Innovating with Concept Mapping
Proc. of the Seventh Int. Conference on Concept Mapping
Tallinn, Estonia 2016

ONTOMAP: FROM CONCEPT MAPS TO SHALLOW OWL ONTOLOGIES

Guilherme N. Pinotte & Davidson Cury, Universidade Federal do Espírito Santo, Brazil
Amal Zouaq, Royal Military College of Canada, Canada
Email:guilhermepinotte@gmail.com

Abstract. Knowledge representation is one of the areas that Education is concerned with. Recently, ontologies have gained great importance as a way of representing knowledge. Thus, several researchers have investigated the creation of ontologies from concept maps. This paper explains the process of mapping a concept map into an OWL ontology and shows the conceptual architecture of a system designed to accomplish this task. Functions aspects are also discussed and necessary adaptations to conventional concept maps editors in order to make them able to support such mapping. In addition, a comparative analysis of existing web tools is provided, highlighting their strengths and weaknesses. Besides, we present a prototype as a proof of concept.
Keywords: ontology, concept map, OWL

1 Introduction

In the context of Education, there are several ways of knowledge representation, such as text, drawings, concept maps, among others. Recent studies have tried to bring the benefits of ontologies into the field of Education, but its excessive formalism has hindered researchers seek for more informal ways. Thus, many studies are being conducted on the use of concept maps for building ontologies. On one hand, concept maps are informal, simple to construct and easy to understand. On the other hand, ontologies are formal and difficult to create, hence requiring the presence of an expert to create them.

Concept maps are graphical tools for organizing, representation and knowledge construction. They are formed by concepts, generally represented by circles or boxes of some type, an arrow to link two or more concepts and a label for defining the nature of the relationship (Novak & Cañas, 2008). An Ontology is an explicit and formal specification of a shared conceptualization (Studer, Benjamins & Fensel, 1998). In (Guarino, 1998**)**, can be found more information about this definition.

In Information Technology, there are several modeling languages able to represent knowledge about a particular domain, each with its own objectives and levels of complexity in the creation of their models. Some of these languages can be more expressive, e.g. UML (OMG, 2003), OntoUML (Guizzardi, 2005) and OWL (Dean et al., 2003), while others are less expressive, e.g. ER (Chen, 1976) and concept maps (Novak & Gowin, 1984). It is worth to note, although more expressive languages can represent knowledge more clearly and with good fidelity to the domain, their use is not always better suited when compared to other less expressive languages, as show in (Siau, Erickson & Lee, 2005). Along the expressiveness comes complexity, which means that the more expressive, more complex is the development of models, in addition to increasing the computational complexity (Brachman & Levesque, 2004). This fact complicates the process of building models by domain experts, especially in Education, where experts possibly do not have expertise to use those complex modeling languages, such as OntoUML (Guizzardi, 2005) or OWL (Dean et al., 2003).

Figures 1, 2 and 3 represent the same domain described in different modeling languages, showing their different purposes and levels of complexity in the construction of models. By analyzing in particular Figures 1 and 2, which show typically ontology languages, it is evident the difficulty that a domain expert would have in knowledge representation. In contrast, Figure 3, showing a concept map, makes clear the simplicity and informality of the language.

Therefore, this work proposes, by means of concept maps, provide to non-experts in ontologies the ability to represent knowledge of any domain, simply and without the need for a computer specialist. In other words, this work aims to transform the informal knowledge described by concept maps in formal knowledge of shallow ontologies described in OWL language, which is the standard adopted by the Semantic Web.

Why Ontologies can be important in Education? What are the possibilities of their uses? As far as knowledge representation is concerned, concept maps and ontologies have the same purpose. Its main difference lies in the fact that one is formal and the other not. In computer terms, ontological models carry higher semantics compared to concept maps. Because of this, their representative capacity increases exponentially.

Several activities can be supported by ontologies in Education, such as: 1) the initial identification of the domain of knowledge, for example, for the automatic generation of concept maps, is very difficult. A software agent, driven by an ontology, can identify with a certain precision and quick way the source text; 2) ontologies can guide intelligent agents in building cognitive profiles of apprentices, providing them customized mediations; 3) questions in natural language on a given domain of knowledge can also be automatically answered by an agent oriented by domain ontologies; 4) texts can be grammatically corrected, among many other applications.

