Building an Ontological Base for Experimental Evaluation of Semantic Web Applications*

Peter Bartalos, Michal Barla, György Frivolt, Michal Tvarožek,
Anton Andrejko, Mária Bieliková, and Pavol Návrat

Institute of Informatics and Software Engineering, Faculty of Informatics
and Information Technologies, Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia
{Name.Surname}@fiit.stuba.sk

Abstract. The increasing number of Semantic Web applications that work with ontologies implies an increased need for building ontological knowledge bases. In order to improve ontologies during their development as well as to allow applications to be experimentally evaluated prior to their complete implementation and deployment, ontology bases must be filled with experimental data (i.e., instance ontologies), which can be used to evaluate methods used for information processing. We describe several approaches together with a method of building an ontological knowledge base for purposes of experimentation with Semantic Web applications. We also discuss characteristics and suitability of particular approaches to the development of experimental ontological knowledge bases.

1 Introduction

The advent of modern software applications that take advantage of Semantic Web technologies caused an increasing demand for well-built ontologies, filled with a statistically relevant amount of (experimental) data of acceptable quality. The process of ontology creation is non-trivial and is ideally accompanied by a thorough analysis of the target domain [5]. Furthermore, experimental data are required prior to the application and/or the ontology deployment to experimentally evaluate the quality of the designed methods, as well as to identify possible problems and mistakes.

The Semantic Web community has not yet reached a de facto consensus on standard methods for development of large-scale ontologies. Several ontology development methodologies have been proposed (see overview of the methodologies in [5,12]) that obviously elaborate several basic steps of ontology development: identifying purpose, building the ontology, evaluation and documentation. The ontology building step is realized by one of the two primary approaches, which include different manual approaches and (semi)automatic approaches.

In this paper, we describe the main properties of both approaches with respect to the creation of Semantic Web applications Experimental Evaluation

* This work was partially supported by the Slovak State Programme of Research and Development "Establishing of Information Society" under the contract No. 1025/04.

(SWEE) ontologies. Furthermore, we propose a method of SWEE ontology creation and discuss the roles, advantages, disadvantages and suitability of individual approaches for the development of experimental knowledge bases in the Semantic Web environment. We describe our evaluation of the proposed method by giving examples from the domain of job offers.

2 Approaches to ontological base creation

2.1 Manual approaches

Manual SWEE ontology building is primarily based on the use of ontology editors [7], which can either be generic domain independent ontology editors such as Protégé (`protege.stanford.edu`) or Altova SemanticWorks (`www.altova.com/products/semanticworks.html`), or special custom made editors for specific ontologies such as JOE (`nazou.fiit.stuba.sk`). Since there are many ontological editors we do not discuss their individual properties or functionality. For anyone who is interested we recommend a survey made by Michael Denny that provides complex information about 94 ontology editors [6].

A natural characteristic of manual ontological base building is the presence of humans in the creation process, thus involving the human factor with both its advantages and disadvantages. At present, the involvement of humans theoretically contributes to higher quality of the created data because of superior human intelligence. For example, a human user can easily distinguish the minimal and maximal salary in a job offer or specific qualification requirements, whereas the automation of this process may be non-trivial or in most cases inaccurate.

The involvement of humans also has disadvantages in the form of mistakes that humans inherently make. Another somewhat negative aspect is the sole necessity of humans and the high amount of time that is required to produce even a relatively small amount of data of supposedly higher quality.

Generic ontology editors. Generic ontology editors present a straightforward approach to ontology definition for experts who understand the underlying principles of ontologies and their structure. They can also be used by less experienced users, who are normally not able to fill in large and complex ontologies since generic editors do not allow for any simplifications based on the structure of a particular ontology.

Generic editors only work with or "understand" the generic structure of the ontology as defined by the respective ontological language (e.g., OWL) and thus work with generic ontological concepts such as classes or properties in OWL. This makes them effectively domain independent, because they make no assumptions related to the structure or content of the ontology itself. While this can be considered to be an advantage as any kind of ontology regardless of its use can be created, it is a disadvantage if a SWEE ontology is not developed by ontology experts, but by inexperienced users who are often not disciplined enough to follow standards or best practices (as a consequence of ignorance of principles of ontologies), and thus introduce (many) errors.

