Using Goal Modeling and OntoUML for reengineering the Good Relations Ontology

Jordana S. Salamon, Cássio C. Reginato, Monalessa P. Barcellos, Renata S.S. Guizzardi

Ontology & Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo– Vitória – ES – Brazil

{jssalamon,cassio.reginato,monalessa,rguizzardi}@inf.ufes.br

Abstract. *Ontologies have been used as vehicles to assign semantics to data in several domains, since they represent a conceptualization about the domain of interest and establish a common vocabulary to be shared. For properly addressing the domain of interest and serving as a means to attribute meaning to data, ontology should provide a correct understanding of the domain conceptualization. This paper presents some results of an ontological analysis of the Good Relations Ontology (GRO). For the ontological analysis, we used Goal Modeling to delimitate the ontology scope and the Unified Foundational Ontology to analyze its conceptual model. Our analysis discovered several limitations that prevent GRO from properly addressing the domain of interest. Thus, we reengineered GRO to correct such limitations, and implemented it using OWL.*

Keywords*: Goal Modeling, OntoUML, UFO, Ontological Analysis, Good Relations Ontology.*

1. Introduction

Trading over the Internet is a marketing model that has increased in the last years and tends to be the major trading method in the near future. For succeeding in this market, organizations have to make their products and services visible to potential customers. Having product and service offerings information captured by search engines, recommender systems and other applications can increase the visibility of product and service offerings and promote competitiveness. The use of ontologies to assign semantic to data is useful in this context.

The Good Relations Ontology (GRO)[1] is a lightweight ontology addressing the e-commerce domain, particularly product and service offering. It is basically a domain structural model of generic concepts that can be used as a schema to describe e-commerce related data, such as products, prices, and product properties. It provides a common vocabulary that can be embedded into existing static and dynamic web pages and processed by computers. It is used by several companies (e.g., Google, Kmart, BestBy) and allows for searches involving offerings and suppliers.

The quality of the ontology model is crucial for gathering consistent information from data linked to the ontology (Tartir et al. 2010). According to Vojislav and Leon

[1] http://www.heppnetz.de/projects/goodrelations/

(2000), the quality of a model can be measured by its syntactic and semantic correctness and completeness. The syntactic correctness and completeness means that the language constructs applied in the model produce a valid and consistent model. The evaluation of this criteria can be done by checking if the model does not violate any language constraint. The semantic correctness and completeness, in turn, means the right representation of the domain conceptualization. It is a hard task to satisfy and evaluate this criterion, mainly because it depends on the goals behind the model building. Thus, as argued by Lindland et al. (1994), scope clarity is one of the main aspects to be considered when evaluating model semantics. Moreover, satisfy and evaluate semantic correctness and completeness is not trivial because it is difficult to ensure a unique and correct understanding of a given domain conceptualization (Guizzardi 2005). This issue is related to ontological commitment, which expresses what exactly a domain conceptualization represents (Guarino et al. 2009).

Considering that GRO has been used as the e-commerce data model of the Schema.org[2] initiative and has a lot of data linked to it, we decided to evaluate GRO taking the aspects previously mentioned into account. Aiming to understand the GRO domain of interest and, thus, the GRO scope, we applied Goal Modeling. Goal Modeling defines procedures to identify and express domain requirements in software engineering and it has been proposed as a method to establish ontology scope by means of the competence questions that the ontology should be able to answer (Fernandes et al. 2011). By applying Goal Modeling, we identified the competence questions that GRO should be able to answer and compared them with the ones defined to GRO. Moreover, we verified if GRO is able to answer the identified competence questions. We noticed several limitations that prevent GRO from properly addressing the domain of interest. Then, in order to get closer to the ontological commitment, we analyzed GRO conceptual model at light of the Unified Foundational Ontology (UFO) (Guizzardi 2005). Based on the ontological analysis results, we redesigned the GRO conceptual model and represented it by using OntoUML (Guizzardi 2005), a modeling language whose metamodel is based on the ontological metaproperties of UFO. Finally, we produced an operational version of the ontology in OWL.

This paper addresses the GRO ontological analysis and presents some of the produced results. It is organized as follows: Section 2 provides the background for the paper, talking about Goal Modeling, UFO and OntoUML; Section 3 addresses the use of Goal Modeling to understand the GRO scope; Section 4 describes GRO ontological analysis and redesign; and Section 5 presents our final considerations.

