as the one employed here should not be seen in opposition to a Model-Driven one. Typically, in the former, languages are related either by producing a merged metamodel (using a language such as OMG’s MOF), or by defining a transformation model (using a language such as OMG’s QVT), which relates the constructs of the complementary languages. Now, one should bear in mind that, in any case, language interoperability is first and foremost a *semantic interoperability problem*. Hence, before we can define a set of transformation rules mapping constructs from a metamodel A to a metamodel B, we must establish the relationship between these constructs. But these relations can only be discovered once we know the relationship between their respective interpretations, i.e., between the

elements in the underlying domain conceptualizations which are represented by them. In summary, as demonstrated in this article, the Ontology-Based and Model-Driven approaches can be seen as complementary: while the former focuses on semantic aspects of language interoperability, the latter focuses on syntactic ones.

### **3 ARKnowD Methodology: Combining Tropos and AORML**

ARKnowD's life cycle is composed of four activities, namely requirements elicitation, requirements analysis, architectural design and detailed design. These activities may be iteratively executed up to the point that the solution is modeled in enough detail to enable implementation. Tropos is applied in the first three activities while AORML covers the fourth one. More about the lifecycle of this methodology and detailed guidelines on how to proceed to apply it can be found in [1].

#### **3.1 Tropos**

The Tropos methodology [2] uses visual modeling language and a set of techniques for goal analysis. Basic constructs of the conceptual modeling language are: *actor*, representing a stakeholder in a given domain, a *role* or a set of roles played by an actor in a given organizational setting, and actor's *goal*, *plan* and *resource*. Moreover, a *dependency* link between pairs of actors allows to model the fact that one actor depends on another in order to achieve a goal, execute a plan, or acquire a resource. Goal analysis is conducted from the point of view of each individual actor; i.e. for each actor's goal, we may consider: means to satisfy it (*means-end relationship*); alternative ways to achieve it (*OR decomposition*); possible sub-goals (*AND decomposition*); goals or plans or resources that can contribute positively or negatively to its achievement (*contribution*). This type of information can be graphically depicted in actor and goal diagrams.

#### **3.2 AORML**

The Agent-Object-Relationship (AOR) modeling approach [3] is based on an ontological distinction between active and passive entities, i.e. between *agents* and *objects*. In AORML, an entity can be an *agent*, *an event*, *an action* (also specialized in *interaction*), *a claim*, *a commitment*, or an *object*. Agent and object form, respectively, the active and passive entities, while actions and events are the dynamic entities of the system model. Commitments and claims establish a special type of relationship between agents. These concepts are fundamental components of social interaction processes and can explicitly help to achieve coherent behavior when these processes are semi or fully automated. Besides AOR models human, artificial and institutional agents. Institutional agents are usually composed of a number of human, artificial, or other institutional agents that act on its behalf. Organizations, such as companies, government institutions and universities are modeled as institutional agents, allowing us to model the rights and duties of their internal agents. For further

reference, we refer to [9] and to the AOR website: [<http://oxygen.informatik.tu-cottbus.de/aor/>].

### 3.3 Using Tropos and AORML to Engineer KM Systems

KM can be defined as a systematic process for acquiring, organizing and communicating knowledge to all members of an organization, enabling them to be more effective and productive in their work [1,6,7,8]. This process is based on practices and technologies that motivate knowledge exchange, so that knowledge can be replicated and amplified to be used in all points-of-action within the organization.

Perhaps, the main attractive characteristic of Tropos is the fact that it is based in *goal modeling*. According to Nonaka and Takeuchi [8], one of the most important KM references today, one of the main drivers of knowledge creation is the organization's intention, defined as "an organization's aspiration to its goals". This turns goal modeling into an important step towards understanding the strategies of the organization regarding knowledge creation and sharing.