Ultimately generic ontology editors provide good means for creating and manipulating ontologies for experienced users, while lacking proper support for the input validation and simple use by inexperienced users which are necessary for the creation of ontology instances.

Specialized ontology editors. Specialized editors for SWEE ontologies offer maximum freedom in adjusting to a given ontology and user requirements. The main benefit of using specialized editors is the correctness and consistency of the resulting data because many problems related to the use of generic ontology editors are resolved by the designers of specialized ontology editors. These can use their knowledge of the ontology to create a convenient application, which makes editing of ontological data intuitive, more effective and more reliable.

Disadvantages of specialized editors are the overhead costs of their development and maintenance. The problem is primarily the time needed for the editor development because it can not start until the ontology (or its early version) is deployed. Moreover, if it does not support automatic forms generation, the resulting application is tightly coupled to a specific version of the ontology and generally must be appropriately updated when the ontology changes.

3 Automatic approaches

Different automatic approaches can be used for SWEE ontology creation, which can either work with real-world data or with completely artificial data. For artificial data, the simplest automatic approach is to generate random data, which correspond to values of properties of the classes in the SWEE ontology. The prime disadvantage of this approach is that the data are random and thus have little meaning. This can be partially resolved by more advanced generators that use parts of existing instances to create seemingly realistic data.

More accurate data can be acquired from actual sources in a particular domain such as some database or a third party data repository accessible via a defined machine readable interface. The Amazon E-Commerce Service is a good example for the domain of articles and books. Another examples are DBLP (Computer Science Bibliography, `dblp.uni-trier.de`) and CiteSeer (Scientific Literature Digital Library, `citeseer.ist.psu.edu`), which both provide their data in the form of an XML file. However, only few sources are accessible in this way on the current Web. Alternatively, wrappers present an approach to real-world data acquisition from existing web sources.

Wrappers. For many newly developed ontologies counterparts in the form of Web portals, which provide the same or similar information already exist. For example, if a SWEE ontology for job offers had to be created, existing job portals can be used. Information presented on these portals can be extracted by means of web page wrappers – applications that parse web pages and produce structured outputs (in the form of XML, RDF, database, ontology).

Consequently, wrappers can be used to obtain this information from existing Web sources with some limitations mainly related to resources needed for

the wrapper development and problems related to its robustness due to frequent changes of web page design. The main advantage of wrappers is that large amounts of data can be acquired. Different types of wrappers can be implemented based mainly on the used language. There are several approaches ranging from scripting languages (Perl, Python) [9], through visual wrapper creation (Lixto [3]) to machine learning techniques [4].

Wrappers are efficient tools for the creation of a relatively significant amount of relatively simple data for a SWEE ontology. The implementation of the wrapper itself is reasonably time consuming, however the maintenance of a wrapper would pose a significant disadvantage during prolonged use, which is not necessarily required for SWEE ontology creation. If various data sources were used, data integration issues would become more pronounced and result in increasing yet still acceptable demands on human and time resources.

Generators. Generators take existing data from an already partially filled ontology and combine them to create seemingly new instances. The prime purpose of generators is to increase the size of a SWEE ontology by utilizing its existing content of known quality. In general, generators can create a SWEE ontology of the desired size, but there are practical limitations concerning the amount and acceptability of duplicate data in the final ontology. Another purpose of generators is to create new instances with such desired properties that are not covered in the original instances. Furthermore, since the sensibility of the generated data is important for a SWEE ontology, suitable instance generation algorithms must be developed in order to create usable data.

Generators are well suited for the creation of a SWEE ontology, but require a "large enough" set of high quality data to work with prior to their use. The development of a suitable generator requires a moderate amount of human and time resources. Once implemented, generators create a SWEE ontology of adequate size with quality depending on the "intelligence" of a generator.

