Transforming a Business and Information Need Goals' Ontology into a Formal Specification useful for a Strategy Pattern Recommender System

Belen Rivera, Pablo Becker and Luis Olsina

GIDIS_Web, Engineering School, UNLPam, General Pico, LP, Argentina,
[riveramb, beckerp, olsinal]@ing.unlpam.edu.ar

Abstract. From a computer science point of view, ontologies are aimed at getting knowledge from a certain domain and providing a consensual understanding of it. In the conceptualization stage, ontologies can be represented by means of UML models. However, UML models are not semantically machine processable. Therefore, during the implementation stage, ontologies expressed as UML models should be translated into formal languages, which allow inferring the ontology. In this work, we present the ontology transformation made from an UML conceptual specification to an OWL formal specification for the quality evaluation domain. This ontology represents concepts and relationships related with goals (both business and information need goals) at different organizational levels in addition to projects, strategies and strategy patterns, which help to achieve specific evaluation goal purposes. The final aim of the transformation is to have a shared knowledge about evaluation goals and strategy patterns that permit to instantiate the suitable strategy to carry out goal purposes. This will allow us to develop a strategy pattern recommender system, which can be useful during the strategy selection process when implementing quality measurement and evaluation projects.

Keywords. Business and Information Need Goals. Ontology. Strategy Pattern. OWL. Recommender System.

1 Introduction

Every organization that implements quality assurance activities should have aligned its measurement and evaluation (ME) information need and business goals at operational levels with business goals established at strategic levels. In this sense, Basili *et al.* [3] argue that a main issue in an organization is the lack of alignment between goals at strategic levels with those formulated at operational levels. Consequently, goals of ME projects should not be a final aim, but rather a key reason to reach information need and business goals in an organization [6].

Often for reaching a business goal, it is necessary to have additional information, which usually allows to know the extent to which a business goal is being achieved [19], and therefore justify results. This supporting goal is called an information need goal. A particular kind of an information need goal is the ME information need goal. This goal is operationalized by ME projects and gives added-value information, which is useful for satisfying its related business goals and therefore for fostering the decision making process. Thus, by linking information need goals with business goals at different organizational levels is essential for not being considered them as established in a fuzzy and isolated manner, but rather in the context of the organizational strategic business goals.

Furthermore, in order to reach any business or information need goal successfully, it is necessary to use the suitable strategy. A strategy indicates what to do and how to do it by defining a set of well-established activities and methods. Therefore, for a given business goal, the suitable strategy should be selected considering the evaluation goal purpose (such as to understand, improve, monitor) and the amount of quality views [20].

In [19], we have presented a quality evaluation approach that considers the following aspects: *i*) the definition of multilevel business and information need goals; *ii*) the definition of different evaluation purposes and the consideration of quality views in ME information need goals; *iii*) the formulation of ME (and change, MEC) projects to operationalize goals; and *iv*), the adoption of strategy patterns for instantiating specific strategies that helps to achieve evaluation goal purposes.

Regarding the latter aspect, the reason of using strategy patterns is to provide a reusable solution to the problem of selecting the suitable strategy in a ME/MEC project. The selection of the strategy relies on key issues like the evaluation purpose of the goal, and the amount of quality views. Strategy patterns provide a generic course of action, which specifies the activities that have to be instantiated to reach the goal in addition to method specifications, which specify how to carry out the generic activities that the pattern provides. Additionally, both process and method specifications share a common domain conceptual base. As a consequence, the strategy pattern determines the concrete strategy to be instantiated as discussed in [20].

Regarding the abovementioned four aspects of the quality evaluation approach, they rely on conceptual bases that are structured as ontologies. These ontologies [19] link business and information need goal concepts with those related to projects, strategies, strategy patterns and non-functional requirements. So far, from the ontology development lifecycle [7] standpoint, we have performed the conceptualization stage. But the ontology implementation stage has been left apart. Having in mind that we are planning to develop a strategy pattern recommender system that suggests the suitable strategy pattern to fulfill a given evaluation business goal, we propose to fill this gap by accomplishing the formalization of the ontology.