2. Background

2.1. Goal Modeling

Goal modeling has been successfully applied in requirements engineering (Yu, 2010) and has also been used to enrich the requirements analysis phase of the ontology engineering process, as in (Fernandes et al, 2011). The incorporation of explicit goal representations in requirement models provides a criterion for requirements

[2] http://schema.org

completeness, i.e., the requirements can be judged as complete if they are sufficient to achieve the goals they refine (Liu & Yu 2004).

When applying goal modeling, it is necessary to define the framework or terminology used to express the goal models. The *i** framework (Yu, 2010) is currently one of the most widespread goal-oriented and agent-oriented modeling and reasoning methods (Franch et al. 2016). The *i** modeling approach is an attempt to bring social understanding into the system engineering process by putting social concepts into the core of the daily activity of system analysts and designers. The main constructs of the *i** language are: *actor, hardgoal, softgoal, task* and *resource*. In *i** modeling, the focus is on intentional properties and relationships rather than actual behavior. By attributing intentionality, we can express why an actor undertakes certain actions, or prefers one alternative over another. This kind of analysis can be made within a *Strategic Rationale* model, where hardgoals, softgoals, tasks and resources are allocated to an actor to meet his intentions. The rationale analysis allows understanding what an actor can achieve and how he can achieve it. It is possible to reason about what he needs to fulfill his goals and the alternatives he can choose to achieve them. *i** also allows for dependency analysis within a *Strategic Dependency* model, demonstrating the relations and dependencies between actors, revealing the weak spots of a relationship and the network and propagation of dependencies that can be generated from a single relationship (Yu 2009) .

In the ontology engineering context, goal modeling can help understand the domain of interest and the involved actors. Moreover, from goal models it is possible to derive competency questions to delimit the ontology scope. Fernandes et al. (2011) propose a method to use goal modeling as the starting point of competency questions definition. It is based on the use of *i** to elicit the competency questions of an ontology-to-be and derivate the ontology initial concepts. According to the method, first, it is necessary to identify the actors involved in the ontology domain of interest and the goals related to them. Then, competency questions related to the goals must be identified and initial concepts of the ontology are captured and linked to the competency questions. Last, concepts are refined, specified and the ontology models are developed.

2.2. UFO and OntoUML

The Unified Foundational Ontology (UFO) (Guizzardi 2005) is a foundational ontology that has been developed based on a number of theories from Formal Ontology, Philosophical Logics, Philosophy of Language, Linguistics and Cognitive Psychology. UFO consists of three parts: UFO-A, an ontology of endurants (objects), UFO-B, an ontology of events (perdurants), and UFO-C, an ontology of social entities built on the top of UFO-A and UFO-B. In this paper, we focus on a UFO-A fragment. The top distinction of UFO is between Individuals and Universals. An **Individual** is an instance of a **Universal**. Thus, Individual is an instance that actually exist in the real world, as *John,* which is an instance of the universal *Person*.

Figure 1 depicts the endurant universals hierarchy in UFO. An **Endurant Universal** is either a **Substantial Universal** or a **Moment Universal**. Naturally, these are kinds of universals whose instances are **Substantial Individuals** and **Moments**, respectively. Substantials are existentially independent entities (e.g., *John*). Moments, in turn, are entities that can only exist by inhering in other individuals (e.g., *color*). A

Moment can be a **Mod**e, which inheres in a single individual (e.g., *color*) or a **Relator**, which depends on a plurality of entities (e.g., *employment*) and, for this reason, provides the material connection between them.