4 Method for SWEE ontology creation

The employment of several different approaches is required to create a suitable SWEE ontology. We propose a method for SWEE ontology creation, which takes advantage of the individual benefits of different manual and automatic approaches (see Figure 1). The method consists of two basic steps:

1. Manual approaches are employed to develop the ontology and to create an initial set of experimental data of good quality. These take advantage of human intelligence, which is indispensable in the instance creation process in order to achieve the required level of detail and quality.
2. Automatic means are used to increase the size of the SWEE ontology to the desired volume of data. Wrappers are used to increase the variability and scope of data while maintaining at least the minimum acceptable level of detail. Generators are used to "synthesize" the required amount of data from

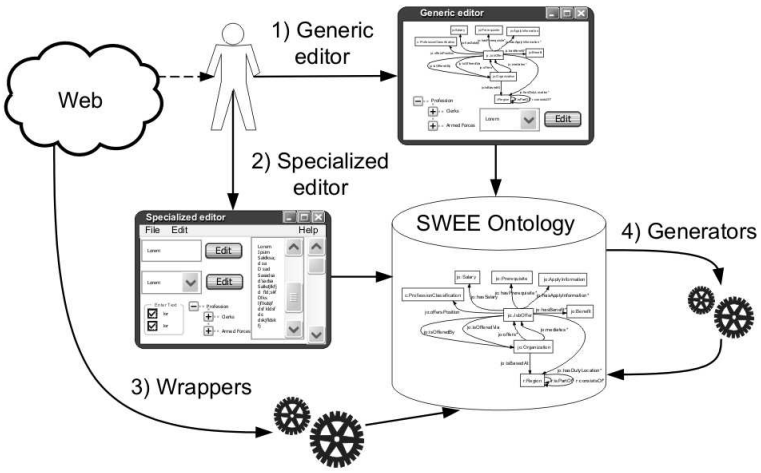

Fig. 1. Overview of the creation of a SWEE ontology.

existing instances and add details to instances created by manual approaches and wrappers missing due to insufficient data in source documents or due to the lack of "intelligence" in wrappers.

The initial creation of an ontology is to be performed using a generic ontology editor or an existing ontology can be used. Preliminary instance data are created manually so that most classes have coverage of some tens or hundreds of instances (the number depends on the complexity of the SWEE ontology). This preliminary part of the SWEE ontology creation is performed with generic ontology editors. At this point the use of specialized editors would be ineffective due to the high maintenance requirements of the SWEE ontology.

Once the ontology becomes "stable" and changes become less frequent, the use of specialized editors to speed up instance creation becomes feasible. Ideally, a specialized editor is found and configured for the ontology, or alternatively a new application is developed specifically for the given ontology.

Employment of several wrappers increases the size and broadens the scope of the SWEE ontology. Creating a wrapper for a specific site is quick with proper tools (visual wrapper designer environments). The process of wrapping itself is automatic and depends on the desired amount of acquired data together with Internet connection speed and access restrictions of particular sites. Although wrappers are designed by humans, they are automatic tools and as such can extract only a limited amount of data from a web page. While they are good at extracting data from (partially) structured web pages, e.g., from tables, they cannot be used effectively to extract data from unstructured text in natural language. As a result, acquired instances lack many of the details, which humans can input by means of generic/specialized ontology editors.

Generators are used to add details where necessary and add new instances until the desired size of the SWEE ontology is reached. It is only now that

generators can actually be used since they require enough existing instances to work with. If properly implemented, generators do not decrease the quality of instances since they reuse fragments of existing ones. Furthermore, many instances can be generated quickly with relatively little effort.

