Using Ontology Research in Semantic Web Applications: A Case Study

J. Aguado, A. Bernaras, T. Smithers, C. Pedrinaci, M. Cendoya

San Sebastian Technology Park, Paseo Mikeletegi 53
20009 San Sebastian, Spain
{jessica, amaia, tsmithers, carlos}@miramon.net
mcendoya@miramon.es

Abstract. In the light of improving the World Wide Web, researchers are working towards the Semantic Web. Ontologies and ontology-based applications are its basic ingredients. Several ontological environments, categorizations and methodologies can be found in the literature. This paper shows how we have investigated the state of the art in these areas in an ontology building process that is the basis for an application developed at a later stage in an events organisation domain

1 Introduction

Two main components are required for the Semantic Web to become a reality with defined business value: *Ontologies* and ontology-based *applications*. Realising the Semantic Web goes from ontology building to its transparent integration in an application. We show an ontology building process that is the basis for an application developed at a later stage in an events organisation domain. We first investigated the state of the art in three key aspects that a knowledge engineer needs to understand before engaging into an ontology-based application building: ontological environments, categorizations and methodologies.

2 An Ontology-Based Application: Online Design Of Events

Online Design of Events is an ontology-based application that aims to support customers and suppliers in the process of organising events such as symposiums, conferences or exhibitions. It is being developed within the EU funded OBELIX project (IST 2001-33144) via a web-based system that will support the collaborative process of designing events [1].

Fig.1 presents a view of the system architecture. It shows how different ontologies needed are integrated to provide the desired support to event designers and clients. This system architecture is derived from a Knowledge Level theory of designing: KLDE [2,3,4,5]. Applying KLDE made it possible to identify the kinds of knowledge-based support needed, and how the different kinds of knowledge need to be integrated.

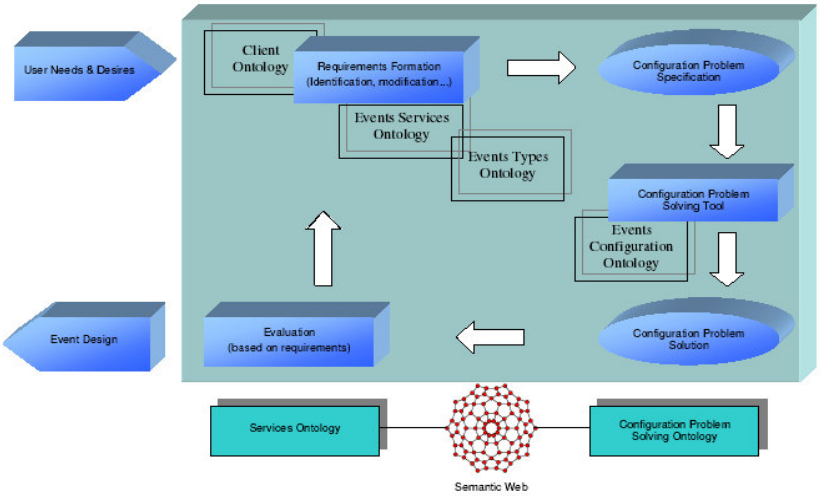

Fig. 1. Architecture of the *Online Design of Events* application, ontologies required and how they relate to each other

The event design support system integrates a *Client Ontology*(suports identification and construction of the requirements set), an *Events Services Ontology* and *Events Type Ontology* (support formulation of well formed problems) and an *Events Configuration Ontology* used in "Event Configuration Problem Solving Toolbox". Each of them model different kinds of knowledge, i.e. *Events Types ontology*, contains different types of events that can be configured in terms of resources used.

3 Ontological environments, categorizations and methodologies

In this section we ilustrate how the state of the art of three key aspects in an ontology building process, is considered in an example case: *Events Types ontology* building.

Ontology development environments include tools for ontology building, merging, mapping, integrating, querying...[6,7]. We reviewed the state of the art of: Ontosaurus, OntoEdit, WebOnto, Protege, OilEd, Ontolingua and WebODE. From the overview of existing surveys on these envionments we preselected Protege[8], WebODE[9] and OntoEdit[10]. Last, we selected OntoEdit since it covers our needs: it allows ontology storage in databases, it is language independent and it integrates an inference engine.

Researchers agree that it is useful to categorize ontologies depending on the levels of generality. I.e., in [11] *domain*, *generic* , *application, representational*, *task* and *method* ontology types are defined. In [12] N. Guarino defines *top-level*, *domain*, *task* and *application* ontologies. Both characterisations are close to each other, i.e. both refer to *application* and *domain* ontologies as the same and what in one case is called *generic* ontology in the other is called *top-level* ontology. We classify the *Events Types ontology* as a *domain* ontology, since it represents the events organisation domain. As such, (i) it represents the event organisation domain in a generic way so that it is shareable and reusable among different applications (generic terminology to be used by any events organising entity); (ii) it is specific enough to be used by the *Online Design of Events* application (constraints and rules are general and usable by any event organising entity setting some restrictive values to its needs).

There are several methodologies for building ontologies both from scratch and based on existing ontologies. [13] reviews some well known methodologies such as (KA)2, Cyc, SENSUS, KACTUS, METHONTOLOGY, CO4, Uschold and King, Grüninger and Fox, and On-To-Knowledge. Appart from this in [14] five approaches for ontology design and modification are proposed: *synthetic* (based upon existing ontologies), *inspirational* (individual viewpoint of the domain), *collaborative* (several viewpoints of the domain), *inductive* (specific case within the domain) and *deductive* (general principles of the domain). Hybrids are also possible. In the *Events Types ontology* development, *synthetic*, *inductive* and *collaborative* approaches are used.