If on one hand, Tropos provides a good abstract view of the organization, on the other hand, this methodology's weakness stems from the fact that it does not provide tools to model agent's interaction and behavior with an appropriate amount of detail. We propose to overcome this limitation by also adopting AORML, an UML-based language to model agent-oriented systems. Understanding how well people interact is crucial to grasp how knowledge flows within the organization. This understanding is also important to enable system agents to go through detailed design, thus being prepared for implementation. AORML offers a set of three types of *interaction diagrams*, modeling agent's interaction protocols as well as their internal behavior. Another strength from AORML is providing deontic modeling constructs such as commitments and claims, which form the basis for the establishment of norms and contracts. Such *normative dimension* is an important one when dealing with agent-mediated KM [7], so as to regulate coordination and operational mechanisms within the organization, while dealing with knowledge creation and dissemination.

## 4 Ontology-Based Analysis and Design of ARKnowD

As mentioned in section 2, in order to guarantee the consistency of the resulting modeling language, we seek theoretical support in a foundational ontology. The UFO foundational ontology covers concepts such as entities (agents and objects), events and actions, but also what we call social concepts such as plan, action, goal, agent, intentionality, commitment etc. UFO has been assembled mainly based on works from Philosophy and Cognitive Sciences [5,9,10]. The positive outcomes of UFO's application has been multiple. First, because this work provided us with a consistent method to evaluate and combine the Tropos's notation and AORML. But also because it confirmed our intuitions (discussed in section 3.3) that these two applied approaches are indeed suitable for the KM domain. In other words, the concepts which we found suitable to model the KM domain (which are included in the foundational ontology) were mainly the ones covered by these two modeling languages applied in combination.

## 4.1 UFO

In the sequel, we discuss a fragment of UFO in line with the purposes of this article. For a full discussion regarding this foundational ontology, one should refer to [5,9,10].

We start with the fundamental distinction between universals and individuals. The notion of universal underlies the most basic and widespread constructs in conceptual modeling. Universals are predicative terms that can possibly be applied to a multitude of individuals, capturing the general aspects of such individuals. Individuals are entities that exist instantiating a number of universals and possessing a unique identity.

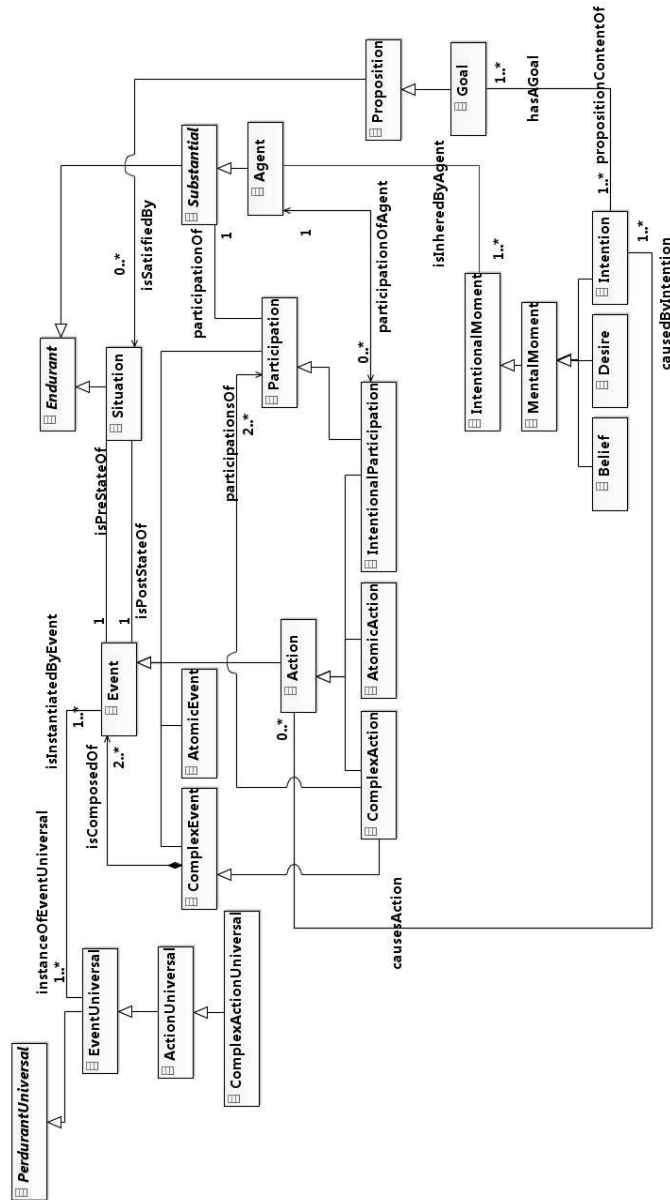
Further, UFO makes a distinction between the concepts of Endurants and Events (also known as Perdurants). Endurants are individuals said to be wholly present whenever they are present, i.e., they are in time, in the sense that if we say that in circumstance c1 an endurant e has a property P1 and in circumstance c2 the property P2 (possibly incompatible with P1), it is the very same endurant e that we refer to in each of these situations. Examples of endurants are a house, a person, the moon, a hole, an amount of sand. For instance, we can say that an individual John weighs 80kg at c1 but 68kg at c2. Nonetheless, we are in these two cases referring to the same individual John. Events (Perdurants), in contrast, are individuals composed by temporal parts, they happen in time in the sense that they extend in time accumulating temporal parts. An example of an Event is a business process. Whenever an Event occurs, it is not the case that all of its temporal parts also occur. For instance, if we consider a business process “Buy a product” at different time instants when it occurs, at each of these time instants only some of its temporal parts are occurring.