5 Evaluation of approaches to SWEE ontology development

Developing a SWEE ontology was motivated by work on a research project aimed at support of acquisition, organization and presentation of information on the Web for the online labor market domain [10]. Several cooperating software tools (`nazou.fiit.stuba.sk`) that realize a sequence of successive steps from acquiring data containing job offers from the Web [11] through identifying documents in which job offers are present, offers extraction, organization [8] up to their personalized presentation to the user [14] are developed. This could be characterized as the transformation of a part of the Web to the Semantic Web, where existing documents are transformed to a representation, which augments the presented information by utilizing semantic concepts and their automated processing. The need for the creation of a SWEE ontology for experimentation purposes became apparent as work on the project continued and methods for data and offer acquisition, analysis, organization, maintenance and presentation realized by individual software tools had to be experimentally evaluated.

The ontological representation of the domain itself is subdivided into several ontologies, which represent geographical and political regions, languages and currencies that are used in these regions, different hierarchical classifications (e.g., industrial sectors, professions, educational levels, qualifications) and generic offers respectively. The whole domain ontology is fairly large and complex enough such as to make it difficult for a single person to completely understand all the concepts it contains (a total of about 740 classes of which 670 belong to hierarchical classifications with a maximum depth of 6 levels).

We considered the following key requirements during the development of the SWEE ontology of job offers:

- A reasonably large amount of individual instances had to be created, so that conclusions analogous to those based on a statistically relevant amount of data could be made. The volume of the data should also enable performance tests, important in the Web environment.
- Instances from various sources were needed to simulate heterogeneous sources of data (i.e., we had to process job offers from different job offer portals).
- Instances with a broad range of properties were needed to create a rich enough ontology. As a result, job offers from different industrial sectors, with various positions and employers were gathered in our project.
- Instances with different levels of detail were needed to simulate the availability of data or lack thereof. Various job offer portals provided more or less details about job offers.

– Instances of different quality were needed to simulate human and/or other errors in source data. Moreover, the different quality levels of the data has to be known as they are important for the simulation of inaccuracy of software tools for data and offer acquisition that is inevitable in the case of automatic offer acquisition from the Web.

5.1 Creation of a SWEE ontology of job offers

During the development of an ontological base for our project, we employed a distributed approach with both generic (Protégé) and specialized editors (JOE, Job Offer Editor) [2]. Based on the gathered experiences we also developed a web-based application for dynamic form generation for ontology instances [1].

In the first stage, we distributed about a hundred of source documents (HTML pages) with job offers manually acquired from the Web among people involved in the project who manually filled the ontology with instances of job offers using the generic ontology editor Protégé.

Once integrated, the resulting SWEE ontology (ignoring its size) was suitable for initial evaluation of software tools despite the fact that it contained a lot of inconsistencies and faults. The most common problems were missing data, data input into wrong properties, incorrect IDs, inconsistent and incorrect use of taxonomies. The majority of instances had significant "problems" with complex taxonomies used to express requirements imposed on job applicants.

After the evaluation of the first stage of base creation we invested resources into the rapid development of a specialized standalone single purposes desktop editor JOE (Job Offer Editor). Especially, we needed to increase the annotation speed because instance creation was very slow. Before JOE was employed, one person was able to create ontology instances at a mean rate of 3 job offers per hour. With JOE the rate increased to 5-6 job offers per hour.

In order to further enlarge our SWEE ontology we developed an environment for wrapper creation, which enables users to specify a wrapper able to extract data from web sites and store it in a structured format (XML files or ontological repositories) [13]. We performed several sessions during which we acquired data from different web portals. All sessions consisted of phases of wrapping and integration. We gathered 1 937 job offers from six sources (*EuroJobs*, *CareerBuilder*, *LinkedIn*, *TotalJobs*, *UKworkSearch*, *BritishJobs*). We encountered several problems common to web page wrappers such as inconsistent, ambiguous, missing or incorrect data (e.g., abuse by advertisements).

The implementation of a wrapper using our tool took about one day of developer time but we had to spend much effort on the integration of data since every job offer portal used a somewhat different structure or classification of data.

