

Edhibou: a Customizable Interface for Decision Support in a Semantic Portal

Fadi Badra, Mathieu d'Aquin, Jean Lieber, Thomas Meilender

▶ **To cite this version:**

Fadi Badra, Mathieu d'Aquin, Jean Lieber, Thomas Meilender. Edhibou: a Customizable Interface for Decision Support in a Semantic Portal. 7th International Semantic Web Conference - ISWC 2008, Oct 2008, Karlsruhe, Germany. inria-00336330

HAL Id: inria-00336330

https://inria.hal.science/inria-00336330

Submitted on 3 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EdHibou: a Customizable Interface for Decision Support in a Semantic Portal

Fadi Badra
LORIA (UMR 7503 CNRS–INPL–INRIA-Nancy 2–UHP)
Vandœuvre-lès-Nancy, France
badra@loria.fr

Mathieu d'Aquin
Knowledge Media Institute
The Open University
United Kingdom
m.daquin@open.ac.uk

Jean Lieber
LORIA (UMR 7503 CNRS–INPL–INRIA-Nancy 2–UHP)
Vandœuvre-lès-Nancy, France
lieber@loria.fr

Thomas Meilender
LORIA (UMR 7503 CNRS–INPL–INRIA-Nancy 2–UHP)
Vandœuvre-lès-Nancy, France
meilendt@loria.fr

ABSTRACT

The Semantic Web is becoming more and more a reality, as the required technologies have reached an appropriate level of maturity. However, at this stage, it is important to provide tools facilitating the use and deployment of these technologies by end-users. In this paper, we describe EdHibou, an automatically generated, ontology-based graphical user interface that integrates in a semantic portal. The particularity of EdHibou is that it makes use of OWL reasoning capabilities to provide intelligent features, such as decision support, upon the underlying ontology. We present an application of EdHibou to medical decision support based on a formalization of clinical guidelines in OWL and show how it can be customized thanks to an ontology of graphical components.

1. INTRODUCTION

The Kasimir project is a multidisciplinary project which aims at providing oncology practitioners of the Lorraine region of France with decision support and knowledge management tools. The Kasimir system is a clinical decision support system which relies on the formalization of a set of clinical guidelines issued by the regional health network. It uses decision knowledge contained in an OWL ontology to provide decision support to clinicians. In such an ontology $\mathcal{O}$, a class `Patient` denotes the class of all patients, a class `Treatment` denotes the class of all treatments and a property `recommendation` links a class of patients to a class of recommended treatments. Then to a class `P` of patients is associated a treatment `T` by an axiom

$$\texttt{P} \sqsubseteq \exists \texttt{recommendation.T} \quad \text{where} \quad \begin{cases} \texttt{P} \sqsubseteq \texttt{Patient} \\ \texttt{T} \sqsubseteq \texttt{Treatment} \end{cases} \qquad (1)$$

A medical situation is represented by an instance `a` of the class `Patient` in the ontology $\mathcal{O}$. The system then exploits axioms of the form (1) to associate a set of recommended treatments to the patient represented by `a`. Deciding which treatments to recommend to the patient represented by `a` amounts to finding the most specific atomic concepts `T` in $\mathcal{O}$ such that $\models_{\mathcal{O}} (\exists \texttt{recommendation.T})(\texttt{a})$ holds.

The original motivation when developing EdHibou was to provide a user interface for the Kasimir system that lets the user describe a medical situation for which a decision has to be taken. Such a graphical user interface should let the user complete the description of an OWL instance `a` and trigger some reasoning tasks on the underlying OWL representation of clinical guidelines to propose a set of recommended treatments. We built EdHibou as a generic framework, allowing application developers to generate customizable interfaces to ontologies and ontology reasoning. The Kasimir system takes advantage of this framework as an application of EdHibou. The key idea in EdHibou is to allow the end-user to edit an OWL instance without having to manipulate the OWL syntax, by simply filling in values in a form. When developing this application, the main requirements were to make it generic — so that it can be easily reused in other applications, and easy to deploy. It also had to be customizable.

2. SYSTEM ARCHITECTURE

Our goal in developing EdHibou was to build a lightweight knowledge edition tool with (1) a very flexible knowledge model, and (2) highly configurable knowledge acquisition forms. Apart from the dynamic user interface update mechanism, anything had to be configurable, including the choice of the components to display and how they are displayed. Requirement (1) has been met by externalizing the knowledge model to a distant knowledge server. The role of this knowledge server is to manage a knowledge base and perform all reasoning tasks over OWL ontologies. Requirement (2) was fulfilled by pushing application configuration into an ontology. The generation of the user interface is then handled by a simple wrapper that takes as input an automatically generated XML representation of the content of an ontology together with a set of graphical component implementations.