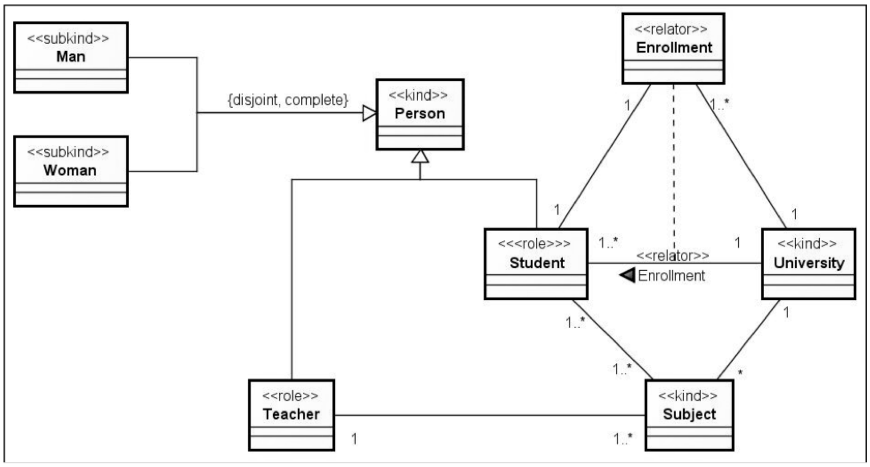

Figure 1. Example of a domain represented in OntoUML.

```
<owl:Class rdf:about="http://example.com/test#Man"/>
<owl:Class rdf:about="http://example.com/test#Teacher">
  <rdfs:subClassOf>
    <owl:Class rdf:about="http://example.com/test#Person"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://example.com/test#Subject"/>
<owl:Class rdf:about="http://example.com/test#Student">
  <rdfs:subClassOf rdf:resource="http://example.com/test#Person"/>
</owl:Class>
<owl:Class rdf:about="http://example.com/test#Woman"/>
<owl:Class rdf:about="http://example.com/test#University"/>
<owl:ObjectProperty rdf:about="can be">
  <rdfs:range rdf:resource="http://example.com/test#Woman"/>
  <rdfs:label xml:lang="en">can be</rdfs:label>
  <rdfs:range rdf:resource="http://example.com/test#Man"/>
  <rdfs:domain rdf:resource="http://example.com/test#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="has">
  <rdfs:label xml:lang="en">has</rdfs:label>
  <rdfs:range rdf:resource="http://example.com/test#Subject"/>
  <rdfs:domain rdf:resource="http://example.com/test#University"/>
</owl:ObjectProperty>
```

Figure 2. Source code snippet of a domain represented in OWL generated by OntoMap service.

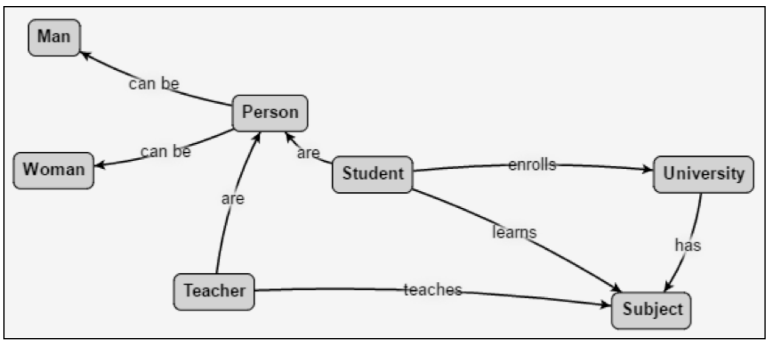

Figure 3. Example of a domain represented by a concept map.

We present a proposal of a conceptual architecture of a system we are calling OntoMap. The OntoMap is an idealized tool with the purpose of transforming concept maps in OWL ontologies. The tool is the result of the interaction of two basic parts: an adapted concept maps editor, and a service that contains the logic for mapping concept maps into OWL ontologies. The editor can be accessed through our Portal (Cury & Menezes, 2012). The service is hosted on a web-oriented platform services, called CMPaaS (Cury, Perin & Santos Jr, 2014). This platform is a major project which houses several services on concept maps and will be presented in Section 2 of this paper.

For the idealization of OntoMap tool, a critical analysis of some of the main tools available for download was held. The OntoMap differs from them because:

- It is part of a larger project that uses services for communication between client and server;
- Its implementation is based on SOA architecture, where its business logic was developed as a service;
- It incorporates the best features of the main available tools;
- It intends to enrich the shallow ontology created promoting integration with DBpedia.

This paper is structured as follows: Section 2 presents the context to which our tool belongs. Section 3 presents a literature review of the state of the art and technology. Section 4 discuss about the mapping process adopted. Section 5 describes the proposed architecture explaining each component of it. Finally, Section 6 presents a discussion and further work.

2 On the Context

We have been exploring, for about 15 years, concept maps applied to the learning process. During this time we have supervised numerous undergraduate and masters as well as PhD projects on the subject. Each of these jobs generated prototypical tools that have evolved over time, some of them already in production. However, as isolated tools they are inefficient. Therefore, we decided to integrate them. This section will be a brief discussion on concept maps in learning and why we need ontologies from a map, and also something about our web platform.