Hence, the main contribution of this work is the transformation of the goal ontology, which was represented in an UML class diagram, into an OWL (*Web Ontology Language*) formal specification. The rationale for this transformation is to implement the ontology with machine-processable information about organizational goals, strategies and strategy patterns in order to enable the reuse and integration of knowledge about these sub-domains. Particularly, the resulting ontology implementation will allow to query and reason about the suitable strategy pattern to be selected regarding a specific evaluation goal. In order to evaluate the ontology and its scope, a couple of competency questions are listed. This will benefit ultimately the development of the strategy pattern recommender system.

The rest of this paper is organized as follow. Section 2 describes the specification and conceptualization stages of the organizational goals ontology documented in [19]. Section 3 presents the formalization of the ontology describing the steps related with the implementation and its evaluation. Then, Section 4 describes the utility of strategy patterns for a strategy pattern recommender system. Section 5 presents related work devoted to organizational goals ontologies. Finally, Section 6 draws conclusions and future work.

2 Conceptual Description of the Organizational Goals Ontology

As for any software product, the ontology development lifecycle should be guided by a process. Ideally, this process should allow the formalization of the domain and its consensus, either by the knowledge captured from well-known sources or by the domain expert discussions about the concepts, properties, relations and constraints that integrate the ontology or both at the same time. Considering the implementation stage, the ontology should be also formalized or implemented. Further, it has to be evaluated by means of the verification and validation of competency questions [5].

For the development of our organizational goals ontology, we considered many of the stages established in the process of the METHONTOLOGY approach [7]. This approach identifies a set of stages though which the ontology moves during its life cycle such as the specification, conceptualization, formalization, integration, implementation and maintenance stages. In the following subsections, we discuss its first two main activities, *Specification* and *Conceptualization,* considering the abovementioned organizational goals ontology.

2.1 Specification

During the specification stage, it is necessary to understand the domain to be modeled in order to define the ontology's purpose and scope in addition to determine the sources of knowledge to be used. In our case, the domain is the one related to organizational business and information need goals, projects and strategies.

Therefore, the scope of the ontology embraces the business and information need goal, project and strategy key concepts which were relevant to be linked with non-functional requirements terms of the previously developed ontology of metrics and indicators [15].

A way of determining and validating the ontology scope is by designing a set of competency questions to which the ontology should answer. In this sense, we have identified two main competency questions, namely:

1) *Which is the suitable strategy pattern, given a business/information need goal with an evaluation purpose and a certain amount of quality views?*
2) *Which is the suitable quality view for a given entity super-category and quality focus?*

The core sources of information used to have, to a certain extent, adherence or contrast when defining the concepts to be included were the following papers and standards: the Barcellos *et al.* papers titled "*A Well-Founded Software Process Behavior Ontology to Support Business Goals Monitoring in High Maturity Software Organizations*" [2] and "*A Strategy for Preparing Software Organizations for Statistical Process Control*" [1]; the Basili *et al.* proposal known as GQM^+Strategies, which is documented in several works such as for example "*Linking Software Development and Business Strategy through Measurement*" [3]; the Guizzardi *et. al.* work about foundational ontologies viz. "*Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology*" [9]; the ISO/IEC 15939 [10], 25010 [11] and 9126 [12] standards; the CMMI (*Capability Maturity Model Integration Dev. v.1.3*) *de facto* standard [6]; the *Business Motivation Model* (BMM) [16] proposed by OMG; the PMBOK (*Project Management Body of Knowledge*) guidebook [18]; the Sing and Woo work titled "*A Methodology for Discovering Goals at Different Organizational Levels*" [21]; the Goal-Driven Measurement approach described in [17], in addition to the vocabulary given in the non-functional requirement component, which is part of the metrics and indicators ontology documented in [14, 15].

2.2 Conceptualization

From the above cited sources of knowledge, and taking into account the ontology scope and purpose, we documented in [19] the conceptual model of the organizational goals ontology as shown in Fig. 1.

