BRAIN. Broad Research in Artificial Intelligence and Neuroscience,
ISSN 2067-3957, Volume 1, October 2010,
Special Issue on Advances in Applied Sciences,
Eds Barna Iantovics, Marius Măruşteri, Rodica-M. Ion, Roumen Kountchev

MY E-WALL: DESIGN OF A WEB PORTAL FULLY CUSTOMIZABLE AND ONTOLOGY BASED

Roberto Paiano and Andrea Roggerone

Abstract. Web is a measureless growing information source. Here we could have the common user that surf the web for fun, and the particular user surfing the web to work, for instance, the financial analyst that hopes to find some financial information.These users are similar in their habit to visit ever a pool of 10-15 websites where they are sure to find interesting information. In this paper, we introduce the architecture of a Rich Internet Application (RIA) that uses the power of Semantics to create a new type of Web Portal fully customizable that we will call MyE-Wall. In this portal, the user can't attach widget, but modules containing each one the section of a particular web site. Therefore, the idea is allow a user to create of a new website composed by a patchwork of different website's sections selected by himself. Here we use semantics for two reasons: (i) to process the structure of a particular website and create the corresponding module to attach on the portal (ii) to provide the application of a set of features (reasoning engine, swrl rules, etc...) to reason on the information collected, according the preferences that the user could have specified in his own profile.

Keywords: *Rich Internet Application, Ontology, Reasoner, Portal, semantic*

2000 *Mathematics Subject Classification*: 68M11, 68Q55

1.Introduction &Background

Both Web users that use Internet for fun and those that use it to work, they always visit the same 10-15 websites containing interesting information. This task is very time-expensive, especially for those users having the working necessity to find particular information, for instance, among hundreds website articles. Starting from these considerations, we can introduce the idea to develop in this paper. This idea consists in the creation of an ontology based Web Portal that we will call *My E-Wall*; it will be highly customizable like iGoogle (http://www.google.it/ig) and NetVibes (http://www.netvibes.com/#Generale) but here the user may attach particular modules containing some chosen website contents instead of widget. In literature we can't find any attempt to face the problem proposed in this paper, however we have found several approaches to minor problems related to the custom representation of an existing website and to the mapping of this one over an ontology. For example, Robert E. Filman in [1] highlights the existence of Platotypus (http://platypus.mozdev.org/) and GreaseMonkey (http://greasemonkey.mozdev.org/source.html). These Firefox add-on permits to customize all the page of a particular website, by adding or removing elements, rearranging pages etc...Therefore to obtain a website mapping over an ontology, we can adopt two approaches: (i) *Top-down* that uses a particular set of algorithms to generate the desired ontology from the webpages of a partcular website; this is a sort of HTML form-based reverse engineering[2] approach; (ii)*Bottom-up* creating the ontology starting from the website database and structure. The Architecture proposed in this paper is a Rich Internet Application (RIA), that use a Bottom-up approach to map over an ontology, the information content and the structure of a particular website. This ontology will be used to incorporate one or more section of a chosen website in a My E-Wall module. It will be possible to use the semantic sublayer to extract knowledge from the information contained in all the mapped websites, and propose them to the user according to his profile preferences. For instance, we can imagine that a financial analyst must daily visit the financial section belonging to fifty websites chosen by himself. This user must also collect financial information from the articles published in these websites, searching any information about the Google's shares. In this case, the application My E-Wall could permit to propose a more interesting section in a customized portal, and also it could permit to use semantics to filter the collected information,

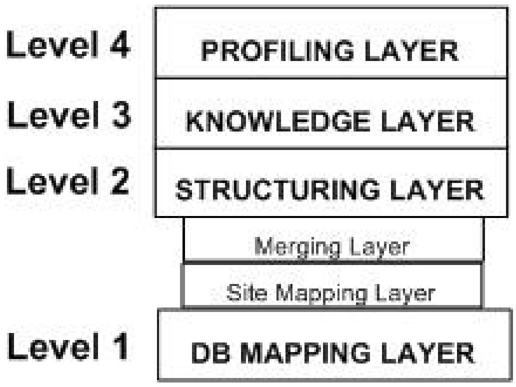


Figure 1: The Architecture

according to the user preferences. Until now we have only implemented the first two layer of the architecture that generates an ontology starting from the wesite's database. In the section 2 we present the proposed architecture. In the sections 3 and 4 we explain and detail all the architecture layers, instead the section 5 contains conclusions and future works.