2.1 Maps in Learning

Concept maps are graphical representations of relationships between concepts. They have been used in many different fields of knowledge. In particular, they have attracted the interest of educators worldwide. Novak (Novak & Cañas, 2008), their creator, defines concept map as a tool for organizing and representing knowledge. The concept map, based on the meaningful learning theory of Ausubel, defined in (Ausubel et al., 1968), is a graphical representation of a set of concepts constructed in such a way that the relationships between them are evident. Concepts appear in boxes while the relations between concepts are specified by means of phrases that connect the concepts. Concepts are defined by nouns while linking phrases must have a verb or a verbal composition. Two or more concepts connected by linking phrases creating a semantic unit, we call a proposition. Each proposition defines a truth, a fact, detachable and understandable by itself. The propositions are a particular feature of the conceptual maps as compared to other similar structures such as mental maps or flowcharts. According to Novak, a concept map is a hierarchical tree structure, for the concepts, where the more general, or inclusive concept, appear at the top and the more specific ones in the lower parts of the tree.

However, there are different pedagogical approaches where the use of concept maps can help students in the processes of signification of new contents or even on the resignification of those already learned concepts. This happens mainly because the maps allow students to locate and establish relations of composition, similarity, differentiation, or equivalence between what they are learning and the concepts already present in their cognitive structure. Therefore, various researches are being conducted and new tools developed to enable the use of maps for different pedagogical practices.

According to Jean Piaget's Genetic Epistemology (Piaget, 1988), the mechanisms involved in the process of conceptualization imply an assimilation (the incorporation of an external element to a scheme of action or to a concept of the subject), starting from the active coordination of the actions of the subject and also the coordination of the observables of the objects of knowledge. Thus, the fact that an assimilation occurring in accordance with the possible accommodation (which is the need of the assimilation of taking into account the particularities of the elements to assimilating) requires the transformation of the systems of signification of the subject (which implies an update of the so-called "prior knowledge") so that these can integrate (and not just “to anchor”) new knowledge.

According to Dutra *et al* (2004) we can follow the representation of the system of meanings of a student on the dynamics of building a concept map. In this system we also recognize relating subsystems, supporting each other, for the construction of these meanings.

It is essential to highlight the central role of linking phrases in a concept map. When we compare concept map with Piaget´s knowledge structure, we can conceive the linking phrases as the structuring functions since they are responsible for the laws of composition of the system represented by the map. Jonassen (1996) stresses the effort to choose a phrase that represents a relationship between two concepts, both due to the large number of possibilities as well as the need of placing such a relationship in the context in which the pair of concepts is presented.

As learning processes are the result of student-student and student-teacher partnerships, we understand that concept maps can also serve as an important guide for students, alerting them about the constant possibility of enhanced versions with concepts and relationships qualitatively more significant.

2.2 Ontologies from Concept Maps: Why?

Ontologies and concept maps are very similar languages, especially under a point of view of their structures. Because of their graphical topologies, computers can easily process both. However, concept maps do not require the rigid formalism of the ontologies. Besides, their propositional structures are very similar to the structure used to represent the properties in description logic. Authors in (Zouaq, Nkambou & Frasson, 2007), (Starr, 2009), (Gomez-Gauchia & Diaz-Agudo, 2004), among others, suggested a procedure to support the transformation of concept maps in a knowledge base represented in description logic.

The ontologies have been represented primarily in description logic. They have played an important role in countless activities, especially in knowledge management with respect to the construction of intuitive human-machine interfaces, intelligent information retrieval and semantic web, among other activities. Ontologies have also been used to capture knowledge about some domain of knowledge.

We are interested in shallow ontologies especially in support of learning assessment. Using an architecture based on agents, ontologies can also guide the construction of virtual environments to support learning and cognitive modeling of students, considering their individual productions and those resulting from their collaborations. They are also good in intelligent tutoring systems, when a tutor agent can infer more appropriate mediation paths, on a straight dependence on the learner's profile.

An ontology may also play an important role on the automatic generation of concept maps from text. Herein, to find the domain of the knowledge is a very difficult task. An agent guided by a domain ontology can make it easier. Ontologies can also be useful in supporting the construction of grammatically correct text.

2.3 On our Platform

When dealing with the development of computational solutions, the subject in vogue in recent years is undoubtedly cloud computing. This is a computing model in which processing, storage and computing solutions (software) are offered by a service provider accessed remotely via the Internet. This technology allows applications to perform to retrieve information from anywhere, from any platform at any time, using only web instead of locally installed applications.

The main advantage offered by cloud computing is the ability of its services to be easily extended and integrated into the various applications, thus increasing productivity when creating new applications. Because of this, large companies in the information technology industry (e.g. Facebook, Apple, Google, Twitter, etc.) provide their API's (Application Programming Interfaces) to access their services using this computing model. Just to cite one example, a cloud application is widely used Google Maps.