A Substantial is an Endurant that does not depend existentially on other Endurants, roughly corresponding to what is referred by the common sense term “Object”. In contrast with Substantials, we have Moments (also known as particularized properties and objectified properties). Moments are existentially dependent entities, i.e., for a Moment x to exist, another individual must exist, named is bearer. Examples of Substantials include a person, a house, a planet, and the Rolling Stones; examples of Moments include the electric charge in a conductor, a marriage, a covalent bond as well as mental states such as individual Beliefs, Desires and Intentions (or internal commitments). The last three examples fall in the subcategory of Mental Moments.

UFO also adds distinctions concerning the intentionality of events to this basic core. Examples include the concepts of Action, Action Universal, Action Contribution and Agent.

Actions are intentional events, i.e., events which instantiate a Plan (Action Universal) with the specific purpose of satisfying (the propositional content of) some Commitment of an Agent. The propositional content of a commitment is termed a Goal. Only agents (entities capable of bearing intentional moments) can perform Actions. As events, actions can be atomic (Atomic Action) or complex (Complex Action). While an Atomic Action is an action event that is not composed by other action events, a Complex Action is a composition of at least two basic actions or Participations (that can themselves be atomic or complex).

Participations can themselves be intentional (i.e., Actions) or non-intentional Events. For example, the stabbing of Caesar by Brutus includes the intentional participation of Brutus and the non-intentional participation of the knife. In other words, we take that it is not the case that any participation of an agent is considered an action, but only those intentional participations called Action Contributions.



**Fig. 1.** Fragment of UFO.

The category of agents further specializes in Physical Agents (e.g., a person) and Social Agents (e.g., an organization, a society). In an analogous manner, Non-Agentive Substantials (or Objects) can also be categorized as Physical Objects (e.g., cars, rocks and trees) or Social Objects (e.g., a currency, a language, the Brazilian constitution). Agents can also be further specialized into Human Agent, Artificial Agent and Institutional Agent,

which can be represented, respectively, by human beings, computationally-based agents and organization or organizational unit (departments, areas and divisions). Institutional Agents are composed by a number of other agents, which can themselves be Human Agents, Artificial Agents or other Institutional Agents.