In order to increase the size of the SWEE ontology base we developed several generators that produce different sets of ontology instances that satisfy several (often contradicting) requirements. We concentrated on creating such instances, which extensively use taxonomies defined in our ontology. Different logically separable parts of entities (e.g., salary, position, benefits) were taken and combined into new instances. This combination was arbitrary based on the use of random

generators. We defined the structure of the resulting instance using several methods – random, statical, or taking an existing instance/class from the ontology as a template. Our generator implementation needed an average of 10s (4-20s) to generate a job offer instance. We have generated thousands of new instances.

The use of generators leads to the creation of a sufficient amount of new instances, but the generated instances are not assured to be completely meaningful. While some concepts can be mixed arbitrarily without the creation of meaningless data (e.g., salary), others are strongly coupled (e.g., position and prerequisites). Due to these reasons we analyzed the relations between properties of instances. The knowledge of these relations allowed us to adjust the generation process to create instances, which maintain their meaning.

5.2 Discussion and related work

The presented method of SWEE ontology creation was successfully used in the domain of job offers. While developing our SWEE ontological base, we validated that different approaches are needed to create a suitable SWEE ontology. Table 1 shows the main properties of approaches used in our method.

Table 1. Key properties of approaches to SWEE ontology development.

	Generic	**Special**	**Wrapper**	**Generator**
Tool development cost[a]	none	high	medium	medium
Instance creation speed[b]	low	medium	high[c]	very high
Instance creation cost[d]	high	medium	low	very low
Performance/cost ratio[e]	medium	low	medium	high
Standard level of detail[f]	high	high	low	high
Errors introduced into instances[g]	high	low	low	low
Resemblance to real web data	high	high	medium	low[h]
Typical number of instances[i]	hundreds	hundreds	very high[c]	very high
Human involvement required[j]	yes	yes	no	no
Cost of ontology change[k]	none	very high	high	medium
Cost of data source change[k]	none	none	high	none

[a] The relative amount of resources spent on tool development directly by the ontology developer assuming that an existing generic editor is used.
[b] The relative amount of time required to create an ontology instance.
[c] Disregarding the limitations of the communications link and the web site host server.
[d] The relative amount of human and time resources required to create an instance.
[e] The overall effectiveness based on the number of created instances and the total cost.
[f] The level of detail normally achieved during instance creation.
[g] The relative amount of errors introduced during instance creation.
[h] More realistic data can be created with more advanced generators.
[i] The number of instances that can be created with a reasonable amount of resources.
[j] Concerns direct human involvement in the instance creation process.
[k] Maintenance cost related to adjustments to changes in the ontology or data source.

The presented results support the conception that the combination of different approaches leads to the best results. When requirements on the SWEE ontology are properly defined, using the presented method allows to create a result that maximally satisfies the specified needs. None of the approaches can separately fully satisfy the requirements. When development and maintenance costs are considered, the best solution is a generic ontology editor. Although one can use the generic editor immediately, only a limited amount of high quality data can be created. Some limitations of generic ontology editors can be eliminated with special editors for the price of development cost. The limitations of manual approaches respective to the amount of instances created can be eliminated employing wrappers, which can create large amounts of data usually with a lower standard level of detail than other approaches. To guarantee that the SWEE ontology contains a large amount of data with a high level of detail generators are the best option.

The field of the Semantic Web and especially experimenting with the results is currently rather immature. Datasets for experimental evaluation of methods and techniques exist in various fields. We name here at least the well known dataset in the UCI Knowledge Discovery in Databases Archive (`kdd.ics.uci.edu`) that serves for data mining in database methods evaluation. Up to our best knowledge, no such dataset exists for the Semantic Web community in the form of a sufficiently large ontology with thousands of instances.

Several research groups attempt to tackle the problem of acquiring, analyzing, organizing and presenting information and knowledge from the Web, such as project AKTORS (`www.aktors.org`) supported by the British government, projects supported by the European Union, i.e., Knowledge Web (`knowledgeweb.semanticweb.org`), On-To-Knowledge (`www.ontoknowledge.org`), REWERSE (`rewerse.net`), or project SIMILE (`simile.mit.edu`) that is a result of cooperation of a consortium consisting of W3C, MIT Libraries and MIT Computer Science and Artificial Intelligence Laboratory. These projects use ontologies from various domains developed just for specific purposes of particular aspects of each project. Our approach together with the developed SWEE ontology for the job offer domain has the potential to serve as a "common ground" used for experimental evaluation and comparison of various methods developed for the Semantic Web environment.

