AN ASSISTED PROCESS FOR BUILDING SEMIOTIC WEB ONTOLOGY

Júlio Cesar dos Reis
Department of Information Systems, Institute of Computing, University of Campinas and
Center for Information Technology Renato Archer, Campinas, São Paulo, Brazil
julio.reis@cti.gov.br

Rodrigo Bonacin
Center for Information Technology Renato Archer and Faculty of Campo Limpo Paulista (FACCAMP)
Rua Guatemala, n° 167, 13231-230, Campo Limpo Paulista, São Paulo, Brazil
rodrigo.bonacin@cti.gov.br

Maria Cecilia Calani Baranauskas
Department of Information Systems, Institute of Computing, University of Campinas (UNICAMP)
Av. Albert Einstein, 1251,13083-970, Campinas-SP, Brazil
cecilia@ic.unicamp.br

Keywords: Organizational Semiotics, Semantic Analysis Method, Web Ontology Language, OWL, Semantic Web.

Abstract: The evolution of the Semantic Web (SW) depends on novel methodologies to represent the meanings of the huge amount of information available nowadays. Recent proposals in literature have explored new approaches based on Semiotics. The ‘Semiotic Web Ontology’ (SWO) is an attempt to model the information in a more precise way, and, at the same time to be compatible with the SW standards. This paper presents a computer assisted process for building SWO. The process includes transformation rules for deriving an initial Web Ontology (WO) described in Web Ontology Language (OWL) from Ontology Charts (OCs) produced by the Semantic Analysis Method (SAM). In this paper the process and a tool are illustrated and discussed. Moreover results of the application of the process in a real case study are presented showing the potential of the approach.

1 INTRODUCTION

One of the main challenges in the contemporary Web evolution is related to the treatment of the information meaning. There is a growing need for methods, techniques and tools to better represent the semantic aspects of the information available in Web systems. The first initiatives taken by Berners-Lee et al. (2001) already aimed at creating a Web that also takes into account the meanings of information and not just its structure and protocols. Web Ontology (WO) is a core concept in this initiative.

Nevertheless, recent studies such as Gärdenfors (2004), Tanasescu & Streibel (2007), among others, point out that there are still various limitations and problems regarding technologies coming from the Semantic Web (SW) initiative. Even with the advent of the WOs, there are still no tools to assist in the organization of the information in a way suitable for human mental operations in an individual or societal way. In order to overcome such limitations and problems, some approaches have been described in literature, as the one proposed by Gärdenfors (2004).

Modelling approaches for WOs which uses the Semantic Analysis Method (SAM) (Liu, 2000) as a starting point may improve the representation of the information. This proposal uses the concept of “Semiotic Web Ontology” (SWO) (Reis et al., 2010) which combines SAM concepts with technologies of the SW to describe ontologies using OWL (W3C, 2004). This approach enables to incorporate into SW ontologies, concerns and possible representations arising from the Ontology in a Semiotic perspective, building more representative WOs (Reis et al., 2010).

In order to create a SWO, heuristics and transformation rules are necessary to construct an OWL ontology from an OC. A set of heuristics was proposed in Reis et al. (2011) that makes explicit a relation between these models. This paper presents a new procedure that uses the transformation rules in a tool to build an initial model of a SWO. The paper also presents a case study and the discussion about the outcomes.

As semantic refers to meaning, and meanings are socially created by humans, we expect to create a more faithful computer ontology considering an information system as a more abstract conceptual model that can capture the behaviour of the involved agents.

Assuming that the Semiotic approach contributes with improvements in business modeling, it is plausible to have both: the Organizational Semiotic (OS) (Stamper et al., 1988) methods with a valuable view of the social context, and a WO described in OWL that interoperates with what already exists. Among the main benefits of the proposal are the computer-supported transformation process and the production of an initial design model more closely related to the real world concepts.

The objective of this paper is to present the process supported by a software tool to construct SWO, and the analysis of a real case study. The transformation is made by an assisted process with human intervention in the SONAR tool (Santos et al., 2008). The paper is organized as follows: Section 2 presents the implemented transformation rules including examples that illustrate the process; Section 3 presents the Case Study; Section 4 discusses the results and Section 5 concludes.

2 THE PROCESS WITH THE TRANSFORMATION RULES

In order to computationally accomplish the SWO, it is necessary firstly to model the OC. For that, the SONAR tool is used. The SWO is obtained through new functionalities of SONAR extended with the transformation rules implemented based on the heuristics.

