Towards an Enterprise Ontology Pattern Language

Ricardo de Almeida Falbo[1], Fabiano Borges Ruy[1,2], Giancarlo Guizzardi[1], Monalessa Perini Barcellos[1], João Paulo Andrade Almeida[1]

[1] Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espírito Santo, Vitória, Brazil
+55-27-4009-2167

[2] Informatics Department, Federal Institute of Espírito Santo, Campus Serra, Serra, Brazil

{falbo, fabianoruy, gguizzardi, monalessa, jpalmeida}@inf.ufes.br

ABSTRACT
Enterprise ontologies are useful for many purposes. Over the years, there have been a number of efforts aiming at building them. However, due to the complexity of the enterprise domain, enterprise ontologies tend to be complex and difficult to reuse. In this paper, we advocate in favor of organizing Core Enterprise Ontologies as Ontology Pattern Languages, since ontology patterns are more and more recognized as an approach that favors ontology reuse. Moreover, we present an initial version of the Enterprise Ontology Pattern Language (E-OPL), and show how it was used for building an enterprise ontology for a specific domain.

Categories and Subject Descriptors
D.2.13 [**Software Engineering**]: Reusable Software – *reuse models, reusable libraries, domain engineering.*

General Terms
Design, Languages.

Keywords
Enterprise ontology, ontology pattern, ontology pattern language, ontology reuse.

1. INTRODUCTION
Nowadays, it is recognized that, in order to cope with many challenges regarding an enterprise, such as communication, integration, support, and development of enterprise systems, a conceptual model of the enterprise is needed [6, 23].

An enterprise model should identify the basic enterprise elements and should specify the information, resources and organizational requirements of these elements [17]. An enterprise model can be a complex model, consisting of several models, focusing on different aspects of an enterprise, such as the enterprise structure and location, its activities, processes, and behavior, enterprise roles, resources, people and capabilities, enterprise goals and constraints, and so on [11, 26, 27].

Ideally, the enterprise conceptual model should be an ontological model [6]. An enterprise ontology should provide a coherent reference model establishing a common conceptualization on enterprises. This common conceptualization can be used to ensure that all parties involved (inside an enterprise and across enterprises) have a shared understanding of the relevant aspects and abstractions of the enterprise [23, 26].

Enterprise ontologies are useful for many purposes. In the context of the Semantic Web in particular, enterprise ontologies can be used for supporting linked data publishing of organizational information across a number of domains [27], for managing enterprise knowledge [16, 19], and for supporting enterprise application integration [1, 6, 25], among others.

Over the years there have been a number of efforts aiming at building enterprise ontologies [20], such as the TOVE (TOronto Virtual Enterprise) Project [9, 10], The Enterprise Ontology [26], CEO (Core Enterprise Ontology) [3], and ORG (the W3C Organization Ontology) [27]. However, since the enterprise domain is too broad, the coverage of the existing enterprise ontologies varies greatly. Some of them address several of these aspects, such as the TOVE Ontology and The Enterprise Ontology. However, as a consequence, they tend to be considered complex and difficult to use [3]. Others, such as CEO and ORG, are core ontologies aiming at gathering the most general business concepts, relations and properties, common to the majority of enterprises, independently of the specific activity field. These core ontologies can be extended to specific domains, such as banking, education, manufacturing, and so on [21].

In this paper, we share the view of the CEO Project [3, 21] and of the W3C Government Linked Data Working Group [27], and advocate in favor of a core enterprise ontology. Moreover, we claim that a core enterprise ontology should be: (i) flexible enough for allowing ontology engineers to explore alternative models in the design of specific enterprise ontologies; (ii) broad enough to cover the entire enterprise universe of discourse; and (iii) modular, in order to allow the ontology engineer to select the ontology fragments that are relevant to the problem in hands. For achieving these characteristics, we argue that a core enterprise ontology should be organized as an Ontology Pattern Language.

An Ontology Pattern Language (OPL) [7] is a network of interconnected domain-related ontology patterns that provides holistic support for solving ontology development problems for a specific domain. An OPL offers a set of interrelated domain patterns, plus a process with explicit guidance on what problems

SAC'14, March 24-28, 2014, Gyeongju, Korea.

can arise in that domain, informing the order to address these problems, and suggesting one or more patterns to solve each specific problem.

In this paper, we present the first version of the Enterprise Ontology Pattern Language (E-OPL). This first version addresses five aspects common to several enterprises: Organization Arrangement, which includes patterns related to how an organization is structured in terms of organizational units, as well as how complex organizations are organized in terms of other organizations; Team Definition, which deals with defining teams for projects, organizations or organizational units; Institutional Roles, which regards roles and positions to be played by enterprise employees; Institutional Goals, which deals with institutional agents' goals; and Human Resource Management, which treats several human resource relations in an enterprise, such as employment, allotment to an organizational unit, team allocation, and position occupation. Besides presenting the first version of the E-OPL, we illustrate its use in the development of an enterprise ontology of Governmental Brazilian Universities.

This paper is organized as follows. Section 2 discusses the importance of enterprise ontologies, and introduces the notion of Ontology Pattern Language. Section 3 contains the main contribution of this paper, presenting E-OPL. Section 4 illustrates how E-OPL was applied in the development of the Brazilian University ontology. Section 5 discusses related work. Finally, Section 6 presents our final considerations.