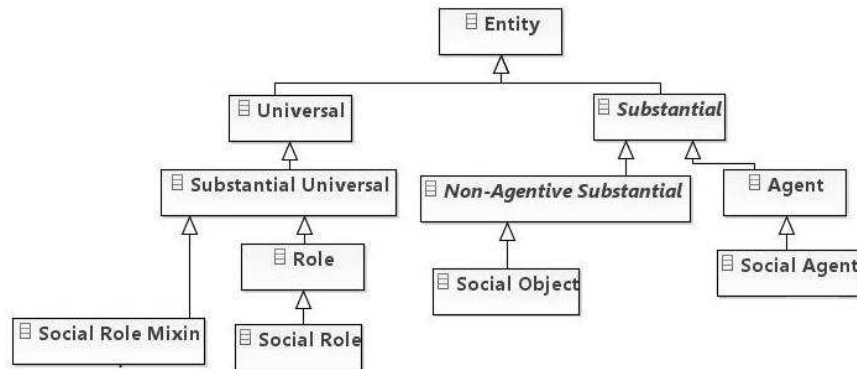


Fig. 2. Fragment of UFO with social aspect.

#### 4.2 Applying UFO to analyze Tropos

We interpret the metaclasses *Actor* and *Role* in Tropos as the concepts of Agent and Social Role in UFO (respectively). An agent role is defined by the set of social moment universals (commitments and claims implied by the role) [10].

We view Tropos goals as Goals in UFO. Goals in UFO are sets of intended states of affairs of an agent. The relation between an Actor in Tropos and a Goal (through the meta-association *wants*) is interpreted indirectly by making use of the concept of Intention (or Internal Commitment) in UFO, which is a Mental Moment of an Agent. As previously discussed, UFO contemplates a relation between Situations and Goals such that a Situation (or possibly a number of Situations) may satisfy a Goal. In other words, since a Goal is a proposition (the propositional content of an Intention), we have that a particular state of affairs can be the truthmaker of that proposition. This interpretation choice seems to model directly the intention behind the concept of *hardgoal* in Tropos. For the case of *softgoals*, a different analysis must be conducted.

The concept of softgoals does not have a uniform treatment in the Tropos community. Sometimes, softgoals are taken to represent non-functional requirements [11]. In other times, a softgoal is considered as a fuzzy proposition, i.e., one which can be partially satisfied (or satisfied to a certain degree, or yet, *satisfied*) by Situations [12]. We here take a different stance, namely, that a softgoal is one “subjective to interpretation” and “context-specific”.

As a consequence of this conception, for the case of softgoals, it seems to be impossible to eliminate a judging agent (collective or individual) from the loop. Thus, instead of considering in the ontology a new *satisfices* relation between Situation and Goal which perhaps should contemplate a fuzzy threshold of satisfaction, we take a different approach. We consider the relation of satisfaction as a ternary relation that can hold between an agent, a goal and situation. An instance of this relation is derived from *the belief of an agent that a particular situation satisfies the goal at hand*. Now,



in this view, different agents can have different beliefs about which sets of situations satisfy a given goal. In fact, it is exactly this criterion which seems to capture the aforementioned notion of softgoals and its differentiae w.r.t. hardgoals: (i) a goal  $G$  is said to be a hardgoal iff the set of situations that satisfy that goal is necessarily shared by all rational agents; (ii) a goal  $G$  is said to be a softgoal iff it is possible that two rational agents  $X$  and  $Y$  differ in their beliefs to which situations satisfy that goal.

Seeing the distinction between these subcategories of goals under this light, allows us to talk about different levels of “softness” between different formulations of a goal. In one end of the spectrum, each individual agent would have a different belief about which situations satisfy a goal. In the opposite end, we have a hardgoal. In between, we can have communities of agents (or collective agents) of different sizes which share a common belief regarding this set of situations.

The mapping of the Plan concept from Tropos to some UFO concept is established in a direct manner. In section 3.1, we stated that a Plan in Tropos is a specific way of doing something to satisfy some Goal (or *satisficing* some Softgoal). From the UFO ontology (section 4.1), we have that an Action (instance of an Action Universal) is an intentional event performed by agents with the purpose of achieving goals. Consequently, the Tropos *Plan* construct can be interpreted as an Action Universal.