Substantial Universals are distinct regarding principle of identity, rigidity and dependence. Concerning the principle of identity, the distinction is between **Sortal** and **Mixin Universal**. The instances of a Sortal have a unique principle of identity while Mixin Universals are abstract entity types that have instances with different principles of identity. For instance, *John* is an instance of *Person* (Sortal Universal) that obey a unique principle defined in function of Person's essential attributes. In this case, *Person* provides the principle of identity to its instances. *John* can be also an instance of an abstract entity type *Client* (Mixin Universal) that can have instances that are *Person* or *Organization* and, thus, have different principle of identity. Thus, Mixin Universals do not provide any principle of identity for its instances, instead, they just aggregate properties of distinct principle of identity providers. Both Sortals and Mixin Universals are classified in terms of rigidity. Rigidity is a modal property that defines the behavior of the instances of a given entity type. An entity type is said rigid iff its instances never cease to be its instances in all possible worlds (e.g., *Person*, *Chair*). Otherwise (if some instances of an entity type can cease to be instance of this type in a possible world), this type is said non-rigid. An entity type is anti-rigid if it is non-rigid and if all its instances necessarily can cease to be its instances in a possible world (e.g., *Student*, *Employee*). **Substance Sortals** (Kind and Collective) are rigid entity types that provide principle of identity for theirs instances. A **Kind** represents a functional complex which means that it has distinct parts (e.g., *Human Being*) while a **Collective** is an entity type that is composed of equal parts (e.g., *Committee*). All other **Sortals Universals** (i.e., Role, Phase and Subkind) are entity types that can only occur as specializations of a Substance Sortal, since all instances must have exactly one principle of identity provided by a unique Sortal. Thus, a **Subkind** is a **Rigid Sortal Universal** that is a specialization (directly or indirectly) of a Kind or a Collective. Nonetheless, Role and Phase are anti-rigid types differentiated by the dependence aspect. A **Role Universal** is externally dependent, which means that its instances necessarily depend on another instance (e.g., *System Analyst*). A **Phase Universal** is defined by intrinsic properties (e.g., *Child*). The distinction between Mixing Universal types only regards rigidity. Then, a **Category** is rigid and **RoleMixin** is anti-rigid.

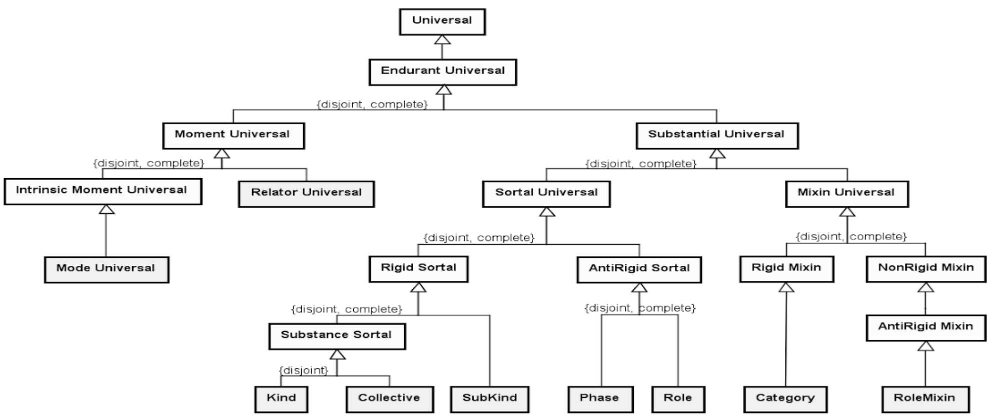

Figure 1 - UFO-A Universal Hierarchy.

The UFO-A categorization was used as a basis to OntoUML (Guizzardi 2005), a modeling language for ontology-driven conceptual modeling. Its profile comprises a number of stereotyped classes and relations implanting a meta-model that reflects the UFO structure and axiomatization. The leaves of the UFO-A hierarchy (Figure 1) are the possible categorization of entity types that can be applied when using OntoUML.

3. Analyzing GRO Scope by using Goal Modeling

In order to evaluate GRO, it is important to understand its domain of interest and, thus, its scope. Therefore, we applied Goal Modeling to represent the main aspects and goals related to the domain. It is important to point out that, although GRO is said to be an ontology for the e-commerce domain, in fact, it addresses aspects related only to offering. More specifically, only to product offering (further discussions about that are made later). Thus, we applied Goal Modeling to represent knowledge about product web-offering and establish the GRO scope.

We started by analyzing a number of offerings available on the web and some literature about the topic aiming to gather knowledge about relevant information regarding web-offerings. Then, we identified the main actor (*Offering Specifier*) and stakeholders involved in the domain (*Shipping Company, Payment Company, Delivery Company* and *Business Warrantier*) and built the Strategic Dependency model (not shown here due to space limitation) to identify the relations between them. After that, we identified the main actor goals to understand his needs. The main goal of an Offering Specifier is *have the offering specified*. In order for him to accomplish this goal, it is necessary to accomplish the following ones: *have the price specified, have the offering description specified, have the delivery or shipping methods specified, have the payment methods specified*. Once the actor's goals were identified, for each goal, we derived tasks to accomplish it and identified the necessary resources to perform them. Figure 2 shows a fragment of the Strategic Rationale model produced.