2. ENTERPRISE ONTOLOGIES AND ONTOLOGY PATTERN LANGUAGES

Ontology solutions are gaining importance in all activities that require a deep understanding of an enterprise and the business sector in which it operates [3]. As a consequence, several enterprise ontologies have been proposed, among them the TOVE (TOronto Virtual Enterprise) Project [9,10], The Enterprise Ontology [26], The Resources-Events-Agents (REA) Ontology [12], Dietz's Enterprise Ontology [6], and ORG (the W3C Organization Ontology) [27].

These enterprise ontologies vary in the coverage to the enterprise domain and in the purposes they intend to achieve. Among the intended uses for some of the enterprise ontologies available, we can point out: (i) enhancing communication between humans [26], (ii) serving as a stable basis for understanding, specifying and modeling requirements for enterprise applications [26], as well as for developing several types of enterprise-related applications [3], (iii) assisting enterprise knowledge acquisition, representation, and management [16, 19], (iv) supporting linked data publishing of enterprise information [27], (v) supporting enterprise application integration [1, 6, 25], among others.

Concerning the enterprise domain coverage, among the broadest ontologies are the TOVE Ontology and The Enterprise Ontology. The TOVE Ontology Project[1] addresses aspects related to activities, resources, organizations, products and requirements, compliance to the ISO 9000 standards, and costs. The Enterprise Ontology [26], in turn, covers aspects related to activities, plans, resources, organizations, strategy, and marketing.

[1] http://www.eil.utoronto.ca/enterprise-modelling/tove/

A deep presentation of the various types of entities countenanced by existing enterprise ontologies is outside the scope of this paper. Our purpose with this short summary of existing proposals is simply to call attention for the complexity of defining an all-encompassing reference model in this domain. In fact, the construction of an enterprise ontology is a challenging task, that requires significant time and cost to be accomplished [3, 20].

An alternative to the inherent complexity of the enterprise domain is to build a core enterprise ontology that is designed to allow domain-specific extensions [3, 21, 27]. Core ontologies provide a precise definition of structural knowledge in a specific field that spans across different application domains in this field [24]. Thus, a core enterprise ontology should capture the general business concepts, relations and properties, common to the majority of enterprises, independently of the specific activity field. This core ontology can be extended to specific domains, such as banking, education, manufacturing and so on [21]. The Core Enterprise Ontology (CEO) [3] and the W3C Organization Ontology (ORG) [27] are examples of core ontologies for the enterprise domain.

The rationale underlying core enterprise ontologies is that it is difficult to design a precise and comprehensive enterprise ontology, and that, in modeling a specific industry sector or even a specific enterprise, it is useful to start with a few, well established set of general concepts that will guide business experts in defining their own enterprise ontology [3]. However, depending on their scope, even core enterprise ontologies can be complex, and thus hard to reuse. In such cases, ontology patterns arise as a promising alternative to organize core ontologies [7].

According to [8], "an Ontology Pattern (OP) describes a particular recurring modeling problem that arises in specific ontology development contexts and presents a well-proven solution for the problem". In this paper, we are more interested in Domain-related OPs (DROPs), which are reusable fragments extracted from domain ontologies. DROPs should capture the core knowledge related to a domain, and thus they can be seen as fragments of a core ontology of that domain [8]. A domain ontology typically results from the composition of several OPs, with appropriate dependencies between them, plus the necessary extensions based on specific needs [4].

However, in order to truly favor reuse, organizing DROPs in catalogues is not enough. In a conventional catalog, there is a lack of a stronger sense of connection. We need something stronger than simply knowing that another pattern in the collection is related in some way. A pattern language provides this stronger sense of connection, since it expresses several types of relationships among patterns, such as relations of dependence, temporal precedence of application, or mutual exclusion between them. This is especially important for reusing DROPs, and thus ontology pattern languages are an improved way to organize DROPs [7].

An Ontology Pattern Language (OPL) aims to provide holistic support for using DROPs in ontology development for a specific domain. It provides explicit guidance on what problems can arise in that domain, informs the order to address these problems, and suggests one or more patterns to solve each specific problem. Moreover, an OPL supports the explicit consideration of complementing or conflicting pattern combinations to solve a given problem, along with guidelines for integrating patterns into a concrete ontology conceptual model. To ensure a stable and sound application of patterns, the patterns are presented in a suggested application order. OPLs encourage the application of one pattern at a time, in the order resulting from the chosen paths through the language [7].

In summary, an OPL gives concrete and thoughtful guidance for developing ontologies in a given domain, addressing at least the following issues [7]: (i) What are the key problems to solve in the domain of interest? (ii) In what order should these problems be tackled? (iii) What alternatives exist for solving a given problem? (iv) How should dependencies between problems be handled? (v) How to resolve each individual problem most effectively in the presence of its surrounding problems?

In the next section we present an initial version of an OPL for the enterprise domain. This version has a somehow limited coverage, but it can be extended to incorporate other enterprise aspects. As pointed by Buschmann et al. [5], pattern languages, in general, should be considered as a work in progress, and are subjects to continuous revision, enhancement, refinement, and completion.

3. E-OPL: AN ENTERPRISE ONTOLOGY PATTERN LANGUAGE

Figure 1 shows a UML activity diagram giving an overview of the current version of the E-OPL. As suggested in [7], in this activity diagram, Domain-related ontology patterns (DROPs) are represented by action nodes (the labeled rounded rectangles). Initial nodes (solid circles) are used to represent entry points in the OPL, i.e., DROPs in the language that can be used without solving other problems first. Control flows (arrowed lines) represent the admissible sequences in which DROPs can be used. Decision nodes (diamond-shaped symbols) represent alternative paths in the OPL. Fork nodes (line segments) are used to represent independent and possibly parallel paths. Finally, an extension to the original UML notation (dotted lines with arrows) is used to represent variant patterns, i.e. patterns that can be used to solve the same problem in different ways.