Today, there are numerous applications that extend their functionality by offering complementary services such as geolocation applications that control the route, pace and/or calories consumed by an athlete in a physical activity. For this project, we seek to exploit this capacity expansion and productivity. We are creating the basic editing services, management and manipulation of concept maps that will be available to anyone in the world through our service platform.

One of the key features of the software architecture to be used in this project, known as SOA (Service Oriented Architecture), is its ability to promote integration. This means that new services that extend the functionality of the services offered by our platform can be developed and made available by anyone and anywhere in the world.

To date, this platform aims to offers the following services:

- Automatic generation of maps from unstructured text (Aguiar, Cury & Gava, 2015);
- Merge maps (Vassoler, Perin & Cury, 2014);

- Comparison of maps;
- Map editor;
- Correction maps;
- Information retrieval from questions in natural language (Perin, Cury & Menezes, 2014)**.**

Section 4 presents more details about the system architecture of our project as a whole, highlighting its two main layers.

3 From Concept Maps to Ontologies: A Literature Review

The first step in construction of this work was to read articles and testing the tools available for download on the web, in order to discover their basic features, their strengths and weaknesses. We have found three systems and, due to a lack of space, only a summary of their most important features is shown below in Table 1. Details are better shown in (Pinotte, Cury & Zouaq, 2015).

Tool	Advantages	Disadvantages
COE (Hayes et al., 2005)	• Mapping done through predefined stereotypes; • Stable environment and available for download; • Editor highlights the difference between classes, individuals and attributes; • Construction of simple map where the user is not forced to choose stereotypes of relations.	• Editor does not automatically change the layout of different elements, such as classes and individuals; • The user does not have a menu where you can select whether to show the different elements of OWL, such as classes, individuals and attributes, which makes polluted editor view (with lots of information); • The user does not know what each stereotype represents in the OWL language.
MAP2OWL (Garcia et al.,2008)	• It works with OWL natively, i.e., it does not perform algorithm transformation between conceptual and OWL maps; • The editor highlights the difference between classes, individuals and attributes; • The editor allows the user to select whether to display different elements of OWL, such as classes, individuals and attributes;	• Being plugin Protégé, the tool is limited to the use of Protégé; • Works only in version 3.3 of Protégé; • For a good representation of knowledge, the user needs to know the meaning of words such as functional, symmetric, transitive, among others;
(unnamed) (Simón, Ceccaroni & Rosete, 2007)	• Mapping is done through natural language processing; • The user creates the concept map without having any knowledge in OWL; • The concept maps editor do not need any kind of adaptation to perform the mapping.	• It works only in Spanish; • Algorithm depends on WordNet in Spanish; • By mapping occur through natural language processing, the rules of inference are limited.

Table 1: The systems main features

We conclude that the tools described in this section serve as a basis to construct the conceptual system architecture of OntoMap (Section 5), which will be explained in more detail in the following sections.
We point out that we had access only to the editors of the COE and MAP2OWL where testing the same was possible. For our prototype, we seek to maintain the main advantages we find these tools analyzed and eliminated what we consider as weakness or weaknesses. For our map editor, we joined characteristics of these two tools. The following are the features that we decided to keep with our editor:

- A list of predefined stereotypes. These stereotypes are available for the user when creating a new relationship: feature from COE;

- Selection Tabs to display the different elements such as the OWL classes, individuals and attributes, thus, for the user to decide which elements he/she wants to display on the screen: feature from MAP2OWL;
- Visually it differentiates the elements of OWL. These elements are: classes, individuals and attributes. For this, we take as a basis the project VOWL[1]: this feature was found in both COE and MAP2OWL;

With these characteristics, we believe that we were able to maintain the simplicity of the concept maps editor without visually polluting the user's view.

4 About the adopted Mapping Process

In the computing area it is common to adopt stereotypes to represent the necessary formalism of modeling languages or even programming languages. Thus, when a certain stereotype is used in a domain, it is understood that it will always have the same meaning, regardless of context. Based on that, this work adopts the use of stereotypes to build the necessary formalism to ontologies. This decision follows both the need for the creation of ontologies inherent formalism, as well as the critical analysis of the tools related to this work (Section 3). In addition, in order to maintain the simplicity of the concept maps, we decided to create stereotypes in natural language, which would allow any domain expert to use the proposed tool without the help of any Information Technology expert.

Being a work at an early stage, we decided to cover only the main constructors of the OWL language. Table 2 shows the stereotypes in natural language, their corresponding constructor in OWL and propositions to illustrate its use.

Stereotype	Constructor in OWL	Proposition
“are”	<rdf:subClassOf>	Professor “are” a Person
“equivalent to”	<owl:equivalentClass>	Person is “equivalent to” Human Been
“cannot be”	<owl:disjointWith>	Man “cannot be” a Woman
“exact the opposite of”	<owl:complementOf>	Human Being is “exact the opposite of” Extraterrestrial
“is composed of”	<there is no constructor>	Person “is composed of” Head and Body
“is a”	<there is no standard defined by W3C>	João “is a” Student
“same as”	<owl:sameAs>	João is “same as” Jhon
“different from”	<owl:differentFrom>	João is “different from” Maria
“is an attribute of”	<owl:DatatypeProperty>	Height “is an attribute of” Person

Table 2: Description of the stereotypes.