In Tropos, goals can be further structured by using different types of relations, namely, AND-decomposition and OR-decomposition. Since Goals are taken here to be propositions, if we have that goals  $G_1 \dots G_n$  AND-decompose goal  $G_0$ , this relation should be interpreted as:  $(G_0 \leftrightarrow (G_1 \wedge G_2 \wedge \dots \wedge G_n))$ . In an analogous manner, and OR-decomposition  $G_1 \dots G_n$  of goal  $G_0$  should be interpreted as:  $(G_0 \leftrightarrow (G_1 \vee G_2 \vee \dots \vee G_n))$ . Here once more, these relations reflect logical relations between propositions and, accordingly, are independent of an Agent’s point of view (contra Fig.2).

We have offered an ontological analysis of the relation of Dependency in Tropos elsewhere [10]. In that paper, we show that Tropos overloads in the same construct the two different (ontological) relations of Dependency and Delegation, which constitutes another case of construct overload in the language. As discussed in depth there, these relations belong to different ontological categories: whilst the former is an example of a formal relation, the latter is one a material relation. To put it baldly, agent  $X$  *depends on* agent  $Y$  for goal  $G$  iff  $G$  is a goal of  $X$ ,  $X$  cannot achieve  $G$ , and  $Y$  can achieve  $G$ . Notice that in this case, agents  $X$  and  $Y$  do not even have to be aware of this dependency. In contrast, if agent  $X$  delegates goal  $G$  to agent  $Y$  then: there is a social commitment  $c$  from  $Y$  to  $X$ ;  $G$  is the propositional content of  $c$ .

The remaining relationship types from Tropos (namely, means-end and contribution) remain to be analyzed in detail. This remains as future work and should thus bring new theoretical and practical implications.

### 4.3 Applying UFO to analyze AORML

The primitives contained in AORML and the ontological categories in UFO bear a rather straightforward relation to each other. Here, due to space limitations we refrain from presenting an AORML metamodel and focus on the fragment of this language which is germane to the purposes of this article. The notions of agent, action, event, commitment and claim in AORML are directly mapped to their counterparts in UFO. The notion of Object in AORML is mapped to the one of Non-Agentive Substantial (or Object) in UFO. An interaction in AORML is interpreted as an Action in UFO, i.e., interactions in AORML can also represent a single-contribution of an Agent in joint

action. Finally, a relationship in AORML is interpreted as a relation universal in UFO (further specialized in both formal relation and material relation) [1].

#### 4.4 Combining Tropos and AORML through Metamodel Transformation

Having clarified the semantics of the modeling constructs through interpretation in terms of UFO, we can establish the correspondence between the constructs in each of the identified fragments of Tropos and AORML (see in Table 1).

**Table 1.** Mapping Tropos into AORML

<b>Tropos Concepts</b>	<b>AORML Constructs</b>
actor	agent
plan	interaction
resource	object
dependency	relationship
delegation	relationship and commitment
resource acquisition	relationship and commitment

In this work, we apply the MDA's metamodel transformation technique, which requires a mapping from the modeling constructs of the source (the Tropos' notation) to the destiny language (AORML). In other words, mapping concepts as prescribed by Table 1 has practical implications in designing the system's model. For example, a Tropos's *plan* may be mapped into AORML's *interaction* concept. In practice, for each plan in a Tropos model, there can be one or more **AOR Interaction Sequence Diagram**, modeling the interactions of the agents participating in this plan (i.e. agents having the plan, or being connected to it by a dependency link). Another interesting illustration comes from the differentiation we introduced between Tropos' *dependency* and *delegation*. The former only maps into an AORML relationship while the latter maps both to a relationship and a commitment. Conceptually, this should be clear from section 4. However, in practice, this leads to the following distinctions: both for Tropos dependency and delegation, an association link may be depicted between these agents in an **AOR Agent Diagram**, typically used for information modeling. Now, besides this association delegations also lead to the establishment of an AORML commitment/claim pair between the (delegate and delegator) agents. This construct is usually depicted in interaction modeling, using one or more types of **AOR interaction diagrams**.