2.The Proposed Architecture

Now we present the designed architecture for My E-Wall RIA. This architecture (Figure 1), is composed by fours layer. Level 1 and 2 have the task to generate an ontology for every chosen website, instead level 3 extracts knowledge by these ontologies. Level 4 is the highest layer of the architecture, and it must create all the modules to attach on the portal according to the preferences collected during the user profiling process.

Now we can describe all the layer composing our architecture: later we will detail layer 1 and 2 that we have implemented.

1.*DB Mapping Layer*: the first layer of the architecture, map the database of a particular website over an ontology. It must be generated an ontology for each website considered. We will call *DB Ontology* the ontology generated in this layer.

2.*Structuring Layer*: it must generate a new ontology reflecting a particular website structure, and it must merge it with the DB Ontology. The Structuring Layer is composed by two sublayers: (i) the *Site Mapping Sublayer* map the website structure over a domain ontology that we will call *Site Structure On-*

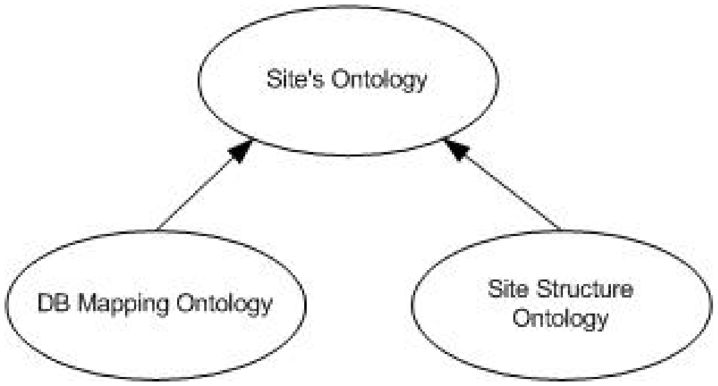


Figure 2: Site's Ontology

tology; (ii) the Merging Layer merge the DB Ontology with the Site Structure Ontology creating a new ontology called *Site's Ontology* (Figure 2).

3.*Knowledge Layer*: now we can suppose to have a Site's Ontology for every website chosen by the user. The task of this layer is to extract knowledge from all the Site's Ontologies, and it has the double function to send to higher layer:(i) the structure of every module to attach on the portal; (ii) additional information obtained through the reasoning task accomplished according to the user preferences.

4.*Profiling Layer*: it's the highest layer of the architecture where are implemented the Web portal features. Here is done the profiling of the user, and are dynamically generated the modules containing the sections of more internet websites chosen by the user. These modules will be attached on My E-Wall. In this layer some RSS feeds, are also pairwised to every module, in order to update the module in realtime.

3.DB Mapping Layer

This layer maps a database over an ontology following a set of default mapping criteria representing an improved version of those used in [3]. The implemented application[4] maps only the database structure, keeping instances into the database. In this way, the application can take advantage of the SQL power instead of the SPARQL that is very far to seem reliable. To maintain a common syntactic basis, the application compares all the terms contained in the database to map with those contained by WordNet (or by a domain ontology). If these terms are not found, is requested the help of a user, that must insert an equivalent term existing in WordNet. In the section below we will see the criteria adopted in the mapping phase, and a little case of study that use our mapping methodology.

DB Mapping Methodology

To map a database over an ontology, it first must be created an empty ontology, and then follow the above methodology:

STEP 1. For each database table, a class must be generated into the destination ontology.

STEP 2. Hierarchies must be established among classes. Analyzing the database records, we check if a table has a parent (as in disjoint relations). This test must be done using the logic exposed below. Having two table A and B, if:

1. all records of the primary keys of the table A are present among the primary keys of the table B, and
2. the number of records related to the primary keys of the table B is higher than a number of records related to the primary key of the table A,

the table A is a child of the table B, and so the corresponding Class A is a subclass of Class B. Disjoint and Union relations must be respectively translated in disjoined or joined subclass relation in the ontology.