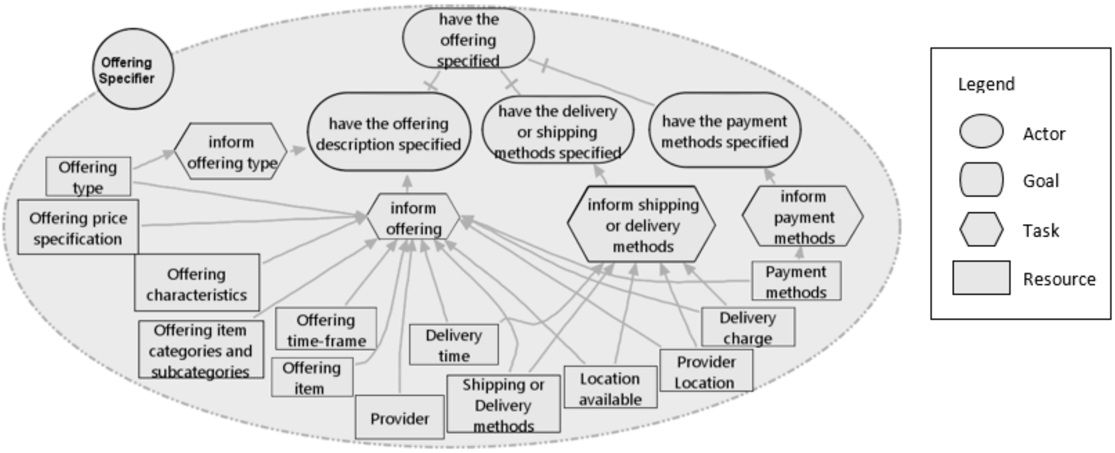

Figure 2 - Fragment of the Strategic Rationale Model.

For achieving the *have the offering description specified* goal, the *Offering Specifier* needs to perform the following tasks: *inform the offering type* (e.g., sell, rent) and *inform offering* (which includes *offering item*, *offering price specification*, *offering characteristics* (e.g., textual description, pictures), *offering item categories and* *subcategories*, *payment methods*, *shipping or delivery methods, provider, delivery time, location available* (e.g., Worldwide, Europe), *provider location and delivery charge*). For achieving the *have the payment methods specified* goal, the *Offering Specifier* needs to *inform payment methods* and for achieving the *have the delivery or shipping methods specified,* the *Offering Specifier* needs to *inform shipping or delivery methods*.

After developing the goal models, we followed the method proposed by Fernandes et al. (2011) and, from the refinement of the goals into tasks and resources, we identified questions that should be answered by an ontology addressing web-offering. For example, considering the model shown in the Figure 2, we can derive competency questions about the payment methods, such as: Which are the payment methods available for a given offering? Thus, based on the defined goal models, the following competency questions were identified: (CQ1) What is the price specification of an offering? (CQ2) Who is the offering provider? (CQ3) Which are the payment methods available for a given offering? (CQ4) Which are the shipping or delivery methods available for a given offering? (CQ5) What are the offering items specified in an offering? (CQ6) Which is the offering type? (CQ7) What is the offered product model? (CQ8) What are the individual products offered in an offering? (CQ9) What are the components of an offering item? (CQ10) What is the individual product warranty? (CQ11) What is the product model warranty? (CQ12) How is the warranty applied? (CQ13) What are the quantitative properties of an offering item? (CQ14) What are the qualitative properties of an offering item? (CQ15) Which is the type of the provider of a given offering? (CQ16) What are the delivery charges of an offering? (CQ17) What is the delivery time of a given delivery or shipping method? (CQ18) To which locations a given offering is available? (CQ19) What is the offering time-frame? (CQ20) What is the shipping or delivery charge for a certain location? (CQ21) What are the business entities responsible for shipping or delivering the offering items? (CQ22) What are the categories and subcategories of an offering item?

Since these competency questions describe the web-offering domain, they should be answered by any ontology addressing that domain. GRO is claimed to be an e-commerce lightweight ontology and, since the web-offering domain is part of the e-commerce domain, GRO should be able to answer those questions. However, GRO does not answer several of them. Moreover, by analyzing the GRO competency questions (Hepp 2008), we can notice that they refer only to web-offering, covering aspects such as offering time-frame, payment methods and price. To be understood as an e-commerce ontology, GRO should answer competency questions related to all the e-commerce sub-domains (e.g., offering, sale, delivery). Therefore, we advocate that GRO does not address the e-commerce domain, but only part of it related to web-offering and, even so, several relevant aspects of this subdomain are not addressed.