Note that one of the most important entities in Tropos, i.e. the concept of *goal* is not mapped into AORML. This is not a contradiction. Conversely, it relates to the fact that ARKnowD applies goal modeling exclusively for requirements analysis and architectural design. On detailed design, all goals have already been dealt with. For instance, goals may have been fulfilled or abandoned. But most commonly, goal analysis leads to the delegation of unsolved goals to new or old agents, who are either part of the organization or a new information system. And finally, concrete plans are assigned to goals with the purpose of accomplishing them. Consequently, when the

detailed design activity starts, plans should be modeled rather than goals. As observed in Table 1, plan modeling may be done through the use of AOR Interaction Sequence Diagrams, which detail the protocol of communication between agents to realize a specific sequence of actions. In the end, we do not, however, lose the connection to the goals initially modeled in during requirements analysis. This is still traceable through the plan trees linked to each of these goals.

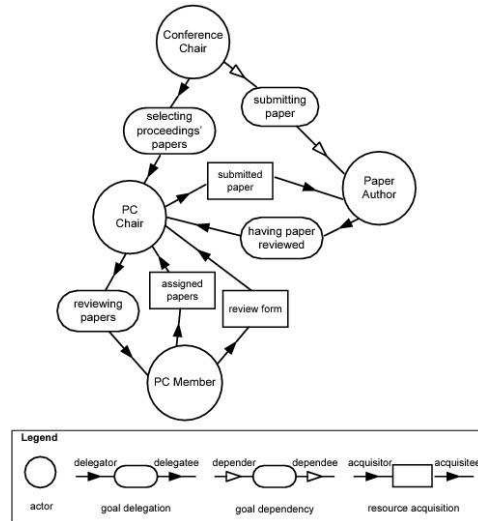
After conceiving the mapping rules of Table 1, it is possible to automate the metamodel transformation between the two languages, by implementing these rules. Aiming at providing automated support to ARKnowD, we started to integrate AORML into an existing Tropos modelling tool named TAOM4E (<http://sra.itc.it/tools/taom4e/>), implementing the mapping of a Tropos Actor Diagram into an AORML agent Diagram. For that, we used a transformation engine named Tefkat (<http://tefkat.sourceforge.net/>). Basically, Tefkat receives as input the metamodels of the two modeling languages (i.e. the metamodels of the Tropos language and AORML), along with the source Tropos model developed with the use of TAOM4E. The mapping between the two metamodels is directly implemented using Tefkat's declarative language. The result is an AORML model. Future work remains on the remaining mappings, so as to deliver a modelling tool which enables full design using ARKnowD.

## 5 Working Example

In this section, we present a simple example of the use of ARKnowD, with the main purpose of illustrating the transformation between the notations of Tropos and AORML. We find the conference review process an appropriate scenario to exemplify ARKnowD, because this is a well-known setting for the academic community. For space limitation, we just exemplify the use of two diagrams (one of each language) that were targeted in the aforementioned automation initiative. For a full description of this working example, including other diagrams, as well as for a more complex scenario of application of ARKnowD in the Knowledge Management domain, please refer to [1].