STEP 3. For each attribute (excluding the Foreign Keys) of each database table, it must be generated a Data Property with a characterization of the data type (string, int, date etc...). Different tables could have attributes having the same name, but an ontology cannot contain two attributes with the same identifier (neither in different domains). For this reason it's necessary to map the attributes adding their domain name (DomainName_AttributeName) and if necessary to indicate if this attribute is a Primary Key (PK_ DomainName_AttributeName).

STEP 4. For each Foreign Key it must be generated an Object Property having : (i) the table containing the foreign key, as a domain; (ii) the table containing the corresponding primary key, as Range. The name of the Object Property will have the format DomainNamehasPK_PrimaryKeyName if a table foreign key is also the primary key for this table otherwise the format is DomainNamehasFK_ForeignKeyName.

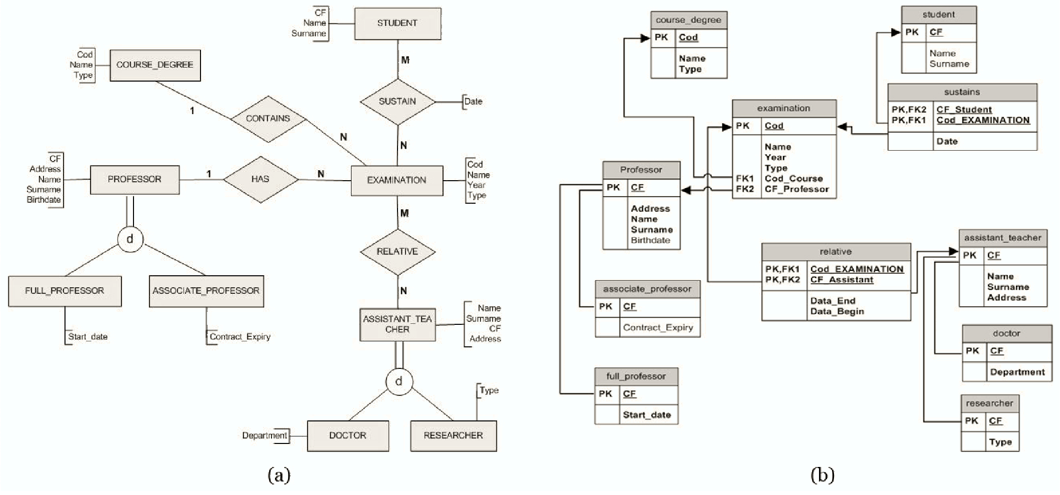


Figure 3: ExaminationDB: (a) E-R Model, (b) Relational Model

STEP 5. It must be created a restriction (cardinality=1) for each attribute related to the database primary key.

STEP 6. In the last step are created the restrictions on the remaining attributes:

1. if a database attribute is NOT NULL, in the destination ontology it must be inserted a minimum cardinality relation as mincardinality=1,

2. if a database attribute is NULL, in the destination ontology it must be inserted a minimum cardinality relation as mincardinality=0.

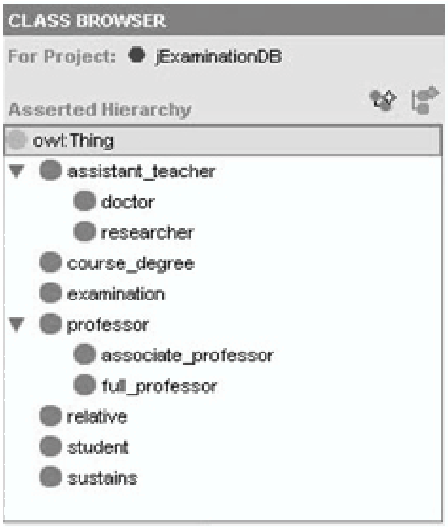
DB Mapping: a case of study

In order to explain how the methodology works, we analyze step by step, a simple mapping on a database sample named ExaminationDB (Figure 3).

STEP 1. Seven classes are generated: assistant_teacher, course_degree, examination, professor, relative, student, sustains.

STEP 2. Doctor and researcher are added as a set of disjoined subclasses of class assistant_teacher; associate_professor and full_professor are added as a set of disjoined subclasses of class professor (Figure 4).