In the proposed assisted process, after modeling the OC, the analyst may transform it into OWL ontology by using wizards. In order to accomplish the entire process it is necessary to consider two important steps using the wizards: first, the analyst needs to specify which affordances are mapped to OWL classes. Figure 1 shows an example of the SONAR’s wizard where the analyst specifies which affordances are mapped into OWL classes. The OC used in the example is presented in Liu (2000:79) for project management.

![Figure 1: SONAR’s interface for choosing the affordances that will be mapped to OWL classes](image1.png)

In the next step, the analyst must specify which affordances are mapped to object properties in OWL, and to associate these with appropriate OWL classes (affordances that were already mapped to classes) that are connected (contain) to each object property selected (see Figure 2).

![Figure 2: SONAR’s interface for choosing the affordances that will be mapped to object properties into OWL classes](image2.png)

This is important due to the necessity of knowing which class will have priority on an object property (i.e. which affordance will have priority on other). For example, thinking in OWL the analyst must ask: Who employs? The “employee” or the “employer”? Thus, using the wizard as shown in Figure 2 the analyst may set that the “employer” employs (i.e. the class “employer” in OWL will be in the domain of the object property “employ”). It is also necessary to observe that in Figure 2 the class “employee” will be connected to the object properties: “assigned_for”, “works_on”, “responsible_for”.
“works_on” and “works”. Using this step, the necessary information will be available to the procedure to determine the appropriate transformation rules that will be applied, and consequently derive an initial version of the SWO.

Next we present how each OC concept is mapped to OWL code, considering the information provided in the steps of the wizard; parts of OWL code are illustrated and exemplified regarding the application of the implemented transformation rules in the SONAR tool.

Affordances – According to the “affordance heuristic” for mapping affordances, they can be an OWL class or an object property. Using the wizards illustrated by Figure 1 and Figure 2, the tool has the information that describes which affordances are mapped to classes; then the rule transformation creates an OWL class for each affordance set as class, as for example to the affordance “task”; see the following OWL code generated for this affordance.

```xml
<owl:Class rdf:ID="task"/>
```

However, since an affordance is set as an object property, the rule creates an object property for each affordance. The domain of each object property created is the class defined at the wizard (Figure 2) by the analyst. Thus, if the “employee” class sets the object property “works”, then the “employee” is the domain of the object property. See the following code.

```xml
<owl:ObjectProperty rdf:ID="works">
  <rdfs:domain rdf:resource="#employee"/>
  <rdfs:range rdf:resource="#department"/>
</owl:ObjectProperty>
```

Considering this fragment of code, the range is the “department” resource. In order to set the appropriate range, the “affordance transformation rule” gets the first affordance that is not an object property from the first ontological dependence connected to the considered object property. If the source affordance was set as an object property, it cannot be set as the range, and then the rule considers the source affordance from the next ontological dependence. When the source element is a set of affordances like in specialization, the range will be set as the most generic affordance. And, when there is a role-name in the ontological dependence, the rule considers the role-name; for example in the next OWL code; in this case, the domain of the object property “employs” is the “employer”, and since the other ontological dependence is different from the one between “employer” and “employs”, it is the one that contains the role-name “employee” that will be set as the range of the object property “employs”.

```xml
<owl:ObjectProperty rdf:ID="employs">
  <rdfs:domain rdf:resource="#employer"/>
  <rdfs:range rdf:resource="#employee"/>
</owl:ObjectProperty>
```

Agents – In this transformation a class named “Agent” is created in OWL, and all agents from the OC form a sub-class of that. See the following OWL code example showing the classes “Agent”, and “person” as a sub-class of “Agent”.

```xml
<owl:Class rdf:ID="Agent"/>
<owl:Class rdf:ID="person"/>
<owl:Class rdf:ID="Agent"/>
</owl:Class>
```

Determiners – For each determiner of the OC an owl:DatatypeProperty is created, which has the name of the determiner. The domain of each data property created is the OWL class of the affordance to which the determiner is connected, and the range is a string resource. The OWL code example shows the property “proj_budget” and its “project” domain.

```xml
<owl:DatatypeProperty rdf:ID="proj_budget">
  <rdfs:domain rdf:resource="#project"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Role-Name – In the “role-name heuristic” classes in OWL are created for each role-name as sub-classes of the agent of the left side of the role-name. According to the following OWL code example illustrated, the “employee” role-name is an OWL class and it is a sub-class of the “person” class.