The enterprise aspects addressed by the current version of E-OPL are: Organization Arrangement, Team Definition, Institutional Roles, Institutional Goals, and Human Resource Management. The patterns in E-OPL were extracted from other enterprise ontologies, including specific ones, such as the software enterprise ontology presented in [2]. Moreover, the Unified Foundational Ontology (UFO) [13, 14, 15] was used to ground them. Every identified pattern was first analyzed in the light of UFO, then represented in OntoUML, and finally incorporated to E-OPL. OntoUML was used to represent the patterns because it is a UML profile that enables making finer-grained modeling distinctions between different types of classes and relations according to the ontological distinctions put forth by the foundational ontology UFO-A (an ontology of endurants) [13].

As Figure 1 shows, E-OPL has two entry points. The ontology engineer should choose one of them, depending on the scope of the specific enterprise ontology being developed. When the requirements for the new enterprise ontology being developed include only problems related to the definition of project teams, the starting point is EP2. Otherwise, the starting point is EP1. In this case (EP1), first the ontology engineer should address problems related to how an organization is structured. One of the three following patterns has to be selected: SOAR, COAR or MOAR.

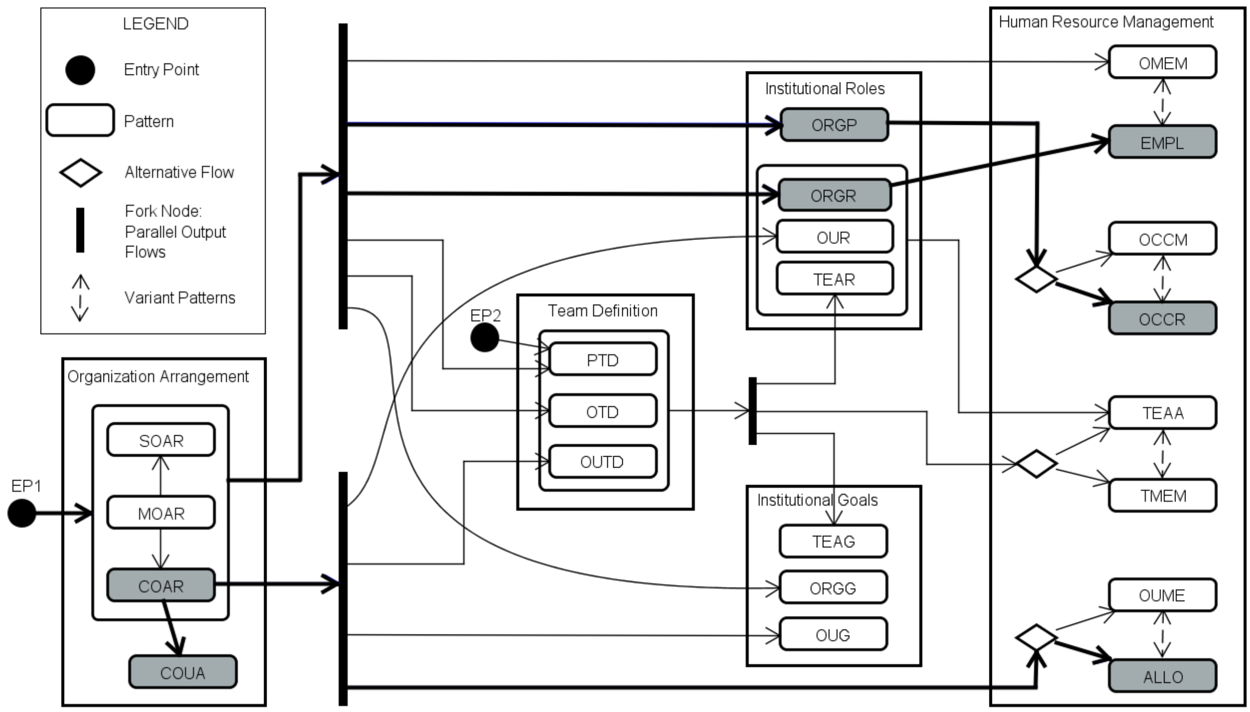

Figure 1 – Enterprise Ontology Pattern Language (E-OPL)

SOAR (Simple Organization Arrangement) should be selected as the first pattern if the ontology engineer needs only to represent very simple standalone organizations, which are neither composed of other organizations nor composed of organizational units. COAR (Complex Organization Arrangement) should be selected as the first pattern if the ontology engineer needs to represent only complex standalone organizations, which are not composed of other organizations, but that are composed of organizational units. When COAR is selected, COUA (Complex Organizational Unit Arrangement) can be used in the sequel, if there is a need to represent complex organizational units, which are composed of other organizational units. Finally, MOAR (Multi-Organization Arrangement) should be selected as the first pattern if the ontology engineer needs to represent organizations that are composed of other organizations. When MOAR is used, the ontology engineer can also use SOAR and COAR to address the organizational structure of the standalone organizations that compose a multi-organization. Figure 2 shows the four patterns related to organization arrangements. The stereotypes shown in this model are those defined in OntoUML. Note that, in E-OPL, the patterns are presented separately. However, when combined, they also form consistent models.