The conceptualization stage has been done by modeling the domain with an UML class diagram, i.e., the terms, attributes and relationships involved in the business and information need goals domain that we have considered as the minimum but necessary set of concepts. As the full description of the ontology is not the main aim of this paper, the interested reader can look for the definition of all terms, attributes and relations in the quoted work. Nevertheless, in the remaining part of this subsection, some of these terms are described (which are highlighted in italic when they appear the first time in the text), mainly those that are important to understand the foundations of the strategy pattern recommender system.

In the organizational goals ontology, we have represented that a *Business* goal can be formulated at different *Organizational Levels* such as *operational*, *tactical* and *strategic*. In turn, a business goal can be divided into business *sub-goals*. In addition, an *Information Need g*oal is a support goal to a given business goal. Usually, an information need goal gives useful information to know in which degree a business goal has been achieved. In turn, an information need goal may require *ME Information Need* goals. A ME information need goal is a more specific kind of goal, which is driven by ME activities.

A ME information need specifies an object (*Entity Category* that belongs to an *Entity Super-Category*) to be evaluated and, at the same time, describes a *Quality Focus.* A quality focus is the root characteristic or *Calculable Concept* to be evaluated. In turn, calculable concepts (i.e., characteristics and sub-characteristics) are included in a *Quality Concept Model* such as for example, the quality models presented in the ISO 25010 [11] standard. Note that a key term that associates an entity super-category with a quality focus is the Q*uality View* [20] concept (see the *quality view* component in Fig. 1). Example of quality views are the system quality view, which relates the system entity super-category with the external quality focus. Or the product quality view, which relates the product entity super-category with the internal quality focus. Quality views and their *influences* and *depends on* relationships are based on the quality perspectives that the ISO 9126 and ISO 25010 describe.

Besides, an organization arranges work by means of *Projects* which allow to *operationalize* the established business goals. A project adopts a *Project Life Cycle,* which establishes the stages the project goes through from its start to its end. The project lifecycle involves at least *Resources* and *Work Definitions* [4] and uses strategies. A *Strategy* is a resource that *helps to achieve* a goal. Specially, we are interested on evaluation strategies, intended to help reaching ME information need goals that are operationalized by ME/MEC projects. These strategies may be instantiated from a *Strategy Pattern* that includes a reusable and customizable solution to a recurrent ME/MEC project problem in similar situations.

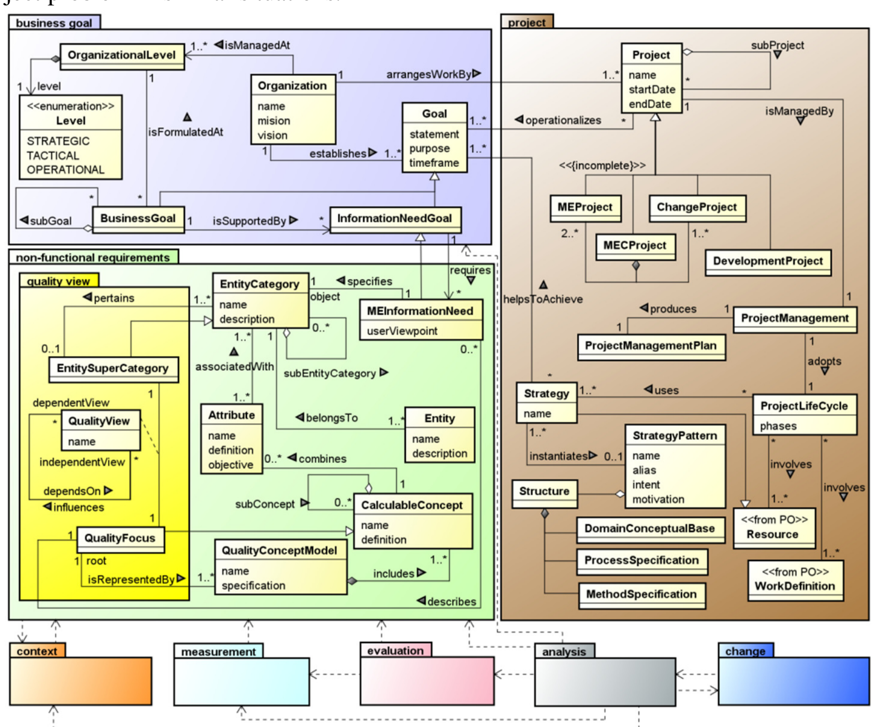

Fig 1. Key concepts from the business goal domain (business goal package) and related relationships with some concepts for the project and non-functional requirements domains (project and non-functional requirements packages). Note that PO stands for Process Ontology [4].