Additionally, the GRO competency questions are described in a poor manner. Some of them mix several questions into only one sentence or provide a lot of options in the same sentence. For example: *Which retrievable Web Resources describe an offer {to sell | to provide the service of | to repair | to maintain | to lease out | to dispose} {a concrete individual | some unknown individuals} of a {given good | given service | spare part for a given good | consumables and supplies for a given good} described by a {type of good | specific make and model} that meet certain requirements on {properties | intervals for properties} for which the offering party accepts a given method of payment and provides a certain method of delivery to {consumers |*

retailers} in a given {country | region}? (Hepp 2008) Questions defined like this can easily lead to misunderstandings and make difficult to properly answer the questions in the ontology.

The competency questions derived from the goal models delimit the scope of a web-offering ontology. By comparing the delimited scope with GRO scope we noticed that several aspects are not addressed by GRO. After understanding the web-offering domain and analyzing the GRO scope, we analyzed the GRO conceptual model. The main results are discussed in the next section.

4. Evaluating the GRO Conceptual Model

Our evaluation of the GRO conceptual model regards two different aspects. In the first, we analyzed the model representation and domain coverage. Then, we evaluated the GRO model, verifying if it can properly answer the competency questions identified from the goal models. Finally, we redesigned the GRO model using OntoUML. Due to space limitation, we do not reproduce the GRO model in this paper. It can be obtained in *http://www.heppnetz.de/projects/goodrelations/*.

4.1 Problems related to Model Representation and Domain Coverage

4.1.1 Lack of Cardinality

GRO is implemented in OWL. Its model is represented by using UML and does not represent the associations cardinality. OWL models, in general, do not apply any mechanism to forbid invalid cardinalities and this may be the reason why the GRO UML conceptual model does not represent cardinalities. However, in order to be properly used for both specifying offerings and processing data linked to GRO, it would be desirable to know the cardinalities. They are important, for instance, to distinguish mandatory from optional associations. For example, an offering should always specify the locations where it is available. Thus, the minimum cardinality from offering to location must be 1 (i.e., a mandatory relation). Contrariwise, there can be products with no brand (i.e., optional relation).

4.1.2 Attributes Misrepresentation and Lack of Ontology Integration

There are problems in some attributes defined in GRO. For instance, what GRO defines as *ProductOrService,* and we have decided to call *Offering Item*, has the *Color* attribute. However, a service does not have a color. Moreover, neither a product always has a color (e.g., virtual products like music files, books, etc.). Additionally, a product can have many colors, and in GRO this attribute accepts only one value. We do not agree to define *Color* as an attribute. We consider color as an instance of *Qualitative Property*. This way, it would only be used when an offering item has color and it would be possible to associate an offering item with many colors.

There are also problems regarding the *Category* attribute, which is defined as a literal. We can find on the web many examples of offerings specifying items that belong to many categories and subcategories. Thus, a literal is not a proper way to specify offering items since it turns complicated, for example, to find instances of the same category by comparing literals. For this reason, we argue that *Category* should be a class instead of an attribute and its instantiation should be done by the Product Types

Ontology[3], which defines a huge and precise dictionary of product types and brand names for GRO. In this case and in others like this, the integration with other ontologies could increase GRO quality. For instance, GRO has a class to define the locations where an offering is available. This class has attributes like *Name* and *Category* as literals. Once more, it is not a good way to specify locations and location's category as literals, since it might be useful to apply this information in a structured way. We argue that in these cases, GRO should link to other ontologies (e.g., the Location Ontology[4]) that address the concepts more suitably.

4.1.3 Domain Coverage

In order to verify domain coverage, we checked if GRO is able to properly answer the competency questions presented in the Section 3. We classified the CQs in four groups. The first one refers to CQs that GRO can answer. The second comprises the CQs that GRO does not address. The third includes CQs that could be better answered by integrating GRO with other ontologies. The last refers to CQs that cannot be answered due to problems in the GRO model. Table 1 summarizes the CQs of each group.