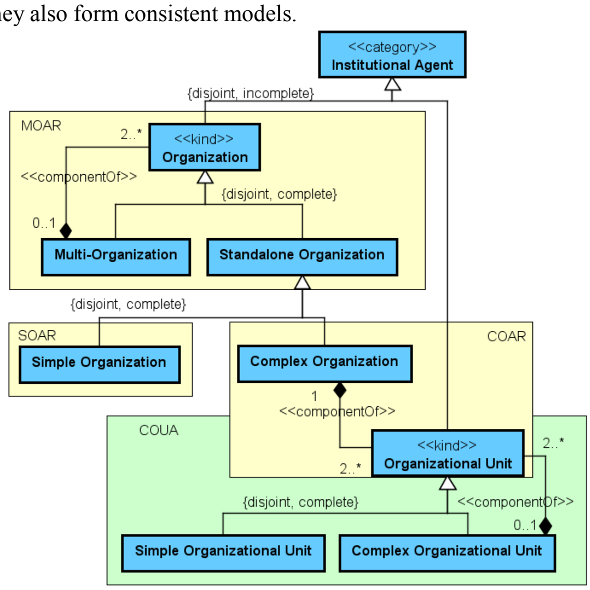

Figure 2 – Organization Arrangement patterns

Once problems related to the organization arrangement are addressed, the ontology engineer can treat problems related to the definition of organizational teams, goals and roles, and some problems related to human resource management.

Concerning team definition, three types of teams are considered: organizational teams (OTD), organizational unit teams (OUTD), and project teams (PTD). The Project Team Definition (PTD) pattern deals with teams that are defined with the specific purpose of performing a project. For this reason, PTD does not require that problems related to organizational arrangement are addressed prior to its use, and thus, it is an entry-point (EP2) in E-OPL.

Regarding goals, in a general view, we consider that Institutional Agents (a generalization for Organizations, Organizational Units and Teams) may define Institutional Goals, and three patterns are available: ORGG (Organizational Goals), OUG (Organizational Unit Goals), and TEAG (Team Goals).

Concerning roles, we consider that Institutional Agents (Organizations, Organizational Units and Teams) may define Institutional Roles. Like the TOVE Ontology [10], we consider two main types of Institutional Roles: Positions and Human Resource Roles. A Position represents some formal position in the organization, such as "president", "sales manager", etc. A Human Resource Role defines a prototypical function of a person in the scope of an Institutional Agent, such as "engineer" or "system analyst". Moreover, we distinguish between formal and informal roles. Formal Human Resource Roles are those recognized by the whole organization and its environment (partners and society in general). Informal Human Resource Roles are those recognized only in the scope of the corresponding institutional agent. Team Roles and Organizational Unit Roles are types of informal roles, recognized, respectively, by a Team and by an Organizational Unit. Organizational Roles can be formal or informal. Formal Organizational Roles are those considered when employments are created. Each employment is made for a specific formal role. On the other hand, a particular person, in the same employment, can assume several informal roles.

In order to deal with institutional roles, four patterns were defined. ORGP (Organizational Positions) deals with positions defined in an organization. ORGR (Organizational Roles) addresses both formal and informal roles defined in an organization. OUR (Organizational Unit Roles) and TEAR (Team Roles) concern informal roles defined by an organizational unit or a team, respectively. Figure 3 shows the patterns related to organizations, namely ORGP and ORGR. We should highlight that, for representing these patterns, we had to extend OntoUML metamodel by introducing the stereotype ⟨⟨hou⟩⟩ to represent high-order universals, i.e. rigid sortals whose instances are types. A concept stereotyped as high-order universal is, in fact, a powertype that will be used as the generalization set of a hierarchy of concepts in a specific enterprise ontology.

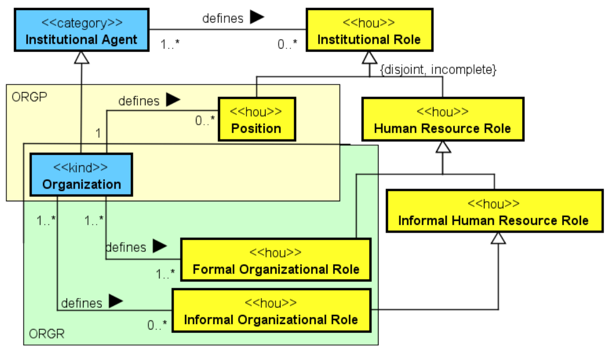

Figure 3 – Organizational Positions (ORGP) and Organizational Roles (ORGR) patterns.

Finally, problems related to human resource management can be addressed. We defined eight patterns treating four material relations that involve human resources, namely: employment, allotment, team allocation, and occupation. Since material relations are derived from relators[2] [13], each one of these

---

[2] Material relations are relations that have material structure on their own. The relata of a material relation are mediated by individuals that are called relators. Relators are complex objectified relational properties [13].

material relations is addressed by two patterns: one explicitly including a concept representing the relator, and another disregarding the relator and considering only the material relation. The patterns considering the relators are more complete. They allow capturing information about the relator (for instance, start date and end date of an employment, allotment, allocation, or occupation). On the other hand, if for a given context, such information is considered irrelevant, then the ontology engineer can choose simpler patterns, which disregard the relators, capturing only the fact that human resources are members of (⟨⟨memberOf⟩⟩) organizations, organizational units, and teams.

Figure 4 shows the two variant patterns addressing employments in organizations. The OMEM (Organization Membership) pattern considers only the material relation ⟨⟨memberOf⟩⟩ between Human Resource and Organization. The EMPL (Employment) pattern considers the relator Employment connecting, via mediation relations, a Human Resource to the Organization that employs him/her, and the Formal Organizational Role for which he/she is employed. This pattern allows also representing properties of the employment, such as start and end dates.