```xml
<owl:Class rdf:ID="employee">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="person"/>
  </rdfs:subClassOf>
</owl:Class>
```

Regarding the relation of the role-name with the right side affordance, whether the right side affordance of the role-name is an object property, considering the ontological dependence to which the role-name belongs, then the role-name once mapped to an OWL class, will be in the domain of this object property. In this situation, the rule transformation verifies if the class role-name is already “owner” of the object property (i.e. whether the class role-name is already in the domain of this object property), thus this relation is already done in OWL; if not, it is
necessary to set the class role-name as domain of the object property. In the following OWL code example, the “employee” affordance was already in the domain of the object property “assigned_for”.

```owl
<owl:ObjectProperty rdf:ID="assigned_for"/>
<rdfs:range rdf:resource="#employee"/>
</owl:ObjectProperty>
```

On the contrary, if the right side affordance (target of the ontological dependence) is not an object property, the transformation rule creates a relation between these two classes, i.e. between the role-name and the class of the right side; then a “depends_on” object property is created, and the domain is set as the OWL class that represents the right side affordance of the ontological dependence, and the range as the role-name.

**Whole-Part** – There are two situations of whole-part relation. First, when both the source affordance and the target affordance are nouns. In this situation the affordances are mapped as classes using the wizard of Figure 2; to represent this, an object property named “partOf” is created, and the target class is set as a restriction of the source class. For example, the agent “department” mapped to an OWL class is part of the “organization” agent that was also mapped to a class in the OWL ontology.

In order to accomplish the “whole-part heuristic”, specific object properties were created. One is the object property “hasPart”, which is inverse of the transitive property “partOf”. Since the affordances (source and target) were known and set as classes using the wizard (Figure 2), the target class will be a restriction of the source class. The OWL “department” class is part of the “organization” class. In order to represent it using the created object properties (“partOf”) and “hasPart”), the “organization” class is set as restricted by a sub-class on the property “hasPart”, in which all values from the “organization” come from the “department” class. The next OWL code describes the representation of the “organization” class.

```owl
<owl:Restriction>
  <owl:allValuesFrom>
    <owl:Class rdf:ID="department"/>
  </owl:allValuesFrom>
  <owl:Class rdf:ID="organization"/>
</owl:Restriction>
```

The “task” class is also part of the “project” class, thus the same idea is applied as in the “organization” to represent the “project” class in OWL. In both situations (“department”-“organization” and “task” - “project”) an object property of ontological dependence between the whole and part affordances is also created. The “ontological dependence rule transformation” will be better explained later in this section. As presented by the “whole-part heuristic”, the part is ontologically dependent on the whole, thus “department” depends on “organization” and “task” depends on “project”. In order to handle it, this transformation rule also creates object properties that represent ontological dependences as described by the OWL example code.

```owl
<owl:ObjectProperty rdf:ID="assigned_for"/>
<rdfs:range rdf:resource="#project"/>
</owl:ObjectProperty>
```

In the other situation, when both affordances (whole and part) are set as object properties in OWL using the wizard (Figure 2), the target object property that represents the part is mapped to a sub-property of the source object property that represents the whole. This situation does not appear in the OC that has been used as example, since the whole-part affordances of the relations in the OC were set as OWL classes. However, this rule transformation foresees this situation and is able to handle it. Regarding the ontological dependence representation of such situation, there are limitations as described in the discussion section.

**Specialization**– Based on the “specialization heuristic”, for example, since the “task” affordance was set as an OWL class in the wizard, the more specific types of “task” must also be set as classes and sub-classes of “task”. The OWL code below shows this situation with the “simple” and “complex” affordances.

```owl
<owl:Class rdf:ID="complex"/>
<rdfs:subClassOf>
  <owl:Class rdf:ID="task"/>
</rdfs:subClassOf>
</owl:Class>
```

In the conducted example, there is not any situation in which the more generic affordance was set as an object property, but it could happen, and
the rule transformation foresees that such situation may happen. Imagine that the affordance “works” that was set as object property have more specific kinds of work such as: “write” or “paint” for example. Then the affordances “write” and “paint” would be sub-properties of the object property “works”. The OWL code would be as:

```xml
<owl:ObjectProperty rdf:ID="write">
    <rdfs:subPropertyOf rdf:resource="#organization"/>
    <rdfs:domain rdf:resource="#project"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="paint">
    <rdfs:subPropertyOf rdf:resource="#organization"/>
    <rdfs:domain rdf:resource="#project"/>
</owl:ObjectProperty>