STEP 3. 32 Data Properties are generated : 11 related to the primary keys (named PK_NomeDominio_NomeAttributo and 21 related to the others attributes and named NomeDominio_NomeAttributo. For each data type of a table attribute, exists a range of values in the corresponding Data Property in



CLASS BROWSER

For Project: ● jExaminationDB

Asserted Hierarchy

- owl:Thing
  - assistant_teacher
    - doctor
    - researcher
  - course_degree
  - examination
  - professor
    - associate_professor
    - full_professor
  - relative
  - student
  - sustains

Figure 4: ExaminationDB Ontology class and subclass

the generated ontology. For example the Data Property professor_Address is a string, because the attribute address of the professor table in ExaminationDb is a Varchar type.

STEP 4. 10 Object Properties are generated . Six Object Properties are related to the Foreign Keys (examinationhasFK_Cod_Course, examinationhasFK_CF_PROFESSOR etc. . .). Four Object Properties are related to the class involved in disjoint relations (associate_professor, full_professor, doctor, researcher), and they have the parent class as a Range and the primary key as a relation key. For example we have the 4 object Properties: researcherhasPK_CF, associate_professorhasPK_CF, full_professorhasPK_CF, doctorhasPK_CF.

STEP 5. It is created a restriction having a minimum cardinality equal to 1 or to 0, on each Data Property corresponding respectively to a NOT NULL (avoiding the PK) and to a NULL attribute. For example we will have assistant_teacher_Address min 1;

STEP 6. On each Data Property corresponding to a Primary Key attribute of the database source, is added a restriction having cardinality exactly 1.

4.Structuring Layer

This layer must create an ontology semantically describing a particular website. This ontology will be passed to the upper layer, and it will be used to extract the knowledge, useful to the web user. Since the fundamental web components are always the same, could be useful to have a standard ontology, to populate in the *Site Mapping Layer*, using the feature of the analyzed website. The Site Mapping Sublayer uses P-IDM[5] to model the ontology of a generic website.

PIDM defines two particular views:

1. a *structure view* defining the structure of the application in terms of page layout and the content of these pages;
2. a *contents view* defining more in detail both the real contents of the application and the interaction with the user.

Information given by these views mapped in an ontology, will be used in the upper level during the creation process of the module to attach on MyE-Wall board. The use of PIDM as a standard permits to represent any web site in a single way mapping both the content and the layout of it and This ontology will be merged with the DB Ontology in the Merging Sublayer, obtaining the Site's Ontology.

5.Conclusions

In this paper we have shown the design of a new type of Web Portal, called My E-Wall. This is an ontology-based RIA that uses the power of semantics to bridle information and whole website sections, into web module modules dynamically created and attached on the Portal. My E-Wall represents a new step towards a user-centric Web, that every user can customize. Our attention is mainly focused on the first two layers of the proposed architecture. We have implemented these levels, and now we can obtain an ontology describing both structure and content of a website. The next steps to face, will be the realization of the Knowledge Level that composes the semantic layer of our application, and the graphics implementation of the portal My E-Wall.

References

[1] Robert E. Filman. Taking Back the Web. IEEE Internet Computing, Vol. 10, No. 1, Jan. 2006, pp. 3-5

[2] Sidi Mohamed Benslimane Et al., Extracting Personalised Ontology from Data-Intensive Web Application: an HTML Forms-Based Reverse Engineering Approach, International Journal of INFORMATICA, Dècember 2007. IOS Press, ISSN 0868-495, pp. 511-534

[3] Wang, M. L.-Y,Learning ontology from relational database, Proceedings of International Conference on Machine Learning and Cybernetics, 2005.

[4] Roberto Paiano, Anna Lisa Guido, Semantic data integration: overall architecture. In International Business Information Management Association (IBIMA 09)- Innovation and knowledge Management in Twin Track Economies: Challenges & Solutions January 4 - 6, 2009 Cairo, Egypt ISBN: 978-0-9821489-0-7 pp. 430-436

[5] Roberto Paiano, Anna Lisa Guido, Andrea Pandurino, Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering Publisher: Information Science Reference (igi-global). Publication date April 2009, ISBN 978-1-60566-300, Page 319

Roberto Paiano, Andrea Roggerone
Department of Engineering Innovation
University of Salento
Via per Monteroni, 73100, Lecce, ITA
email:*roberto.paiano@unisalento.it, andrea.roggerone@gmail.com*