A strategy pattern has a *Structure* compound of three integrated pillars, namely: i) a *domain conceptual base*, ii) *process specifications*, and iii) *method specifications*. The domain conceptual base embraces a terminological base for a given domain, e.g., the ME/MEC domain. The second aspect describes what to do by means of specifying the activities in the process to be considered. The third aspect represents how an activity should be carry out using method specifications based on procedures and rules.

3 Formal Specification of the Organizational Goals Ontology

The above Section dealt with the first two stages for the organizational goals ontology development lifecycle taking into account the METHONTOLOGY approach. But having the ontology represented as an UML conceptual base neither enable for its semantic processing nor its dynamic reuse by other ontologies and agents.

As a listed contribution in the Introduction Section, in the current work, some challenges for the ontology transformation we faced are discussed. Particularly, for the transformation made from the UML conceptual specification to the OWL formal specification. So in the following two subsections the implementation and evaluation stage of the organizational goals ontology are described.

3.1 Implementation

For the conversion of the UML conceptual model into the OWL language, we used the UMLToOWL tool [8]. This tool transforms an UML class diagram, built specifically with the *Visual Paradigm*, *Microsoft Visio 2010* or *ArgoUML* case tools, into OWL DL (*Description Logic*) ontologies, fully compatible with the *Protegè* ontology editor. For this, it transforms each UML element such as a concrete or abstract class, interface, attribute, relationship and comment into OWL elements.

Therefore, the first step we did was redesigning the conceptual model shown in Fig. 1 using the *ArgoUML* tool and then exported it as XMI (*Xml Metadata Interchange*) format. With this XMI file of the ontology, we used the UMLToOWL tool to produce the OWL specification. Fig. 2 illustrates the OWL definition of the *QualityView* class, showing that each individual (instance) of a quality view must participate in a relation between an entity super-category (see the highlighted *hasQualityViewEntitySuperCategoryRelation* relation) and a quality focus (*hasQualityViewQualityFocusRelation* relation) as it was described in subsection 2.2 (look at the quality view component in Fig. 1).

For the further step of evaluation achieved by means of the competency questions, it was also necessary to create instances as members of a class and implement them as individuals of the ontology. For example, an instance created was *System Quality View,* represented in Fig. 3. This instance is an individual of the *QualityView* class and has two object properties (relations): the *hasQualityViewEntitySuperCategoryRelation* object property with the *System* value, and the *hasQualityViewQualityFocusRelation* object property with the *External Quality* value.

```xml
<!-- http://www.tuwien.ac.at/out.owl#QualityView -->

<owl:Class rdf:about="http://www.tuwien.ac.at/out.owl#QualityView">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.tuwien.ac.at/out.owl#"/>
            <owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</owl:minQualifiedCardinality>
            <owl:onClass rdf:resource="http://www.tuwien.ac.at/out.owl#QualityView"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.tuwien.ac.at/out.owl#hasQualityViewQualityViewRelation"/>
            <owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</owl:minQualifiedCardinality>
            <owl:onClass rdf:resource="http://www.tuwien.ac.at/out.owl#QualityView"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.tuwien.ac.at/out.owl#hasQualityViewEntitySuperCategoryRelation"/>
            <owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
            <owl:onClass rdf:resource="http://www.tuwien.ac.at/out.owl#EntitySuperCategory"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.tuwien.ac.at/out.owl#hasQualityViewQualityFocusRelation"/>
            <owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
            <owl:onClass rdf:resource="http://www.tuwien.ac.at/out.owl#QualityFocus"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Fig.2. OWL specification fragment of the *Quality View* class.