5 OntoMap: The Conceptual System Architecture and Technology

The proposed architecture is the result of the review we did of some tools as shown in the previous section. The architecture, however, differs from analyzed mainly because it is part of a broad platform of services based on SOA (Service Oriented Architecture), in which OntoMap will have its hosted service. Being an open-source platform, OntoMap can be used by the general community, and its services can be extended and improved by proper client applications. In addition, the editor of OntoMap can be accessed via our portal, which is a web system, requiring only an Internet connection.

OntoMap carries other innovations that allow building shallow ontologies that can be shared on the web. We idealize our tool as an information provider, i.e., an ontologies provider. A bank of ontologies will be available through connection to DBpedia[2], thus establishing a connection with the Linking Open Data[3]. Moreover, it is

[1] VOWL (Visual Notation for OWL Ontologies). Available in: < http://vowl.visualdataweb.org/>

[2] DBpedia. Available in: <http://wiki.dbpedia.org/>

[3] LOD (Linking Open Data). Available in: <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

possible to make sophisticated queries on Wikipedia and link different sets of Web data with the ones already present on website, so providing interoperability.

As we said, the OntoMap is the union of a web service, hosted on our platform, with an adapted editor of concept maps, accessed through the use of our Portal. Therefore, we can say that the Portal is the user interface layer, and the platform properly is the services layer. Figure 4a shows the architecture in a macro view of two layers, where the arrows indicate the Portal applications accessing services in the service layer. It is noticed that an application in the view layer can access more than one service on the platform, and a service on the platform can serve more than one application on the Portal. The services hosted on our platform will be available for use by the whole community, free. Thus, it is possible that another Portal (client) to consume them and use them any way the users want. Besides, whit this architecture, it is possible to extend our hosted services (Figure 4b).

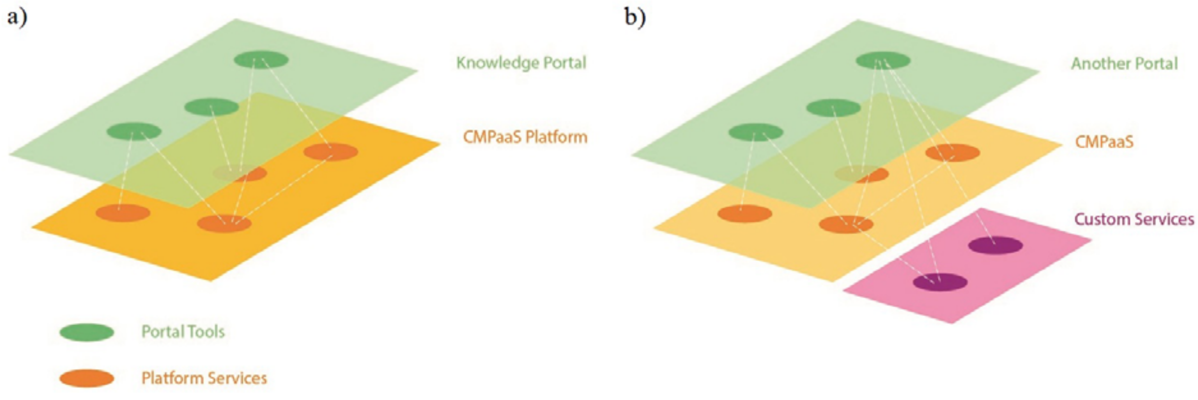

Figure 4. A macro view of the SOA architecture of the project.

The aim of the proposed architecture is therefore to generate shallow ontologies in OWL, straight from concept maps, making unnecessary the presence of experts in knowledge representation, throughout the process. In other words, we intend the resulting tool from that architecture, the OntoMap, to be used by experts of any domain in a simple and friendly way.

Figure 5 shows the conceptual system architecture of OntoMap itself, an updated version presented in (Pinotte, Cury & Zouaq, 2015), highlighting each module thereof. It shows the elements that are arranged in the layer of vision (Extended Editor and Creator module) and the elements of the service layer (Mapper, Coder and Integrator modules) which will be explained in the following steps.