Table 1 – Domain Coverage

Group	CQs	Quantity	%
GRO can answer	1,3,4,6, 12, 13, 14, 16	8	36%
GRO does not address	2, 9, 10, 11, 15, 17, 21	7	32%
Other ontologies are necessary	18,19, 20, 22	4	18%
GRO cannot answer due modeling problems	5, 7, 8	3	14%

As one can see, although GRO can answer many CQs, it disregards several important aspects related to offering specification, like the offering item components. Moreover, a modeling problem prevents GRO from answer CQs 5, 7 and 8. This occurs because of the class that GRO calls ProductOrService and that we understand as Offering Item. In the GRO Model this class generalizes Individual (i.e., Individual Product) and Product Model. The problem with this representation is that since Product Model is a subtype of Offering Item, a Product Model would be always a ProductOrService (i.e., an offering item). This is not the case when the ProductOrService is an Individual Product (e.g., a particular cellphone) and the Product Model (e.g., iPhone 7) is just information about the Product being offered and, thus, it is not a ProductOrService). This problem leads to inconsistent instantiations that impacts in the CQs mentioned.

4.2 Redesigning GRO Conceptual Model Using OntoUML

A conceptual model is intended to represent a domain with the purpose of communication (Guizzardi 2005). In other words, a conceptual model is meant to express domains to the human interpretation. Thus, the GRO conceptual model is a guide for people who want to specify offerings on the web. Therefore, this conceptual

[3] https://datahub.io/dataset/productontology

[4] http://dbpedia.org/ontology/Location

model should be presented in a very precise and readable way to avoid misinterpretations. GRO uses the UML class diagram language to specify its model. The problem is that GRO ignores an expressive part of UML constructs like *association end* and *generalization set properties* and *part of relations*. This turns the GRO model interpretation difficult, causing possible double interpretations for the same domain conceptualization and, sometimes, a unique representation for two different domain conceptualizations. The GRO conceptual model seems to have the same expressiveness than the GRO OWL ontology. The problem of this approach is that OWL expressiveness is more focused on computational aspects, thus not being an appropriate form to present information for humans. We advocate that even the UML class diagram language has a limited expressiveness and a number of semantic problems, as presented in (Guizzardi 2006). For instance, UML does not distinguish classes that are instances of Substantials from classes that are instances of Moments. A unique construct *Class* represent both, turning difficult to distinguish between them.

Since OntoUML is based on UFO categories, it has a number of constructs to better specify domain conceptualizations. In other words, since it applies a number of philosophical and cognitive theories extending the UML syntax, it can offer tools to enrich the semantic of the GRO model and solve the aforementioned problems. Next, we present fragments of the redesigned GRO conceptual model, discussing the advantages brought by the use of OntoUML. In the figures, stereotypes indicate the OntoUML categorization.

OntoUML has a number of constraints that must be respected to use each categorization (e.g., a Role must have a Kind as supertype). To build a consistent OntoUML model that does not violate these constraints, the OntoUML patterns presented in (Ruy et al. 2017) can be applied. By applying these patterns, we solved the GRO problems related to the CQs 5, 7 and 8. Figure 3 shows the model fragment related to these CQs. In this fragment, we represent a generic type called *Model Specification* that means a set of prototypical descriptions that can be used either to specify a product (*Individual Product*) or to specify an *Offering Item*. In the first case, we are offering a specific individual (e.g., my cellphone) and we may want to use a product model to better specify it (e.g., my cellphone is an iPhone 7). In the last case, we are offering a model specification (i.e., *Offered Model Specification*, such as iPhone 7) without determining the individual product that the customer will receive.

By applying the Role Pattern (Ruy et al. 2017), we defined that a *Product Model* assumes this role only when it is specifying some *Individual Product*. It assumes another role whenever we are specifying an *Offering Item* in terms of general properties of a given *Model Specification* (e.g., iPhone 7). The redesigned fragment answers the CQs 5, 7 and 8, since it is now possible to distinguish between *Model Specifications* that are *Offering Items* from *Product Models* that are used only to specify *Individual Products*. Moreover, we can distinguish the *Individual Products* that are specified by a *Product Model* from the ones specified by its own properties (e.g., a handcraft).