<owl:TransitiveProperty rdf:ID="project_dependsOn_organization">
    <rdfs:range rdf:resource="# organization"/>
    <rdfs:domain rdf:resource="# project"/>
    <rdfs:subPropertyOf rdf:resource="# organization"/>
    <rdfs:subPropertyOf rdf:resource="# organization"/>
<owl:TransitiveProperty>
```

Ontological Dependence – In this rule transformation, transitive properties in OWL are created (i.e. object property) for all ontological dependences of the OC. Since both source and target affordances are classes in OWL, the property is created setting the domain and range; the domain is set with the target affordance, while the range is set with the source affordance. Consider the next OWL code examples took from the transformation applied to the OC of the example regarding the “project” affordance depending on the “organization” affordance.

```xml
<owl:ObjectProperty rdf:ID="write">
    <rdfs:subPropertyOf rdf:resource="#organization"/>
    <rdfs:domain rdf:resource="#project"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="paint">
    <rdfs:subPropertyOf rdf:resource="#organization"/>
    <rdfs:domain rdf:resource="#project"/>
</owl:ObjectProperty>

<owl:TransitiveProperty rdf:ID="project_dependsOn_organization">
    <rdfs:range rdf:resource="# organization"/>
    <rdfs:domain rdf:resource="# project"/>
    <rdfs:subPropertyOf rdf:resource="# organization"/>
    <rdfs:subPropertyOf rdf:resource="# organization"/>
<owl:TransitiveProperty>
```

If one of the (source or target) affordances were not set as an OWL class, it cannot be set as domain or range of the object property. In this situation the rule transformation seeks for the class that is in the domain of this object property. Then, the object property will be replaced by this class. Such situations may lead to wrong existential relationships as will be argued in the discussion section. Moreover, when the source affordance of the ontological dependence is a specialization, the relation is made using the more generic affordance due to not being possible to connect an ontological dependence to a specific specialized affordance in SONAR.

3 THE VILANAREDE CASE STUDY

The proposed method to model SWOs was evaluated in the context of the Vilanarede system. It is an online social network constructed to investigate inclusive social network principles. Vilanarede users collaborate in the system by announcing and sharing goods and services, events and ideas. The content domain of the announcements available in Vilanarede is wide-ranging; i.e. there are announcements about various contexts. Actually, one of the main challenges in modeling this context is to deal with the open and informal domain of the announcements. Such announcements are diversified regarding the contents: sale of various handmade products, meals, electronic products, advocacy services, events including debate about education, as well as ideas that cover various subject such as: recipes, environmental awareness, health tips, and so on.

The conducted case study considered 230 announcements created by users of Vilanarede distributed among products, services and ideas. The SAM was applied using these real announcements in an attempt to model the semantics of the social network. Commentaries written by other users about the announcements were also considered in the modeling. Altogether 10 groups of announcements were selected according to subjects’ sets. Based on these, the SAM was applied to each group resulting in 1 or 2 OC(s) for each group. The subjects of the groups were: cooking and meal ordering, sale of products and services, cultural events, announcing Vilanarede, physical exercises and health promotion, social projects including inclusion and citizenship, offer of courses and seminars, health-oriented food, handicraft and environment.

Using the proposed method to create SWO, an activity was carried out by sixteen graduate students in Computer Science, in the role of analysts, who were studying OS by the time. They were divided into groups of two or three people. Each group received one subject (one of those already mentioned) collected from the Vilanarede’s announcements. Each set of data includes among twenty to thirty announcements.

After learning how to apply SAM to construct OCs, each group would model an OC that might represent the semantics of those Vilanarede announcements. The students would use the SONAR tool to model the OCs. They were free to define their own strategies to build up the OCs. The groups had about one week to explore the announcements and to construct their OCs. At the end of the activity, each group presented the results that they achieved during the modeling process; six initial versions of OCs were created by the students. The next step involved modeling an SWO from them using the SONAR
extended with the implemented rules. Then, an OWL ontology file from the modelled OC was created.

The fact of thinking in agents, affordances and ontological dependences resulted in a differentiated representation of the knowledge of the context under study. The SAM and its successful application are crucial to the proposed method, since it provides important outcome to the next steps. Regarding the informal contexts we have worked on, the experience in modeling the OCs have faced two main issues: the relation between the Universal versus the particular, and the diagram granularity.

Due to the variety in the content, too much things were identified as particular things in the majority of the announcements. However, the OC should model the Universal concepts. In order to achieve it, abstractions were created and the particular stayed as possible values of a determiner, for example. Moreover, some affordances modeled are applied to specific interpretation that comes from the ISN content (e.g. a refrigerator was set as a kitchen utensil). Such affordance at the same time could be interpreted as an appliance or as furniture, or a material or a commodity, etc. Thus, the meaning of the context is made by those who use it (i.e. the agent).