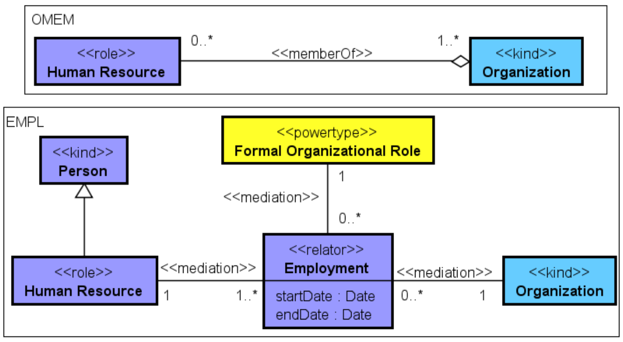

Figure 4 – Employment variant patterns: OMEM and EMPL.

Analogously, patterns are defined for treating organizational unit allocation, position occupation and team allocation. The OUME (Organizational Unit Membership) pattern considers only the material relation ⟨⟨memberOf⟩⟩ between Human Resource and Organizational Unit, while the ALLO (Allotment) pattern considers the relator Allotment connecting a Human Resource to an Organizational Unit. OCCM and OCCR address position occupation. OCCM considers only the material relation "occupies" between Human Resource and Position, while OCCR considers the relator Occupation and the corresponding mediation relations. Regarding team allocation, TMEM (Team Membership) considers only the material relation ⟨⟨memberOf⟩⟩ between Human Resource and Team, while TEAA (Team Allocation) considers the relator Team Allocation and three mediation relations between this relator and Human Resource, Team and the Human Resource Role that the human resource plays in that team.

Next, we discuss the use of E-OPL for building an enterprise ontology for the Government Brazilian Universities. Line paths depicted with thicker lines and patterns depicted in grey in Figure 1 show how we used E-OPL for developing this ontology.

4. APPLYING E-OPL

In the last years, the Brazilian Government has conducted several programs aiming at expanding university education in Brazil. These programs require investments, as well as more planning and monitoring by the universities in order to meet the established goals. In addition, since the promulgation of the Law nº 12,527, which regulates the manner and the timetable for the information to be given by the State, several initiatives have been carried out to obtain, organize and make available information in an accessible and computable format. This scenario has motivated us to build an enterprise ontology on Governmental Brazilian Universities, which can be used to improve information management, and to promote a better access to data. Our focus is on educational issues, and thus for this version of the Government Brazilian University Ontology (University Ontology), we have considered the following competency questions:

CQ1. Which is the university structure regarding units involved in education?

CQ2. Which are the roles and positions involved in this context?

CQ3. Which are the university members playing those roles and occupying those positions?

Since we are interested in describing the university structure (CQ1), we shall enter the E-OPL through entry point EP1. Universities are autonomous, standalone organizations, hierarchically organized in centers and departments. Thus, we selected COAR pattern as the first pattern to be applied. By applying this pattern, ***University*** is considered a subtype of **Complex Organization**. Next, we applied the COUA pattern. ***University Center*** is a subtype of **Complex Organizational Unit** and it is composed of ***Departments***, a subtype of **Simple Organizational Unit**. The model fragment in the upper right of Figure 5 results from applying the COAR and COUA patterns.

Once the competency question related to the organizational structure of universities is addressed, we can treat CQ2. In order to address CQ2, we selected the ORGP and ORGR patterns. By applying ORGP, we define ***University Position*** as subtype of **Position**, representing positions defined by Universities. In a similar way, the pattern ORGR was applied, giving rise to ***University Formal Role***, which extends **Formal Organizational Role**. Both ***University Position*** and ***University Formal Role***, as high-order universals, are *powertypes* that are to be used to indicate the criteria for the corresponding generalization sets. Thus, their instances (the positions Rector, Center Director, Department Head and Course Coordinator, as well as the formal role Professor) are to be taken as subtypes of the corresponding agents that play these roles (in the sense of UFO).

Finally, in order to treat CQ3, we selected the following patterns: EMPL, ALLO and OCCR. By applying EMPL, we define ***University Employment*** as a subtype of **Employment**. ***University Employment*** is the relator connecting a ***University Employee*** to its ***University*** in order to play a ***University Formal Role***. ***University Employee*** is subtype of Human Resource, being further extended by ***Professor***, the only role of interest, since we are focusing on educational issues. ***Student*** is also an important role for us, but, since students are not employed in the University, they are not university employees. Thus, we defined ***Student*** as a role played by a person when she/he is enrolled in a ***Course***. As Figure 5 shows, we defined the relator ***Enrollment***, which connects ***Students*** to their ***Courses***.

Professors are allotted in ***Departments***. By using the ALLO pattern, we define ***Professor Allotment*** as a subtype of **Allotment**, connecting ***Professors*** and ***Departments***.