3.1 Events Types Ontology

Having categorized the ontology, selected what approaches use and the tool for building it, these are the steps followed to reach the *Events Types Ontology*.

We categorized it as a domain ontology so we first ***define the domain and scope of the ontology,*** that in this case is events organisation.

We next ***look for an ontology for reuse*** (synthetic approach). In the events domain *OntoWeb ontology* is identified and enriched with information of events hold in PTSS (inductive approach).

Several partners from OBELIX take part in the dicussion of the results obtained in next steps (collaborative approach). Next step is to ***enumerate important terms of the domain,*** that have to appear in the ontology; i.e., the *Events Types ontology* represents knowledge about *event types*, *equipment* used, *layout* of *participants*... Afterwards we ***define classes and class hierarchy*** with the previously identified terms, defining them as concepts and subconcepts. I.e., the *event_type* concept has *Meeting*, *Exhibition*, *Conference ...* subconcepts. There are concepts without subconcepts too, i.e. *Event* (specifies when the event starts, finishes and what event types it consists of). Next we ***define properties/slots.*** Each concept is mainly defined by its properties/slots, that are inherited by its subconcepts, i.e., *event_type* has *hasRoom*, *hasLength*, *hasSetUp*,..., properties, that are inherited by *meeting*. Afterwards we ***describe the defined properties*** in terms of cardinality, value type and range, based on events hold in PTSS.

Next we make the ***implementation***. Concepts and properties are modelled with OntoEdit and ranges of properties are built as FLogic axioms. Once the ontology is implemented we look how it suits with *Online Design of Events* application: the need of *Equipment* instances is identified. So we last ***create instances.***

The resulting *Events Types ontology* consists of a taxonomy, instances and axioms. The ***taxonomy*** is the conceptual hierarchy of *event types* and *resources* implemented with OntoEdit and exported to Flogic (to be used by the application). We identified two kinds of ***instances***: dynamic instances (built when running the application) and static instances, that are the ones considered as part of the ontology (represent equipment resources of PTSS). ***Axioms*** represent rules and constraints of the domain. RDF is not enough for representing axioms so they are written in FLogic.

4 Conclusions and Future Work

This paper describes an ontology building process based on an analysis of current state of the art in ontology building related research. We tested the feasibility of integrating the ontology in a first version of the *Online Design of Events* application [15]. The conclusions were that support in the ontology runtime should be improved and that this version of the ontology it is well formed and usable from an application.

Next steps in the *Events Types ontology* involves its integration with a generic ontology such as the *Service Ontology*, described in [16]. In this version of the *Events Types ontology*, *meeting* is related with equipment such as *screen*, *videoprojector*, *PC* … because they are equipments of projection type. In future versions, *meeting* will be related to *projection services*, and this services will be related to the projection type equipment as shown in [17]. This way of representation supports better the customer's expression of needs, in terms of services rather than in terms of equipment.

References

1. Cendoya, M., Bernaras, A., Smithers, T., Aguado, J., Pedrinaci, C., Laresgoiti, I., García, E., Gómez, A., Peña, N., Morch, A.Z., Sale, H., Langdal, B.I., Gordijn, J., Akkermans, H., Omelayenko, B., Schulten, E., Hazelaar, B., Sweet, P., Schnurr, H.P., Opperman, H. and Trost, H. : D3 Business needs, Applications and Tools Requirements. Chapter 2. (2002)
2. Smithers, T., 1986: On Knowledge level theories of design process, in J S Gero anF Sudweeks (Eds.) AI in Design '96, Dordrecht: Kluwer AcademicPublishers.
3. Smithers, T., 1988: Towards a knowledge level theory of design process, in J S Gero an F Sudweeks (Eds.) AI in Design '98, Dordrecht: Kluwer Academic Publishers.
4. Smithers, T.: On Knowledge Level Theories and the Knowledge Management of Designing. International design conference - Design 2002 (2002).
5. Smithers, T.: Synthesis in Designing, in J S Gero (ed), AI in Design '02, Kluwer Academic Publ, Dordrecht, pp 3-26 (2002).
6. IST-Ontoweb Project, Technical Roadmap, Deliverable D11. November 2001. Available at http://www.ontoweb.org/download/deliverables/D11_v1_0.pdf
7. Das, A., Wu, W., and McGuinness, D.: Industrial strength ontology management. First Semantic Web Working Symposium, SWWS-01. (2001).
8. OntoEdit web page: http://ontoprise.de/products/ontoedit_en Last visited: june 2003.
9. Protégé web site: http://protege.stanford.edu/ Last visited: june 2003.
10. WebODE web site: http://delicias.dia.fi.upm.es/webODE/ Last visited: june 2003.
11. R. Studer, V. R. Benjamins, and D. Fensel.: Knowledge engineering, principles and methods. Data and Knowledge Engineering, (1998)
12. Guarino,N.: Formal Ontology and Information Systems. FOIS'98 (1998)
13. Fernandez, M.: Overview of Methodologies for Building Ontologies, IJCAI-99 Workshop on Ontologies and Problem Solving Methods (1999).
14. Holsapple, W.C and Joshi, K.D.A: Collaborative approach to ontology design. Communication of the ACM, vol45 no2 (2002)
15. Maier, A., Aguado, J., Bernaras, A., Laresgoiti, I., Pedrinaci, C., Peña, N. and Smithers, T.: Integration with Ontologies. Second Conference on Knowledge Management. (2003)
16. Baida, Z., Akkermans, H. and Gordjin, J.: Serviguration: Towards Online Configurability of Real-World Services. To appear in ICEC'03, (2003).
17. Baida, Z., Akkermans, H., Bernaras, A., Aguado, J. and Gordjin, J.: The Configurable Nature of Real World Services: Analysis and Demonstration. To appear in ICEC'03, (2003).