```xml
<!-- http://www.tuwien.ac.at/out.owl#System_Quality_View -->

    <owl:NamedIndividual rdf:about="http://www.tuwien.ac.at/out.owl#System_Quality_View">
        <rdf:type rdf:resource="http://www.tuwien.ac.at/out.owl#EntitySuperCategory"/>
        <rdf:type rdf:resource="http://www.tuwien.ac.at/out.owl#QualityView"/>
        <out:hasQualityViewEntitySuperCategoryRelation rdf:resource="http://www.tuwien.ac.at/out.owl#System"/>
        <out:hasQualityViewQualityFocusRelation rdf:resource="http://www.tuwien.ac.at/out.owl#External_Quality"/>
    </owl:NamedIndividual>
```

Fig 3. OWL specification fragment of the *System Quality View* instance.

There are other tools that we had tested before choosing UMLToOWL. But we found some technical issues with them (just for space reasons, we cannot quote all the analyzed tools and raised technical issues). Also, we could have implemented the ontology from scratch, e.g., in *Protegè*. However, we found that the transformation from UML to OWL using the UMLToOWL tool was more effective and challenging.

3.2 Evaluation

After creating the instances, we proceeded to validate the ontology in the context of the evaluation stage. This step permitted us to infer new relations and validate the existent ones. To this end, we used the *Pellet* reasoner, which is integrated in *Protegè* and is characterized for being the first reasoner with descriptive logic in addition to be complete and robust to reason on DL ontologies [22].

For the ontology validation it was necessary to formalize the competency questions using the SPARQL language (*SPARQL Protocol and RDF Query Language*). In the specification stage (recall subsection 2.1) we enumerated the two designed competency questions, viz.: 1) *Which is the suitable strategy pattern, given a business/information need goal with an evaluation purpose and a certain amount of quality views?* and, 2) *Which is the suitable quality view for a given entity super-category and a quality focus?* Figure 4 illustrates the first competency question implemented in SPARQL, considering an evaluation goal with the *Improve* purpose and the *System Quality View*. For this case, the answer was that the GOMEC_1QV strategy pattern (fully specified in [20]), is the most suitable strategy pattern for instantiating the concrete strategy for the given goal. Thus, it yielded the valid answer, as expected.

Regarding the second competency question, Fig. 5 shows its implementation in SPARQL considering the *System* entity super-category and the *External Quality* focus. In this case, the answer was the *System Quality View*, due to it relates the *System* entity super-category and the *External Quality* focus, as discussed in [20].

Ultimately by considering these competency questions and more ontology instances, especially more evaluation purposes' and quality views' instances as well as strategy patterns' instances, a semantic web service can be developed. Hence, it can be able to process the queries in SPARQL over the ontology, and recommend the suitable strategy pattern for instantiating the concrete strategy for a given evaluation goal.

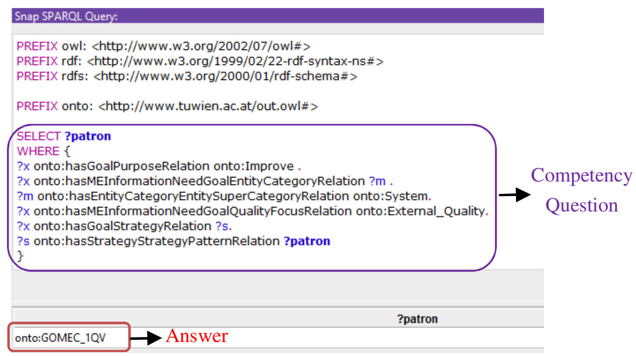

Fig. 4. SPARQL's first competency question implementation: *Which is the suitable strategy pattern, given a ME information need/business goal with an evaluation purpose and a certain amount of quality views?*

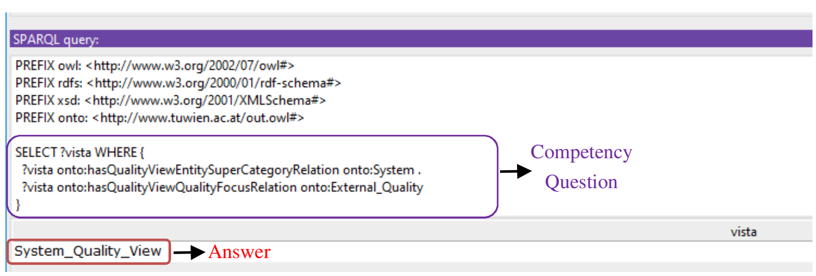

Fig. 5. SPARQL's second competency question implementation: *Which is the suitable quality view for a given entity super-category and a quality focus?*

4 Grounds for Building the Strategy Pattern Recommender System

The final aim for implementing the organizational goals ontology with a higher level of formalism is twofold. On one hand, to contribute with a common terminology related to organizational goals, non-functional requirements and quality views, projects, strategies and strategies patterns for the evaluation domain. Thus, it is able to have semantic-automated processing of the ME information for its correct interpretation. On the other hand, by querying and answering the competency questions can help in the development of a semantic web service to recommend the suitable strategy pattern from which the concrete strategy that helps to achieve a given goal is instantiated. In this Section, we briefly discuss the benefits of having strategy patterns when considering ME/MEC projects.