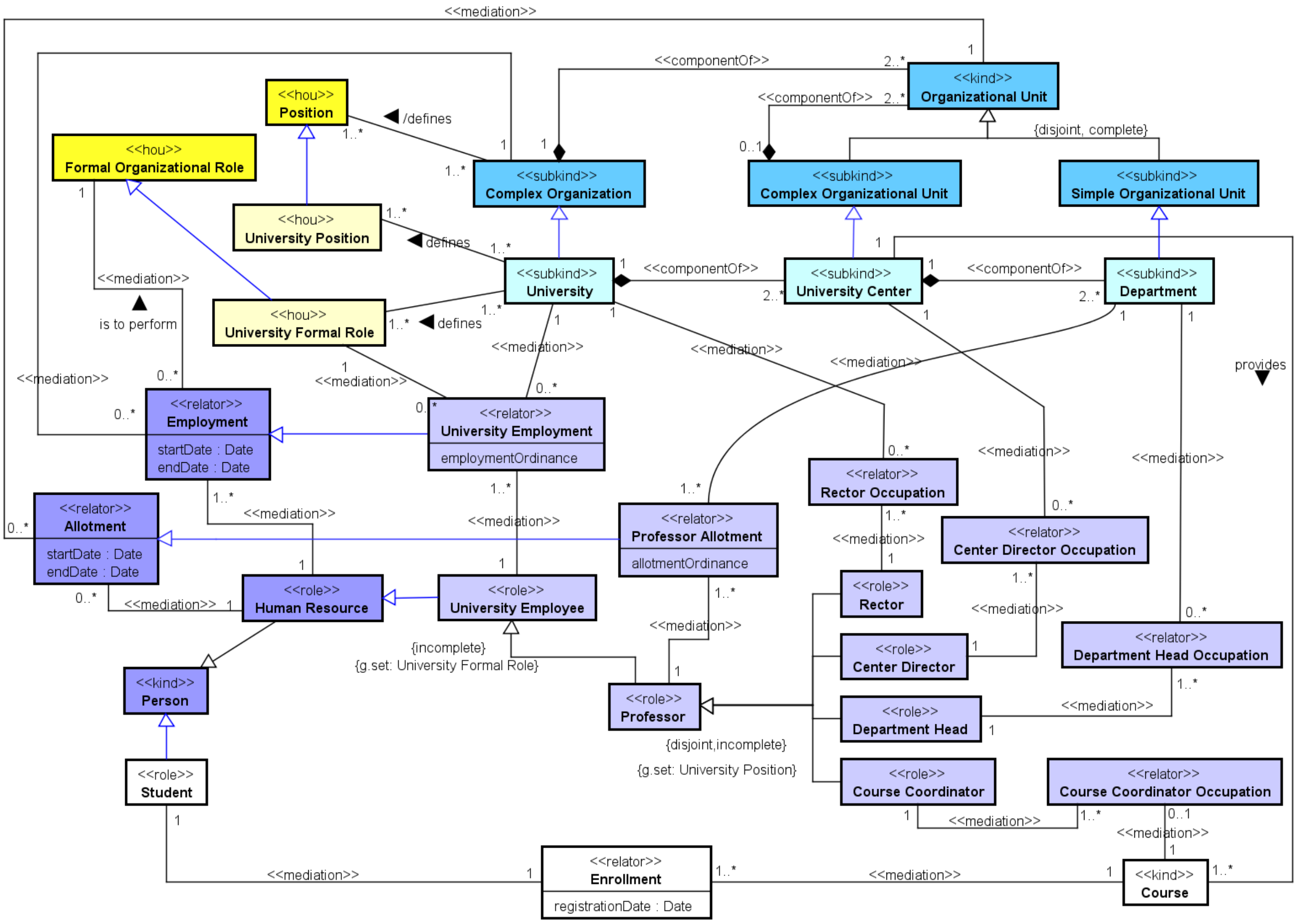

Figure 5 – The University Ontology produced from the Enterprise OPL.

Lastly, regarding position occupations, the pattern OCCR was applied four times to represent the occupation of the four distinct positions existing in the context considered: Rector, Center Director, Department Head and Course Coordinator. These four positions give rise to four roles (in the sense of UFO), extending the role ***Professor***, since all these positions can only be occupied by professors. In each situation, we have a specific occupation relator linking the corresponding entities, namely: ***University*** (to ***Rector***), ***University Center*** (to ***Center Director***), ***Department*** (to ***Department Head***) and ***Course*** (to ***Course Coordinator***). Although not shown in Figure 5, the specific occupation relators (***Rector Occupation***, ***Center Director Occupation***, ***Department Head Occupation*** and ***Course Coordinator Occupation***) are specializations of the relator **Occupation** of OCCR pattern.

We should highlight that, besides extending the patterns, we introduced three concepts that are domain-specific, namely: ***Course***, which represents the courses provided by the university centers; ***Student***, which represents the role of a person enrolled in a course; and the ***Enrollment*** relator, which relates a ***Student*** to a ***Course***. These concepts, although not provided by the E-OPL, are important for the University Ontology.

In order to evaluate the resulting ontology, first the model of Figure 6 was syntactically verified using OLED[3] to check if it is a valid OntoUML model, i.e., if it meets UFO constraints. Since the ontology was built from E-OPL, which is already grounded in UFO, we could observe a sensible reduction of syntactical problems, which were solved. Then, we used OLED to transform the OntoUML model into an OWL specification, generating an operational version of the ontology. With the operational version in hands, we instantiated the ontology, populating it with data of a university. Then, we checked if the ontology was able to answer the competency questions, and also more elaborated questions, such as the students/professors proportion in the university, university centers, and departments. Based on this evaluation, it was possible to confirm that the ontology is consistent and that it is able to answer the proposed competency questions.

We should highlight, however, that both in the ontology checking using OLED and the subsequent generation of the OWL operational ontology (and consequently when querying it), high-order universal concepts were not considered, since OLED and OWL do not deal with second-order constructs.