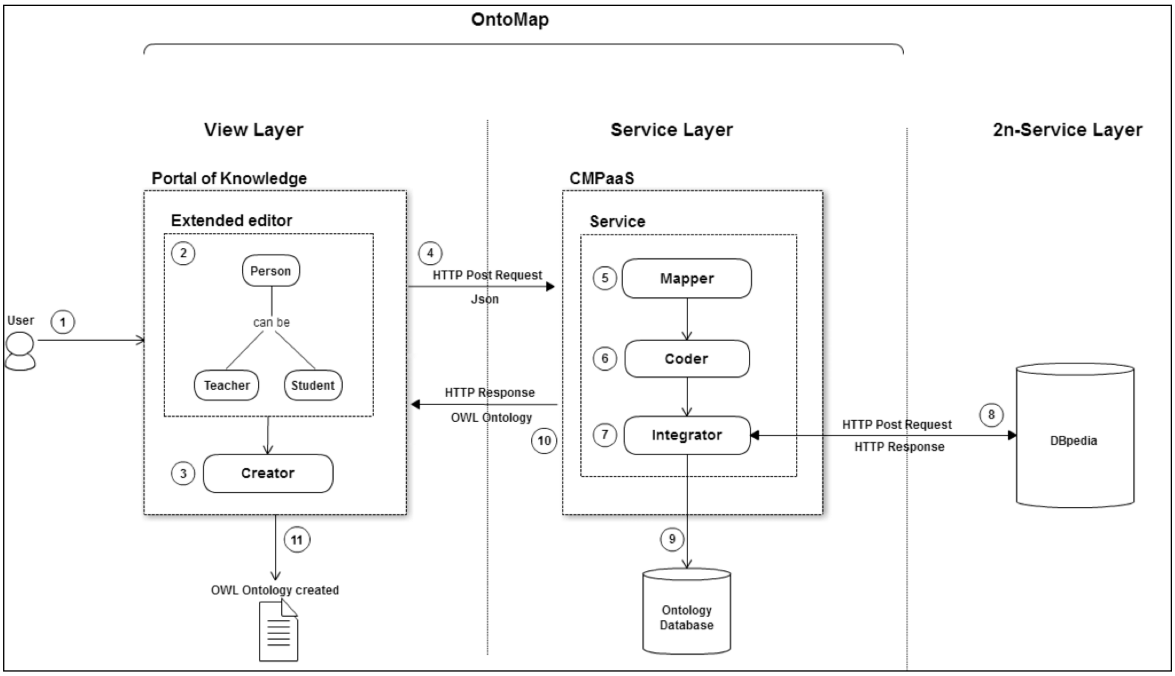

Figure 5. Detailed system architecture.

1. Through the access[4] on our Portal, the user can use the OntoMap tool;
2. When the authentication process is already done, the user can access the ontology editor (Figure 6). The ontology editor provides some additional features for the user compared to a common concept maps editor (further details will be explained later in Section 5). At any time when constructing a map, the user can click at the button labeled “Export Map in OWL Ontology”;
3. After step 2, the Creator module is responsible for the creation of a JSON file corresponding to the map created by the user;
4. After step 3, the Creator module sends the map to the server through a HTTP Post request via Ajax;
5. On the server side, on our service layer, the Mapper module is responsible for receiving the map in JSON format and to interpret it by creating a list of propositions. To perform the whole process, it is important to say the need of compatibility between the file generated by the Creator module and the input file received by the Mapper module;
6. Whit the list of propositions created on the step 5, the Coder module performs the conversion in fact by encoding the map into OWL ontology This module contains an interpreter with the list of pre-registered stereotypes (available on the creation of a new relation on Extended editor) and their constructors of OWL language. Thus, upon receiving the message of the Mapper module, the Coder is responsible for axiomatize the ontology, by converting the concepts, presented on the concept maps, in *Classes*, *Individuals* and *Datatype Properties* (elements of the OWL language) and the relations in *Object Properties*.
7. The Integrator module is responsible for sending a message to the DBpedia’s dataset adding some eventual information requested;
8. DBpedia receives the request and sends back a message with the information requested;
9. After step 8, the Integrator module is responsible for saving the OWL ontology created on our Ontology database, that will eventually be adapted to standard of Linked Data[5] being part of Linking Open Data;
10. After step 9, the Integrator module sends the OWL ontology created back to the client layer;
11. To finish the whole process, the client saves the OWL ontology locally.