A strategy pattern could be seen as a general and reusable solution for recurrent problems in the evaluation domain. A strategy pattern gives a solution for instantiating integrated strategies, providing a well-established process with a set of activities to be customized and ME/MEC methods to carry out the activities. In subsection 2.2 we pointed out that we are interested on evaluation strategies, particularly those that are intended to help reaching evaluation business/information need goals.

An evaluation goal has a specific statement and purpose such as to understand, improve, monitor, among others. It also embeds in the statement one or more quality views (recall the non-functional requirements and quality views components in Fig. 1). For instance, an evaluation goal statement with one quality view is the following: "*Improve the usability of the X web application*". This statement involves the *Improve* purpose for the *System Quality View*, since it embeds the *Usability* characteristic, which pertains to the *External Quality* focus, and the entity *X web application*, which belongs to the *System* entity super-category. Considering the same improve purpose, another evaluation goal statement can be "*Improve the maintainability for the Y Java modules*", where the *Product Quality View* is included due to the *Maintainability* characteristic to be evaluated pertains to the *Internal Quality* focus, while the *Y Java modules* entity belongs to the *Product* entity super-category. For achieving both evaluation goals, the same strategy pattern should be recommended, hence customizing accordingly the concrete strategy activities. But if the evaluation goal statement involves two related quality views, for example, to improve the quality in use of an X web application by improving its external quality characteristics, other strategy should be use. Therefore, the amount and dynamic of activities to be perform are not the same whether one quality view is considered, (such as the system quality view) or if two related quality views are considered such as for example the system-in-use quality view and the system quality view. Clearly, two different strategy patterns emerge.

As a consequence, strategy patterns arise as a way of providing a solution in the instantiation of the suitable integrated strategy for fulfilling an evaluation goal. A set of strategy patterns has been identified considering different evaluation purposes and the amount of quality views. Looking at the above examples, for the case of the *improve* purpose and one quality view (it does not matter which specific quality view) the strategy pattern that prescribes a set of activities and methods for reaching the evaluation goal is named GOMEC_1QV [20]. Regarding the case where two quality views are involved for the same *improve* purpose, the suitable strategy pattern is GOMEC_2QV [20]. The solution (i.e., the activities' and methods' specifications) that both strategy patterns offer must be appropriately personalized by concrete strategies. In fact, the strategy that the GOMEC_1QV pattern instantiates is called GOCAMEC. Then, when using GOCAMEC, for instance it personalizes the name of each activity –regarding the specific quality view to be evaluated- that the strategy pattern GOMEC_1QV prescribes as part of the solution (see the example developed in [20]). For the case of the GOMEC_2QV strategy pattern, if the two related quality views are for example the system-in-use quality view and the system quality view, then the concrete strategy that this strategy pattern instantiates is called SIQinU [13].

In summary, the organizational goals ontology is paramount for building the basis of strategies patterns. By having the main terms involved in the evaluation domain such as quality view, quality focus, entity super-category, business and information need goals, among others, formalized into an ontology specified in OWL, enable the development of the strategy pattern recommender system. Consequently, we envision that the strategy pattern recommender system will be useful when an organization establishes ME/MEC projects that operationalizes evaluation business goals.