EdHibou implements a Model-View-Controller architecture pattern (see figure 2) and was developed using the Google Web Toolkit Java AJAX programming framework. K-OWL, the knowledge server, is a standalone component that plays the role of the model. Though it manages knowledge, and not persistent data, K-OWL has been designed in quite the same spirit as standard database management systems. It

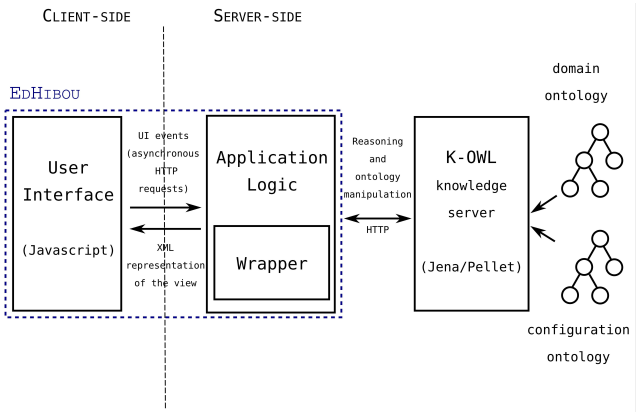


Figure 1: EDHIBOU's software architecture.

stores a set of Java models of OWL ontologies that are created with the Jena Java API coupled to the OWL DL reasoner Pellet. These ontologies are queried upon over HTTP using the SPARQL-DL query language [3]. Though K-OWL could be used remotely, it has been recently integrated to EDHIBOU's application logic for better performances.

3. AN ONTOLOGY-DRIVEN GRAPHICAL USER INTERFACE GENERATION

In EDHIBOU, all configuration information, that is information used to render the application knowledge model onto the user interface, is placed in a separate ontology. This ontology contains an exhaustive description of all graphical components used to build the user interface —including their Java implementation classes, as well as some decision knowledge used to determine which graphical component to associate to each property of the domain ontology. The application thus manages in a knowledge base an implementation-independant model of the user interface, and generating the user interface amounts to wrapping this model onto the application view. Such a wrapper has been implemented. Its role is to produce a suitable XML representation of the graphical components to display and transmit it to the application view for rebuild. Explicitly representing in OWL the configuration knowledge has many advantages. One of these advantages, compared to hard-coding decision procedures in Java classes, is to allow easy customization by a simple ontology extension mechanism. A default ontology $O_{gui-default}$ is provided that contains some default graphical components as well as basic decision knowledge. EDHIBOU can then be customized for particular domain ontology O_{domain} by replacing this default ontology by a customized ontology $O_{gui-domain}$ that extends $O_{gui-default}$. This new ontology may add new components, specify which components to use for a particular property or even add new decision knowledge to change the behavior of the application. This customization can be done by a knowledge engineer by the means of some ontology editor (or ultimately, by EDHIBOU, as it is itself a knowledge acquisition tool).

The set of graphical components to display is determined at runtime according to the description of the currently edited individual. EDHIBOU's default behavior is to select the set of graphical components to be displayed by testing the description of the currently edited instance against the definition domains of the different properties present in the ontology. To determine which components to display, EDHIBOU keeps only the properties `p` of the ontology for which `a` is an instance of the domain of `p` (according to the reasoner).

4. RELATED WORK

A number of systems have been developed with the aim of generating web interfaces on the basis of ontologies and RDF data. These systems generally consider the broad task of creating *semantic portals*, that are websites relying on semantic data. For example, ONTOVIEWS [2] is a tool to create such a semantic portal, presenting information contained in RDFS ontologies and providing navigation and search mechanisms within these ontologies. Another example is ODESEW [1], which generates complete *knowledge portal* dedicated to the publication and management of information in an organization. Compared to EDHIBOU, these tools are generally focused on the use of ontologies for the presentation of data in a website. ODESEW also includes a feature that generates forms to create and edit instances as a way to populate the portal. However, this functionality does not make use of the reasoning capability associated with OWL to guide the instance editing process or to infer new information from the elements entered by the user. For this reason, it could not be use as a way to provide advanced features exploiting the knowledge contained in the ontologies, like it is done in the KASIMIR project with clinical decision support, thanks to EDHIBOU.

5. CONCLUSION

EDHIBOU is a programmatic framework that enables to edit an OWL instance by the means of some user-friendly forms. It implements an ontology-driven graphical user interface generation approach and enables to exploit the standard reasoning on the underlying ontologies to provide intelligent behavior. An application of EDHIBOU is presented in which it is integrated in a semantic portal as a user interface for a decision support system in oncology. A first demo is currently available online at the URI `http://labotalc.loria.fr/Kasimir`.

6. REFERENCES

[1] O. Corcho, A. Gómez-Pérez, A. López-Cima, V. López-García, and M. Suárez-Figueroa. ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets. In *Lecture Notes in Computer Science Vol 2870. The Semantic Web - ISWC 2003*, pages 802–817. Springer-Verlag, 2003.

[2] E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. OntoViews - A Tool for Creating Semantic Web Portals. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *International Semantic Web Conference, ISWC 2004*, volume 3298 of *Lecture Notes in Computer Science*, pages 797–811. Springer, 2004.

[3] E. Sirin and B. Parsia. Sparql-dl: Sparql query for owl-dl. In *OWL: Experiences and Directions Third International Workshop (OWLED 2007)*, 2007.