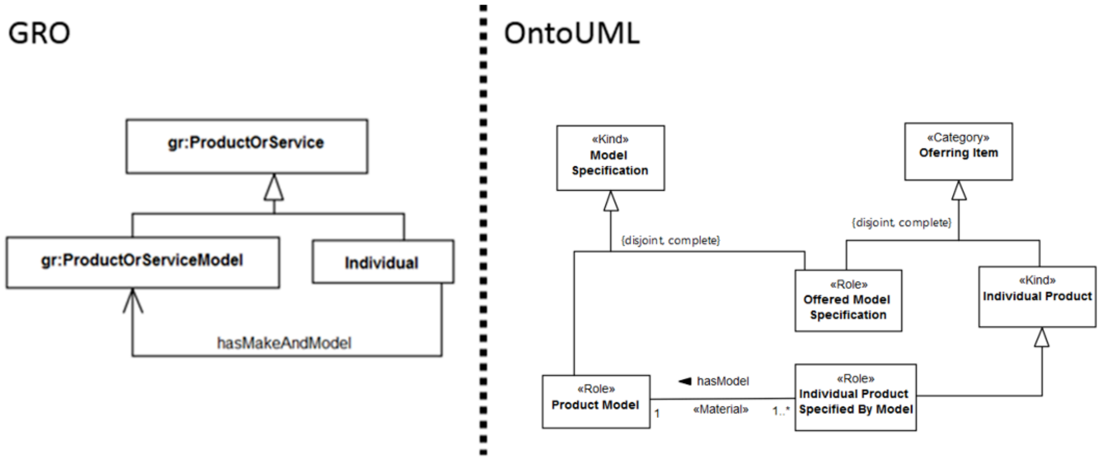

Figure 3 - GRO (left) and OntoUML (right) model fragment to the CQs 5, 7 and 8.

Figure 4 answers CQ9, a competence question not addressed by GRO. To express the *partOf* relation between the *Offering Item* and its components, we applied the OntoUML *componentOf r*elation. This relation applies between a functional complex and its parts. Therefore, the functional complex *Offering Item* contains a set of parts *Offering Item Components*. It is important to point out that we consider as *Offering Item* a set of textual descriptions as it is represented on web-offerings (e.g., the description of a car engine), and not the object being described (e.g., a car engine). The main issue here is that an *Offering Item* may have components that can have recursively other components. This can be solved in OntoUML by applying two subkinds of *Offering Item Component*, such that, one represents an atomic structure (i.e., a component that has no components) and the other a complex one (i.e., a component that has other components). A complex component can be composed by atomic and complex components and, thus, it has the *componentOf* relation associated to the generic type.

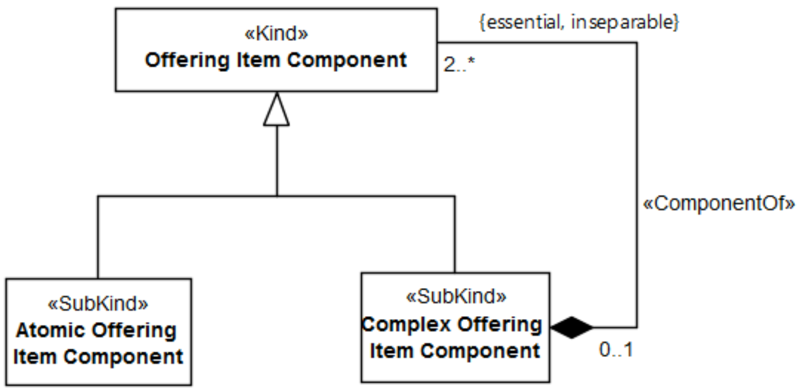

Figure 4 - OntoUML model fragment related to the CQ 9.

After redesigning GRO conceptual model, we rebuilt the GRO OWL model. For that, we applied an automatic transformation from OntoUML to OWL presented in (Barcelos et al. 2013), with some adaptations. The OWL model, the complete redesigned model using OntoUML and the complete goal models produced during this work are available at *http://www.inf.ufes.br/~jssalamon/index.php/contributions*. The

OWL model is integrated to three other ontology models, namely: the Location Ontology[4], the Product Type Ontology[3] and the Data Time Ontology[5] .

5. Final Considerations

Ontological analysis is an important tool to promote improvements in conceptual models and ontologies. This paper presented some of the results of an ontological analysis and reengineer of the Good Relations Ontology, carried out using Goal Modeling and OntoUML. We applied Goal Modeling to understand the GRO scope and domain of interest. From goal models we derived competency questions that should be answered by an ontology addressing the web-offering domain. Aiming to improve the GRO conceptual model, we redesigned it using OntoUML and, then, we developed an OWL ontology. By ontologically analyzing GRO model, we included a foundational layer beneath the concepts and relations, providing the needed ground and consistency to improve GRO towards an e-commerce high quality ontology (Ruy et al. 2014).