5 Related Work

In [2] authors present a measurement goals ontology arguing that the measurement should be aligned with organizational goals to produce useful data for decision making. This ontology contains terms, relations and restrictions related to the alignment between measurement goals and business goals. It states the use of indicators to judge the level of achievement or satisfaction of the goals, but it is not explicit the description of the use of strategies to reach goals. This work was a valuable reference when doing the conceptualization step of our organizational goals ontology briefly described in Section 2 and documented in [19]. However, in [19], its formal specification, i.e., its implementation was not available. Unlike [2], our ontology describes the use of strategies to fulfill business/information need goals as well as considers that information need goals are linked to business goals at different organizational levels. This alignment allows to know in which extent business goals are being achieved. Additionally, our ontology is currently implemented in OWL (as presented in this work), while in [2] does not.

There are also proposals that underline the importance of having ME goals aligned with those established at strategic levels. One worth mentioning is *GQM$^+$Strategies* [3]. GQM$^+$Strategies includes a goal-oriented framework for the design and implementation of software measurement projects at different organizational levels. It promotes the use of strategies to derive lower level goals that are able to satisfy the main strategic goal. This approach has the terminological base structured as a glossary, where the main terms referred to organizational goals are defined. However, it lacks the semantic richness that an ontology provides in addition to well-established process specifications for different strategies regarding diverse evaluation goal purposes.

Another related work is the *Goal-Driven Measurement* approach [17], which describes a process for the definition of measurement goals aimed at helping to understand aspects of the organizational goals. This approach offers a detailed guide that serves engineers and practitioners to implement a goal decomposition process from organizational goals to measurement ones. But authors do not define explicitly concepts such as business goals, organizational level, information need, strategy and strategy patterns, among others, nor specify a family of strategies to be used.

In summary, we have not found proposals for the evaluation domain similar to the one we currently propose. That is, our approach considers the definition of multilevel business and information need goals; the definition of different evaluation purposes and the consideration of quality views; the formulation of ME/MEC projects to operationalize business/information need goals for diverse evaluation purposes; and the adoption of strategy patterns for instantiating specific strategies that help to achieve those evaluation goal purposes. The vocabulary of the approach has been structured in ontologies.

Furthermore, as discussed in the present work, we have followed an ontology development process which transforms the initial product (the ontology specified in UML) into a final product (the evaluated ontology, which has been coded in a semantic formal language like OWL).

6 Conclusion and Future Work

In this article, we have illustrated the transformation made from the conceptual model of the organizational goals ontology into a formal specification. To this end, we have followed the METHONTOLOGY approach in the construction of the ontology. Particularly, for doing the transformation, we used the UMLtoOWL tool, which transforms UML class diagrams into OWL specifications. As a result, we have initially evaluated the organizational goals ontology by means of a couple of competency questions.

We propose the use of evaluation strategies instantiated from strategy pattern to fulfill a business/information need goals, which are operationalized by a ME/MEC project. This fact constitutes a contribution to the evaluation domain when carrying out quality assurance activities, as analyzed in the previous Section.