Diagram granularity was another recurrent problem in our experience. Many times during the modeling we had to decide whether the words should be considered as determiner values or as affordances. Sometimes the better option might be to model a word as an affordance in order to be able to add new modeling details, but it is necessary to establish a limit of detail due to time restrictions. The more the ontology is detailed, the richer the context under study can be described. However, to model a deep and wide context is a hard time-consuming task. Moreover, since we have made the modeling from unstructured texts described in natural (and informal) language, too many details are lost during the modeling; this can also impact in the ontologies application.

During the proposed activity for modeling OC from the VilanaRede content, various OCs were modeled, and some intersection between them could be identified. Based on the results of the activity, it is possible to observe that the variety and informality of the domain is really an aggravating modeling problem. Besides, the results point out that SONAR was appropriate to the process of modeling SWO; nevertheless, some improvements could tune and facilitate the modeling process.

4 DISCUSSION

The first topic to be discussed refers to the different visions of the approaches. SAM follows the subjectivist paradigm while the traditional WO approach follows the objectivist paradigm. Whereas objectivism assumes a single reality and explains differences of ideas as aberrations, subjectivism treats different ideas of individuals as a starting point for a shared reality; subjectivism emphasizes the abilities of individuals, their freedom to choose courses of action and the moral responsibility for their choices, as well as the uncertainty, novelty and strife they bring about (Liu, 2000:24).

Concerning OC and WOs, the differences between them become evident already in the root of both models; in OC the root of everything is a certain Society, while in OWL the root of everything is the class Thing. By using SAM the model exposes a few basic assumptions underpinned by the ontological principles. There is a root agent in the chart which functions as the ultimate antecedent for the whole problem domain under study. This root agent is the social community in which all its members share the same fundamental concepts and culture. In SAM, Ontology is not an absolute knowledge rather it is relative to a specific Society. Thus, regarding the modeling in OWL there is not Society without Thing (the Society is a Thing), and in the SAM perspective there is not Thing without Society (every affordance depends on the Society existence). Moreover, in SAM the focus is on the agents including their patterns of behaviour (affordances) and their ontological dependencies, while in OWL the focus is on the concepts (classes) and their relationships.

Identifying the agents of the context, besides understanding and modeling the invariants of behaviour of these human agents are key points for more accurate and flexible computer-interpretable ontology models. It was not intended here to create an OC in OWL or to substitute the OC at the conceptual or business levels; the intention was to investigate possible contributions of OC to OWL modelling. Furthermore, it is not expected that the OC and OWL ontologies model the same thing, since they represent different concepts, visions and paradigms. There are different concepts in OWL and in SAM, consequently the semantic models exhibit different representations/interpretations of reality.

In this context, it is important to verify whether the heuristics really enable to bring properties and contributions from one model to the other correctly. It is necessary to check whether the OWL ontology
created has inconsistencies from two points of view. First, by conceptually analyzing whether the existential and temporal dependences between the affordances in the OC are kept. This can be reached by observing whether the agent-affordance relationship still exists in the OWL ontology. The proposal clearly maps each entity from one model to the other. The agents and role-names create a well defined hierarchy of classes. Affordances set as classes are mapped to OWL classes, while the affordances set as object properties (i.e. represent the patterns of behaviour as properties) tie the relationship between the agents and role-names with their affordances (action sense) to concepts (mapped as classes). The specifics of affordances are successfully mapped to sub-classes ensuring the relationship with the more generic affordance. Determiners have an immediate mapping to data properties. Whole-part relationships are set as restriction of classes preserving the relation between whole and part mapped in the OC. Ontological dependences are also represented as object properties connecting the affordances that depend on each other to exist.

Moreover, it is important to examine whether the OWL ontology is consistent from the SW point of view. For that, it is necessary to observe whether all the relationship of the model logically make sense. Once the rules implementations have followed the OWL DL restrictions in order to assure computational completeness, it may lead to consistencies. However, it still may not avoid modeling problems. In order to verify it, studies using Venn diagrams could be conducted analyzing the created ontology. For that, instances should be created. Besides, it is also possible to use a semantic reasoner available in ontology editor software tools to test the generated ontology and check its consistency. From the presented results it is possible to see the potential of the proposal to different SW initiatives. For instance, the SW ontology created by SONAR supports tasks such as semantic search and also could be integrated to other ontologies in the Web environment since it is described in OWL.