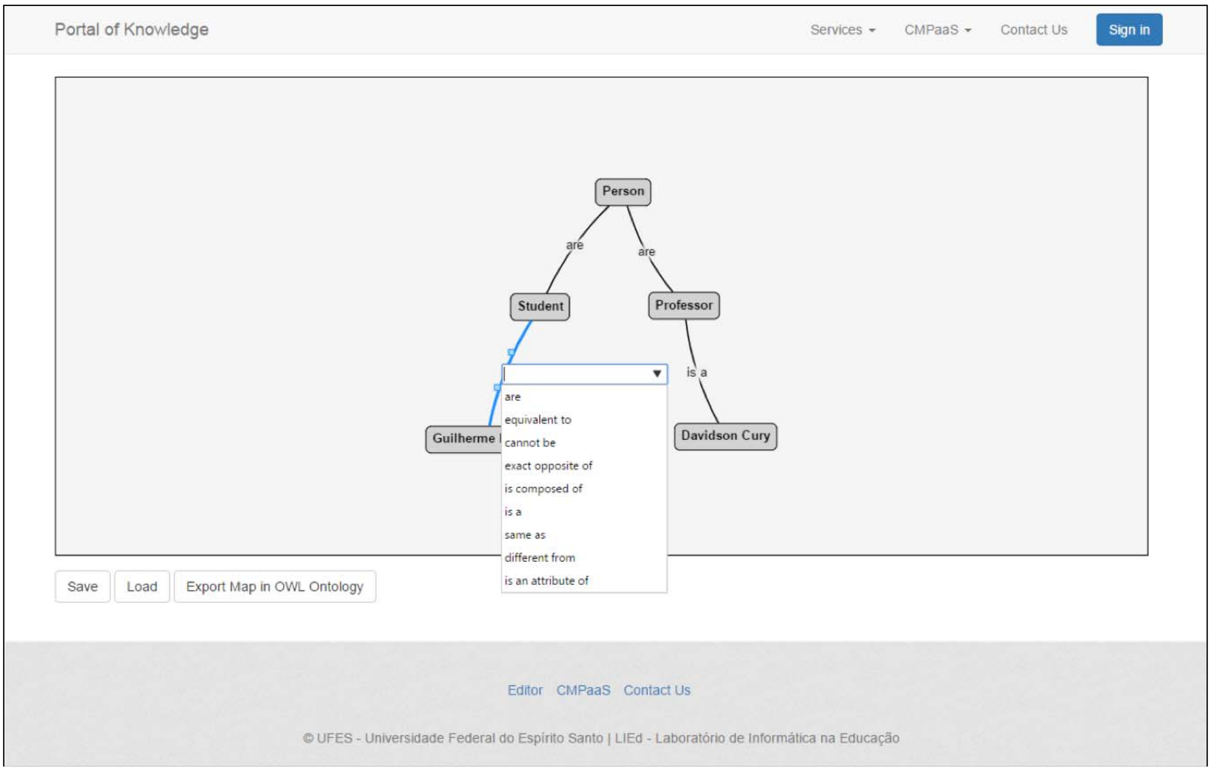

Figure 6. OntoMap’s Editor prototype.

[4] Available in: http://cmpaas.inf.ufes.br/
[5] Linked Data. Available in: <http://linkeddata.org/>

6 Discussion and Further work

Researches in Education are always looking for new ways to represent knowledge. Concept maps and ontologies are instruments of representation and knowledge building, each with their specific purpose and level of complexity. On one hand, concept maps are informal and easy to use. On the other, ontologies are formal and complex, requiring an expert in Knowledge Engineering to create them. Therefore, this study sought to unite these two worlds by giving the possibility to domain experts to create OWL ontologies (standard adopted by the Semantic Web), and then to share it on the web.

Each of the tools discussed in Section 3 were the basis for the idealization of the OntoMap tool. We are striving to maintain the main advantages of the analyzed tools and eliminate what we see as weaknesses. However, the OntoMap maintains certain differences from the others. Being part of a larger environment, our solution is one of the services of a web platform. To our editor, for example, we decided to keep the stereotypes of COE and the tabs from the MAP2OWL. Thus, we were able to maintain the simplicity of the editor of concept maps without visually pollute the user's view, giving him/her the chance to decide what to display on the screen.

It is important to say that this work is only an early version. In order to use and validate our tools, we have a course teaching how to use concept maps in virtual environments, using our Portal, for teachers of public schools in Brazil, Espírito Santo. This course will start in a few months of this year and it will be important for the dissemination and use of our tools as well as their validation and improvement. Thus, future work converges in improving the ontology editor (visual part) as the existing service, which will consist in seeking for the incorporation of new stereotypes of relationships in order to map other constructors of the OWL language. Moreover, we intend to provide the shallow ontologies created on our platform to the Linking Open Data, by means of a link with the DBpedia. To perform it, we are intending to start the development of the Integrator module and the Ontology Database.

References

Aguiar, C. Z., Cury, D. & Gava, T. (2015). Um Estudo sobre Abordagens Tecnológicas para a Geração de Mapas Conceituais. Proceedings of the 20th Congreso Internacional de Informática Educativa, TISE 2015, Santiago, Chile. p. 136-146.

Ausubel, D. P., Novak, J. D., & Hanesian, H. (1968). Educational Psychology: A Cognitive View. New York: Holt, Rinehart and Winston.

Brachman, R. & Levesque, H. (2004). Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc.

Chen, P. (1976). The Entity-Relationship Model: Towards a Unified View of Data, ACM Transactions on Database Systems 1.