Several works have addressed ontology evaluation, attempting to give ontological foundations to conceptual models. Regarding GRO, a few studies were conducted in order to evaluate it. Ashraf and Hadzic (2011) present an Ontology Usage Analysis Framework (OUSAF) that uses metrics to analyze the usage of web ontologies. They applied their framework in GRO as a case study, and realized that a very small part of the ontology had been used on the web and several concepts were not used at all. Later, Asharf et al. (2011) analyzed the GRO usage in terms of data instantiation, conceptual coverage, usefulness and inference provisioning. The analysis shows that data sources that publish their data using GRO, use it mostly to provide information regarding business entity, offering and price. Almost no data source has provided any formal specification of the products being offered. It can be due to GRO poor conceptualization about products and its properties. Different from these works, our study focused on analyzing the GRO conceptualization and improving it by using Goal Modeling and OntoUML.

The main contributions of the work addressed in this paper are: *(i)* goal analysis and definition of competence questions for the web-offering domain, aiming to improve GRO scope clarity; *(ii)* evaluation and redesign of the GRO conceptual model using OntoUML; and *(iii)* a new OWL ontology model aligned to GRO. As future works, based on the results of this work, we intend to develop an e-commerce ontology and use it in interoperability initiatives.

Acknowledgment

This research is funded by the Brazilian Research Funding Agency CNPq (Processes 485368/2013-7 and 461777/2014-2) and FAPES (Process 69382549/2014).

References

Ashraf, J. et al., 2011. Open ebusiness ontology usage: Investigating community implementation of goodrelations. In *Proceedings of the 20th international world wide web conference*. ACM.

[5] https://www.w3.org/TR/owl-time/#time:DateTimeDescription

Ashraf, J. & Hadzic, M., 2011. Domain ontology usage analysis framework. In *Semantics Knowledge and Grid (SKG), 2011 Seventh International Conference on*. IEEE, pp. 75–82.

Barcelos, P.P.F. et al., 2013. An Automated Transformation from OntoUML to OWL and SWRL. In *ONTOBRAS*. Citeseer, pp. 130–141.

Fernandes, P.C.B., Guizzardi, R.S.S. & Guizzardi, G., 2011. Using goal modeling to capture competency questions in ontology-based systems. *Journal of Information and Data Management*, 2(3), p.527.

Franch, X. et al., 2016. The i* Framework for Goal-Oriented Modeling. In *Domain-Specific Conceptual Modeling*. Springer, pp. 485–506.

Guarino, N., Oberle, D. & Staab, S., 2009. What is an Ontology? In S. Staab & R. Studer, eds. *Handbook on Ontologies*. Springer, pp. 1–17.

Guizzardi, G., 2006. On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. *DB&IS, Frontiers in Artificial Intelligence and Applications*, 155, pp.18–39.

Guizzardi, G., 2005. *Ontological Foundations for Structural Conceptuals Models*. University of Twente.

Hepp, M., 2008. Goodrelations: An ontology for describing products and services offers on the web. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, pp. 329–346.

Lindland, O.I., Sindre, G. & Solvberg, A., 1994. Understanding quality in conceptual modeling. *Software, IEEE*, 11(2), pp.42–49.

Liu, L. & Yu, E., 2004. Designing information systems in social context: A goal and scenario modelling approach. In *Information Systems*. pp. 187–203.

Ruy, F.B. et al., 2014. An Ontological Analysis of the ISO/IEC 24744 Metamodel. In *FOIS*. pp. 330–343.

Ruy, F.B. et al., 2017. From Reference Ontologies to Ontology Patterns and Back. *Data & Knowledge Engineering*.

Tartir, S., Arpinar, I.B. & Sheth, A.P., 2010. Ontological Evaluation and Validation. In *Theory and Applications of Ontology: Computer Applications*. Dordrecht: Springer Netherlands, pp. 115–130.

Vojislav, B. & Leon, J., 2000. Evaluating the quality of reference models. In *Conceptual Modeling—ER 2000*. Springer, pp. 484–498.

Yu, E., 2010. Modelling strategic relationships for process reengineering. *Social Modeling for Requirements Engineering*, 11, p.2011.

Yu, E.S., 2009. Social modeling and i*. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 99–121.