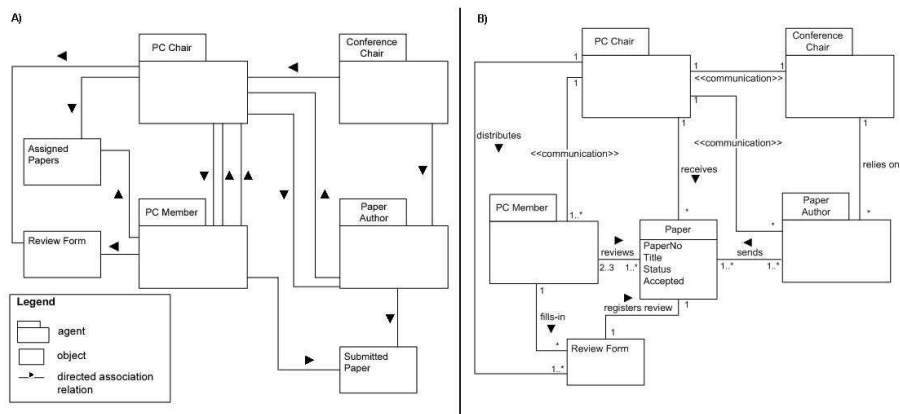
Figure 3 presents a Tropos actor diagram, depicting the main agents of the scenario, along with some goal and resource dependencies between them. The diagram shows that the scenario involves the participation of four agents, namely the Conference Chair, the PC Chair, the Paper Author and the PC Member. For realizing the conference, the Conference Chair depends on the Paper Author to submit papers that will be selected for presentation in the conference (submitting paper goal). For this papers selection, the Conference Chair delegates to the PC Chair the responsibility of selecting the best papers to be published in the conference proceedings (selecting proceedings' papers goal). The PC Chair and the Paper Author have a mutual relationship. While the PC Chair wants to acquire papers submitted by the Paper Author (submitted paper resource), the Paper Author delegates to the PC Chair the goal of having his paper reviewed as part of the papers selection process (having paper reviewed goal). However, the PC Chair does not review all papers on his own. For that, he relies on PC Members (reviewing papers

goal). For accomplishing this goal, the PC Member must receive the papers assigned to them (assigned paper resource), along with the review form (review form resource) from the PC Chair.



**Fig. 3.** Tropos Actor Diagram

At this point, we can already exemplify the first transformation. Figure 4(A) depicts an AOR Agent Diagram (AD) that can be automatically generated with basis on the goal diagram of Fig. 3, using the transformation rules described in Table 1.



**Fig. 4.** AOR agent diagrams: (A) automatically generated from previous Tropos actor diagram and (B) finalized, after designer's edition.

This figure depicts the agents and objects of the scenario, respectively transformed from the Tropos agent and resource constructs. Besides the scenario's entities, the diagram also depicts the relations between them, converted from the dependencies, delegations and acquisitions shown in the previously presented Tropos actor diagram.

Both the number and direction of the relations between agents are inferred from the number and directions of the dependency, delegation and acquisition links on the actor diagram. For instance, between PC Chair and PC Member, there are three relations, corresponding to the two acquisitions and one delegation previously depicted in Fig. 3, and following the same directions of such links. Although this first automatic AD is truthful to our scenario, some modifications may be necessary for enabling its best use in practice. This diagram can then be revised and modified, resulting in the AD of Fig. 4(B). In this second AD, two objects from the AD of Fig.4(A), namely Submitted Paper and Assigned Paper were merged into the Paper object. This comes from the realization that the previously depicted resources on the Tropos actor diagram actually referred to the same object, in two different states (i.e. 'submitted' and 'assigned'). Hence, the two objects originated a single one, and such state is now given by the status attribute in the Paper object. In addition to that change, multiple relations between agents were reduced to one (as a result of a choice made by the designer. In other situations, multiple relations may be considered desirable, thus being maintained) and all relations were named. Finally, some associations between two agents were substituted by a specific type of relation, named communication relation (note the communication stereotype, an extension introduced by AORML). Besides being related by associations, agents typically relate through communication relations, which indicate that they interact to accomplish their goals. Typically, communication relations occur among agents that previously delegated goals or tasks, or acquired resources from one another. In other words, for a delegation or an acquisition to occur, agent A must explicitly interact with agent B, either to ask him/her to accomplish some goal or execute a task on his/her behalf, or to acquire a resource controlled by agent B.

Note also that the diagram of Fig. 4(B) presents the cardinalities (not present in the type of model of Fig. 4(A)) of all agents and objects of the scenario. In the case depicted here, only association relations are necessary among the scenario's entities. In other cases, generalization and composition relations may be necessary. In general, all UML relations may be normally used in the AOR AD.