Cury, D. & Menezes, C. S. (2012). A Portal of Knowledge based on Concept Maps. In A. J. Cañas, J. D. Novak & J. Vanhear (Eds.), *Concept Maps: Theory, Methodology, Technology*. Proc. of the Fifth Int. Conference on Concept Mapping, Malta: University of Malta.

Cury, D., Perin, W. & Santos Jr P.S. (2014). CMPaaS – A platform of services for construction and handling of concept maps. In P. Correia, M. E. I. Malachias, A. J. Cañas & J. C. Novak (Eds), *Concept Mapping to Learn and Innovate.* Proc. of the Sixth Int. Conference on Concept Mapping, Santos, Brazil: Univ. de São Paulo.

Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. & Stein, L. A. OWL Web Ontology Language Reference. W3C Working Draft, 31 March 2003. Available at http://www.w3.org/TR/2003/WD-owl-ref-20030331.

Dutra, Í. M., Fagundes, L. C., & Cañas, A. J. (2004). Un Enfoque Constructivista para el Uso de Mapas Conceptuales en Educación a Distancia de Profesores. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology.* Proc. of the First Int. Conference on Concept Mapping (Vol. x, pp. xxx-xxx). Pamplona, Spain: Universidad Pública de Navarra.

Garcia, A., Norena, A., Betancourt, A., García, L. & Sequeda, J. F. (2008). CMAPS supporting the development of OWL ontologies. Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany.

Gomez-Gauchia, H. & Diaz-Agudo, B. (2004). Two-layered Approach to Knowledge Representation using Conceptual Maps and Description Logics. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept*

Maps: Theory, Methodology, Technology. Proc. of the First Int. Conference on Concept Mapping, Pamplona, Spain: Universidad Pública de Navarra.

Guarino, N. (1998). Formal Ontology and Information Systems. In Proceedings of International Conference in Formal Ontology and Information Systems, pp 3-15.

Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models, PhD Thesis, Enschede: CTIT, Centre for Telematics and Information Technology.

Hayes, P., Eskridge, T. C., Reichherzer, T., Saavedra, R., Mehrotra, M. & Bobrovnikoff, D. (2005) COE: Tools for Collaborative Ontology Development and Reuse. In: Knowledge Capture Conference (K-CAP).

Jonassen, D. (1996). O uso das Tecnologias na Educação a Distância e as Aprendizagem Construtivista. Em aberto, Brasília, n.70, ano 16, abr./jun.

Novak, J. D. & Cañas, A. J. (2008). The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition.

Novak, J. D. & Gowin, D. B. (1984). Learning How to Learn. New York: Cambridge University Press.

Object Management Group (2003). UML 2.0 Superstructure Specification., Doc.# ptc/03-08-02.

Perin, W. A., Cury, D. & Menezes, C. S. (2014). NLP-Imap: Integrated solution based on question-answer model in natural language for an inference mechanism in concepts maps. In P. Correia, M. E. I. Malachias, A. J. Cañas & J. C. Novak (Eds), *Concept Mapping to Learn and Innovate*. Proc. of the Sixth Int. Conference on Concept Mapping, Santos, Brazil: Universidade de São Paulo.

Piaget, J. & García, R. (1988). Hacia una Lógica de las Significaciones. Buenos Aires, Centro editor de América Latina.

Pinotte, G. N., Cury, D. & Zouaq, A. (2015). OntoMap: De Mapas Conceituais a Ontologias OWL. Proceedings of the 20th Congreso Internacional de Informática Educativa, TISE 2015, Santiago, Chile. p. 172-180.

Siau, K., Erickson, J. & Lee L.Y. (2005). Theoretical vs. Practical Complexity: The Case of UML. Journal of Database Management, vol. 16, no. 3, pp. 40-57.

Simón, A., Ceccaroni, L. & Rosete, A. (2007). Generation of OWL Ontologies from Concept Maps in Shallow Domains. In: Conference of the Spanish Association for Artificial Intelligence, 12., Salamanca.

Starr, R. R. (2009). Uso de Mapas Conceituais como Suporte à Aquisição Autônoma de Conhecimento. Master's Degree Dissertation, Instituto Tecnologico da Aeronautica (ITA), Sao Paulo, Brazil. Tavares, Romero. 2007. "Construindo mapas conceituais." Ciências & Cognição 12: 72-85.

Studer, R., Benjamins, R. & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering, 25(1):161–197.

Vassoler, G. A., Perin, W. A. & Cury, D. (2014). MergeMaps – A Computacional Tool for Merging of Concept Maps. In P. Correia, M. E. I. Malachias, A. J. Cañas & J. C. Novak (Eds), *Concept Mapping to Learn and Innovate*. Proc. of the Sixth Int. Conference on Concept Mapping, Santos, Brazil: Universidade de São Paulo.

Zouaq, A., Nkambou, R., & Frasson, C. (2007). Document Semantic Annotation for Intelligent Tutoring Systems: A Concept Mapping Approach. In FLAIRS Conference.