In addition to the potential of the solution, it is possible to point out some limitations as well. First, the concept of ontological dependence is a hard problem to deal with in OWL; the solution treats part of this representation since the ontological dependence is a temporal concept, thus representing it as object properties is not enough. Still concerning ontological dependence, not all ontological relations modelled in the OC are possible to be fully transcribed to OWL following the heuristics. The situations that involve affordances mapped to object properties are more complicated to deal with. For instance, following the heuristics and using OWL DL, it is not possible to represent the ontological dependences between two affordances that were set as object properties, since both will not be mapped to OWL classes; a third object property should be necessary to set the domain and range with such objects that are not classes. In OWL DL, in order to retain computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time) the domain and range can be just set as a class or a set of classes.

There are other limitations, for instance: once the right affordance of an ontological dependence may be set to object property and the left affordance may be an agent, an specialization (i.e. a specific affordance) or a noun affordance, thus using an approach in which the owner of the right affordance set as object property is considered as a possible domain can lead to wrong ontological relationships.

The “ontological dependence transformation rule” is not capable of dealing with the transitivity of the ontological dependence. In the presented solution, just the more immediate ontological dependence is represented. In order to support and treat transitivity dependence, in the future it could be handled by Semantic Web Rule Language (SWRL) (W3Cb, 2004). Therefore, the original model could be extended to better represent the temporal relations; for instance, it should be possible to create rules that represent situations such as: an individual of the “organization” class can only be instantiated whether it is associated to an individual of the “society” class; if this individual (of the “Society” class) is deleted, the other must also be.

OC usually has determiners connected to an affordance, besides the possibility to be connected to agents. Since some affordances can be set to object property by the assisted process, there are problems in representing it in OWL, once it is not adequate in OWL DL to have a data property connected to an object property. Data properties may just be connected to OWL classes. Other limitation is that in OC, it is possible to have a determiner of a determiner. In order to represent that in OWL, it should be possible to set the domain of a data property as other data property, and this is not possible considering OWL DL definition. Both domain and range must be set as a class or a set of classes. Moreover, the specialization situation in which the more generic affordances could be nouns and the more specific affordances could be verbs.
was not considered by the “specialization transformation rule” in this work, since there is not a way for an object property (the more specific affordances - verbs) to be set as sub-classes of a class in OWL (the more generic affordance - noun).

In general, there is a growing need for solutions that deal with deeper semantic aspects in Web systems. The considered OS approach may deal with the shortcomings of conventional SW ontologies, and this work contributes to a computational implementation in this direction. Such solution represents an effort toward a more human and social representative WO.

5 CONCLUSION

The SW evolution demands new approaches to better handling complex meanings. Consequently such evolution depends on more adequate methods to represent the knowledge presented in Web applications content. In this direction, based on previous work, which proposes to represent the knowledge based on OS methods, this work presented a solution to a SWO. A computer assisted process for building SWO was proposed in an attempt to deal with the shortcomings of conventional SW ontologies.

Based on heuristics to support the creation of a WO, transformation rules to describe OWL code from the outcomes of the SAM were presented. These transformation rules assisted by wizards were implemented in the SONAR case tool software in order to model the OC and to derive an initial version of a WO. The solution was analysed in a real modeling context showing the potential of the proposal, and the usefulness of the implemented rules. The heuristics and transformation rules support the assisted process and materialize the Semiotic-based approach ideas for building WOs. The solution brings opportunities to improve the semantic models used in the existing SW applications and initiatives, and also enhance the chances to "interoperate" with what already exists, since OWL is a W3C standard.

Next steps involve refining the implemented transformation rules, generating representations using SWRL as described in the discussion section and also using meta-classes. This strategy makes possible to deal with the transformation of other aspects of the OC. Moreover, practical experiments that may illustrate the application of this approach, including the use of the SWO in a search mechanism are ongoing. Further work can also involve the inclusion of the OS Norm Analysis Method in the process.

REFERENCES