---

[3] OLED (Ontology Light-weight EDitor) is an OntoUML editor used to edit, evaluate, simulate and transform OntoUML models (http://nemo.inf.ufes.br/en/ontoumlsupport).

5. RELATED WORKS

There are many enterprise ontologies published in the literature, most of them having a much greater coverage when compared to E-OPL. However, we should highlight that the problem we want to address is a different one, namely, enterprise ontology reuse. Thus, here we focus on comparing our work to others that aim at improving enterprise ontology development by means of reuse.

Both CEO [3] and ORG [27] are core enterprise ontologies that aim at capturing the most general enterprise concepts, relations and properties, and that are to be extended to specific kinds of enterprises. In this sense, they are also concerned with reuse.

CEO [3] provides four groups of domain concepts (Passive Entities, Active Entities, Transformations and Conditionals) and six meta-relations (refinement, decomposition, predication, relatedness, similarity and instantiation). The groups of concepts are, in fact, hierarchies of concepts. The meta-relations, as the name suggests, are not relations of the enterprise domain. They are closer to foundational relations. Thus, in our opinion, CEO does not properly describe the enterprise domain for purposes of reuse. E-OPL share several ideas that are defended in the CEO Project, in special the one that highlights the importance of building core ontologies grounded in foundational ontologies, and the use of these core ontologies for building ontologies for specific domains [21]. In this sense, E-OPL is a core enterprise ontology grounded in a foundational ontology (UFO) and that serves as basis for building domain-specific enterprise ontologies. On the other hand, E-OPL is based on ontology patterns, which is an important trend in Ontology Engineering [4, 8, 22].

ORG [27] is also a core ontology. It is, however, not exactly an enterprise ontology. It is a core ontology of organizations and, thus, it is more general than other enterprise ontologies. E-OPL, in turn, focuses on enterprises, a special kind of organizations. ORG is relatively simple, covering some important aspects of organizations. However, if in the future other aspects of organizations are incorporated to ORG, without any clear modularization criterion, it tends to become complex. Since E-OPL is organized as a pattern language, we believe it can be expanded in an easier way, since the patterns provide a modularized way to address different aspects of the enterprise domain, without losing the holistic view. In fact, this is, in our view, the striking feature of E-OPL.

O'Leary [20] follows a different approach for ontology reuse, namely, an activity theory-based approach for generating enterprise ontologies. Activity theory provides a template-based approach for capturing the context of individual activities in an organization. It uses eight key class concepts: activity, outcome, subject, object, community, rules, tools and division of labor. In a nutshell, according to this view, *in an activity, the subject modifies an object to generate an outcome, while using a tool, in the context of a community with its corresponding rules and division of labor*. O'Leary's activity theory-based approach, as the name suggests, focuses on activities and thus says little about other aspects of the enterprise domain. In fact, activity theory provides a guide for ontology development by pointing out key theory-based issues in its template-based structure. E-OPL is also based on a theory, in fact, a set of theories given by the UFO ontology. Several of the aspects addressed by the activity theory are also considered in UFO (see, for instance, [14]), and we intend to incorporate in a near future other patterns to E-OPL addressing activity-related aspects of enterprises.

Concerning patterns languages applied to the enterprise domain, Kotzé et al. [18] proposed a pattern framework for enterprise architecture, called PF4EA. PF4EA is a method aiming at guiding the development of patterns and pattern languages for the Enterprise Architecture domain. Kotzé et al. illustrate the use of PF4EA for developing a pattern language whose intent is to assist novice enterprise architects in the development and maintenance of enterprise architectures. The given example does not present the components of a complete pattern language, but is merely for illustrative purposes. This work is related to ours, since both works are interested in pattern languages for the enterprise domain, recognizing that the notions of patterns and pattern languages should be applied to the enterprise domain. However, E-OPL is an ontology pattern language, while [18] focus on conventional pattern languages.

6. FINAL CONSIDERATIONS

Currently, reuse is recognized as an important practice for Ontology Engineering. Ontology patterns are considered a promising approach that favors reuse of encoded experiences and good practices in Ontology Engineering [22]. Moreover, core ontologies organized as Ontology Pattern Languages (OPL) have potential to amplify the benefits of ontology patterns [7]. Agreeing with these statements, we developed an initial version of E-OPL, an Enterprise OPL. As a proof of concept of the utility of E-OPL, we developed a domain specific enterprise ontology, the University Ontology. As we could observed in the development of the University Ontology reported here, the use of pattern language based on a well-founded core ontology such as E-OPL tends to bring the following benefits to the development of domain ontologies: (i) the resulting ontology tends to contain less consistency mistakes given that many of the potentially recurring source of inconsistencies in the enterprise domain tend to be solved by the basic patterns of the core ontology; (ii) the development process of the derived domain-specific enterprise ontologies tends to be accelerated by the massive reuse of modeling fragments and decisions embedded in the patterns of the language; and (iii) E-OPL guides pattern selection, also facilitating combining them. Although we perceived these benefits, real case experiments have to be conducted to truly confirm them.

As an ongoing future work, we intend to enlarge E-OPL by adding new patterns. For instance, we are currently working on patterns to deal with business contracts and with human resources competences and skills. Other patterns, more related to planned and executed activities are being investigated by extracting patterns from UFO. Moreover, new applications of E-OPL can provide relevant feedback for improvements.

7. ACKNOWLEDGMENTS

This research is funded by the Brazilian Research Funding Agency CNPq (Process Numbers 485368/2013-7, 310634/2011-3, and 311578/2011-0).