## 6 Final Considerations

This paper described our approach to design an engineering language to the ARKnowD methodology, which combines two distinct agent-oriented software engineering approaches, namely Tropos and AORML. For mapping the two notations, a theoretical analysis was made with the use of the UFO foundational ontology. Moreover, an MDA-inspired transformation method was used and partially implemented in an agent-oriented modeling tool named TAOM4E, currently under development. Ongoing work in that respect includes analysing further concepts and relationships from the source languages. Moreover, we are currently developing case studies in a real setting. The results of this case study should point out to other directions regarding the evolution of the ARKnowD methodology.

Furthermore, our research group has been involved in several initiatives applying ontological foundations to enable modelling languages evaluation and (re) design.

The most related ones regard: a) the combination of goal modelling (Tropos) and business process modelling (ARIS-EPC) to enable a comprehensive strategic analysis before stepping into business process engineering within organizations [13]; b) analyzing and exposing the semantics behind Software Process Reference Models [9].

**Acknowledgments.** This research has been partially supported by FAPES (Grants #37274554/07 and #45444080/09) and CNPq (Grant #481906/2009-6). The second author of this chapter holds the CNPq (Brazilian National Research Council) Productivity Grant #309382/2008 4. We are grateful to João Paulo A. Almeida, Evellin C. Cardoso and Paulo Sérgio Santos Jr. for their valuable contributions to this work.

## References

1. Guizzardi, R.S.S.: Agent-oriented Constructivist Knowledge Management. PhD Thesis, University of Twente, The Netherlands (2006)
2. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. and Perini, A.: Tropos: An Agent-Oriented Software Development Methodology. *Int. J. of Autonomous Agents and Multi Agent Systems*, vol. 8, no. 3, 203--236 (2004)
3. Wagner, G.: The Agent-Object-Relationship Meta-Model: Towards a Unified View of State and Behavior, *Information Systems*, vol. 28, No. 5, pp. 475--504 (2003)
4. Miller, J. and Mukerji, J.: MDA Guide Version 1.0.1, *omg/2003-06-01*, available at <http://www.omg.org/docs/omg/03-06-01.pdf> (2003)
5. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. PhD Thesis, University of Twente, The Netherlands (2005)
6. Dignum, V.: A Model for Organizational Interaction: Based on Agents, Founded in Logic. PhD thesis, Utrecht University, The Netherlands (2004)
7. Dignum, V.: An Overview of Agents in Knowledge Management. Technical Report UU-CS-2004-017, Inst. Information and Computing Sciences, Utrecht University, The Netherlands (2004)
8. Nonaka, I., Takeuchi, H.: *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, New York (1995)
9. Guizzardi, G., Guizzardi, R.S.S., Falbo, R.A.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In: *Proc. of the Iberoamerican Conference on Software Engineering (CIbSE,2008)*, Recife, Brazil (2008)
10. Guizzardi, R.S.S., Guizzardi, G.: From Tropos to AORML, Using a Foundational Ontology. In Giorgini, P., Maiden, N., Mylopoulos, J. and Yu, E. (eds.): *Tropos/i\*: Applications, variations and Extensions*, Cooperative Information Systems Series, MIT Press (forthcoming)
11. Chung, L., Nixon, B., Yu, E. and Mylopoulos, J.: *Non-functional requirements in Software Engineering*. Kluwer Academic Publishers (2000)
12. Letier, E., van Laamsweerde, A. Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering, *ACM SIGSOFT Software Engineering Notes*. Volume 29, Issue 6 (2004)
13. Cardoso, E.C.S., Almeida, J.P.A., Guizzardi, G., Guizzardi, R.S.S.: Eliciting Goals for Business Process Models with Non-Functional Requirements Catalogues. *10th International Workshop on Business Process Modeling, Development and Support, CAISE 2009*, vol. 29, pp. 33-45, Amsterdam (2009)