Considering the semantic processability, we are currently working on the development of the strategy pattern recommender system as a practical use of the organizational goals ontology.

Ultimately, the recommender system can be useful when an organization establishes ME/MEC projects that operationalizes evaluation goal purposes. Hence, considering the purpose of the evaluation goal and the amount of quality views, the recommender system will suggest the suitable strategy pattern that fits better for the given goal.

Acknowledgments. This work and line of research are supported by Science and Technology Agency of Argentina, in the PICT 2014-1224 project at Universidad Nacional de La Pampa. Also, we appreciate a lot the support given by Andreas Grünwald in some incidents raised when using his UMLtoOWL tool.

References

1. Barcellos M. P., Falbo R., Rocha R.: A Strategy for Preparing Software Organizations for Statistical Process Control. Brazilian Computer Society Journal, 19(4), pp. 445-473, (2013)

2. Barcellos M. P., Falbo R., Rocha A. R.: A Well-Founded Software Process Behavior Ontology to Support Business Goals Monitoring in High Maturity Software Organizations. 14th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 253-262, (2010)
3. Basili V., Lindvall M., Regardie M., Seaman C., Heidrich J., Jurgen M., Rombach D., Trendowicz A.: Linking Software Development and Business Strategy through Measurement. IEEE Computer, 43:(4), pp. 57-65, (2010)
4. Becker P., Papa F., Olsina L.: Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy, CLEI Electronic Journal, 18:(1), pp. 1-26, (2015)
5. Calero, C., Ruiz, F., and Piattini, M.: Ontologies for Software Engineering and Software Technology. Springer Science & Business Media, (2006).
6. CMMI: Capability Maturity Model Integration, CMMI for Dev. Version.1.3. CMU/SEI-2010-TR-033, USA, (2010)
7. Fernández-López, M., Gómez-Pérez, A., and Juristo, N.: Methontology: From Ontological Art Towards Ontological Engineering. In: Proc. of the Ontological Engineering American Asociation for Artificial Intelligence, pp. 33-40, (1997)
8. Grünwald, A.: Evaluation of UML to OWL Approaches and Implementation of a Transformation Tool for Visual Paradigm and MS Visio. Vienna, AT: Vienna University of Technology, (2011)
9. Guizzardi, G., Falbo, R., Guizzardi, R.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. CIbSE, Pernambuco, Brazil, pp. 127-140, (2008)
10. ISO/IEC 15939: Software Engineering - Software Measurement Process, (2002)
11. ISO/IEC 25010: Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models, (2011)
12. ISO/IEC 9126-1: Software Engineering Product Quality - Part 1: Quality Model, (2001)
13. Lew P., Olsina L., Becker P., Zhang L.: An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications. Requirements Engineering Journal, Springer London, 17:(4), pp. 299-330, (2012)
14. Olsina L., Papa F., Molina H.: How to Measure and Evaluate Web Applications in a Consistent Way. Web Engineering: Modelling and Implementing Web Applications, Rossi G., Pastor O., Springer HCIS, Cap. 13, pp. 385-420, (2008)
15. Olsina, L. and Martin, M.: Ontology for Software Metrics and Indicators. Journal of Web Engineering, 2(4), pp. 262-281, (2004)
16. OMG: Business Motivation Model (BMM), Ver. 1.3, (2015)
17. Park R. E., Goethert W. B., Florac W.: A. Goal-Driven Software Measurement. A Guidebook. Carnegie-Mellon, Software Engineering Inst., TR. CMU/SEI-96-HB-002, (1996)
18. PMBOK: A Guide to the Project Management Body of Knowledge, 5th Ed., (2013).
19. Rivera B., Becker P., Papa F., Olsina L.: A Holistic Quality Evaluation, Selection, and Improvement Approach driven by Multilevel Goals and Strategies. CLEI Electronic Journal, 19(3): 3, (2016)
20. Rivera M.B., Becker, P., Olsina L.: Quality Views and Strategy Patterns for Evaluating and Improving Quality: Usability and User Experience Case Studies, Journal of Web Engineering, Rinton Press, USA, 15:(5&6), pp. 433-464, (2016)
21. Singh S., Woo, C.: A Methodology for Discovering Goals at Different Organizational Levels, In Proc, 3th International Workshop on Business/IT Alignment and Interoperability (CAiSE'08 Conference), Montpellier, France, pp. 16-30, (2008)
22. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. Pellet: A Practical OWL-DL Reasoner. Web Semantics: Science, Services and Agents on the World Wide Web, 5(2), pp.51–53, (2007)