8. REFERENCES

[1] Alazeib, A., Balogh, A., Bauer, M., Bouras, A., Friesen, A., Gouvas, P., Mentzas, G., Pace, A. Towards semantically-assisted design of collaborative business processes in EAI scenarios. In *Proceedings of the 5th IEEE International Conference on Industrial Informatics* (Vienna, Austria, June 2007), vol. 2, 2007, 779-784.

[2] Barcellos, M. P., Falbo, R. A. Using a foundational ontology for reengineering a software enterprise ontology. In *Proceedings of the of the ER 2009 Workshops 2009* LNCS 5833 (2009) 179-188.

[3] Bertolazzi, P., Krusich, C., Missikoff, M. An approach to the definition of a core enterprise ontology: CEO. In *Proc. of the International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations.*2001.

[4] Blomqvist, E., Gangemi, A., Presutti, V. Experiments on pattern-based ontology design. In *Proc. of Fifth Int. Conference on Knowledge Capture.* 2009, 41-48.

[5] Buschmann, F., Henney, K., Schmidt, D.C. *Pattern-Oriented Software Architecture Volume 5: On Patterns and Pattern Languages*, John Wiley & Sons Ltd., 2007.

[6] Dietz, J. *Enterprise Ontology: Theory and Methodology*. Springer, 2006.

[7] Falbo, R. A., Barcellos, M.P., Nardi, J.C., Guizzardi, G. Organizing ontology design patterns as ontology pattern languages. In *Proceedings of the 10th Extended Semantic Web Conference - ESWC 2013* (Montpellier, France) 2013.

[8] Falbo, R. A., Guizzardi, G., Gangemi, A., Presutti, V. Ontology patterns: clarifying concepts and terminology. In *Proceedings of the* 4th *Workshop on Ontology and Semantic Web Patterns* (Sidney, Australia), 2013.

[9] Fox, M. S. The TOVE Project: A common-sense model of the enterprise. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. *Lecture Notes in Artificial Intelligence*, 604, 1992, 25-34.

[10] Fox, M. S., Barbuceanu, M., Gruninger, M., Lin, J. An organization ontology for enterprise modelling. *Simulating Organizations: Computational Models of Institutions and Groups*, M. Prietula, K. Carley & L. Gasser (Eds), Menlo Park CA: AAAI/MIT Press, 1997, 131-152.

[11] Fox, M. S., Gruninger, M. On ontologies and enterprise modelling, In *Proceedings of the International Conference on Enterprise Integration and Modeling Technology* (Torino, Italy, October 1997), 1997, 190-200.

[12] Geerts, L. G., McCarthy, W. E. The ontological foundation of REA enterprise information systems. In *American Accounting Association Conference Annual Meeting* (Philadelphia, USA), 2000.

[13] Guizzardi, G. *Ontological Foundations for Structural Conceptual Models*, Universal Press, The Netherlands, 2005.

[14] Guizzardi, G., Falbo, R.A., Guizzardi, R.S.S. Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology. In *Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments* (Recife, Brazil), 2008, 244-251.

[15] Guizzardi, G., Wagner, G., Falbo, R.A., Guizzardi, R.S.S, Almeida, J.P.A. Towards Ontological Foundations for the Conceptual Modeling of Events. Proceedings of the 32th International Conference on Conceptual Modeling (ER 2013), Hong-Kong, China, 2013. 327-341.

[16] Han, K. E., Park, J. W. Process-centered knowledge model and enterprise ontology for the development of knowledge management system. *Expert Systems with Applications* 36, 4 (May 2009), 7441-7447.

[17] ISO 15704:2000 Industrial automation systems -- Requirements for enterprise-reference architectures and methodologies, 2000.

[18] Kotzé, P., Tsogang, M., Van der Merwe, A. A Framework for Creating Pattern Languages for Enterprise Architecture. In Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation. Springer Berlin Heidelberg, 2012. 1-20.

[19] Maedche, A., Motik, B., Stojanovic, L., Studer, R., Volz, R. Ontologies for enterprise knowledge management. *IEEE Intelligent Systems* 18, 2 (Mar-Apr 2003), 26-33.

[20] O'Leary, D. E. Enterprise ontologies: review and an activity theory approach. *International Journal of Accounting Information Systems,* 11, 4 (2010), 336-352.

[21] Partridge, C., Stefanova, M. *A synthesis of state of the art enterprise ontologies.* The BORO Program, LADSEB CNR, 2001.

[22] Presutti, V., Daga, E., Gangemi, A., Blomqvist, E. *eXtreme Design with Content Ontology Design Patterns*. In: Proc. Workshop on Ontology Patterns, 2009.

[23] Rajabi , Z., Minaei, B., Seyyedi, M. A. Enterprise architecture development based on enterprise ontology. *Journal of Theoretical and Applied Electronic Commerce Research*, 8, 2 (2013), 85-95.

[24] Scherp, A., Saathoff, C., Franz, T., Staab, S. Designing core ontologies. *Applied Ontology*, 6, 3, (2011) 177-221.

[25] Umar, A., Zordan, A. Enterprise ontologies for planning and integration of business: a pragmatic approach. *IEEE Transactions on Engineering Management*, 56, 2 (2009), 352-371.

[26] Uschold, M., et al. The enterprise ontology. *The Knowledge Engineering Review* 13, 1 (1998), 31-89.

[27] W3C. 2013. The Organization Ontology. W3C Candidate Recommendation 25 June 2013. Available in http://www.w3.org/TR/2013/CR-vocab-org-20130625/.