6 Conclusions

We described a method for the creation of Semantic Web applications Experimental Evaluation (SWEE) ontologies together with examples from the job offers domain. The proposed method can serve the Semantic Web community for experiments with software tools being developed. Different means used for ontological test base creation improve its usability in experimental evaluation of Semantic Web applications. Without a sufficiently large ontological base no serious experimenting with the implemented Semantic Web method can be made.

We showed that several approaches are required to develop a high quality ontological test base with various characteristics that cover diverse situations occurring in the environment, where applications manipulating ontologies might operate. Several applications ranging from gathering data from the Web, analyzing and organizing data such as duplicate instance removal, clustering and other data mining applications, to personalized presentation of gathered data may benefit from a SWEE ontology of non-trivial size.

References

1. M. Barla, P. Bartalos, P. Sivák, K. Szobi, M. Tvarožek, and R. Filkorn. Ontology as an Information Base for Domain Oriented Portal Solutions. In *Proc. of 15th Int. Conf. on Information Systems Development, ISD'06*, Budapest, Hungary, 2006.
2. P. Bartalos and J. Malečka. Building ontological test base using specialized ontology editor. In P. Návrat et. al, editor, *Proc. of the Workshop on Acquiring, Organising and Presenting Inf. and Knowledge on the Web*, Slovakia, 2006.
3. R. Baumgartner, S. Flesca, and G. Gottlob. Visual web information extraction with Lixto. In P.M.G. Apers et al., editor, *Proc. of 27th Int. Conf. on Very Large Data, VLDB'01*, pages 119–128, Roma, Italy, 2001. Morgan Kaufman.
4. M. Ceresna. *Supervised Learning of Wrappers from Structured Data Sources*. PhD thesis, Vienna University of Technology, 2005.
5. J. de Bruijn. Using ontologies – enabling knowledge sharing and reuse on the semantic web. Technical Report DERI-2003-10-29, DERI, 2003.
6. M. Denny. Ontology tools survey. *O'Reilly XML.COM*, 2004. Available at `www.xml.com/pub/a/2004/07/14/onto.html`.
7. H. Eriksson, R. Fergerson, Y. Shahar, and M. Musen. Automatic generation of ontology editors. In *Proc. of the 12th Banff Knowledge Acquisition Workshop*, Banff, Alberta, Canada, 1999.
8. P. Gurský, R. Lencses, and P. Vojtáš. Algorithms for user dependent integration of ranked distributed information. In M. Böhlen et. al., editor, *Proc. of TED Conference on e-Government, TCGOV'05*, Bozen-Bolzano, Italy, 2005.
9. J. Hammer, H. Garcia-Molina, J. Cho, A. Crespo, and R. Aranha. Extracting semistructured information from the web. In *Proc. of the Workshop on Management fo Semistructured Data*, 1997.
10. P. Návrat, M. Bieliková, and V. Rozinajová. Methods and tools for acquiring and presenting information and knowledge in the web. In *Int. Conf. on Computer Systems and Technologies, CompSysTech 2005*, Varna, Bulgaria, 2005.
11. G. Nguyen, M. Laclavík, Z. Balogh, E. Gatial, M. Ciglan, M. Babík, I. Budínska, and L. Hluchý. Data and knowledge acquisition with ontology background. In W. Abramowicz, editor, *Business Information Systems*, Poznan, Poland, 2006.
12. H.S. Pinto and J.P. Martins. Ontologies: How can they be built? *Knowledge and Information Systems*, 6(4):441–464, 2004.
13. P. Sýkora, A. Janžo, P. Kasan, M. Jemala, I. Berta, and V. Szöcs. Automatized Information Retrieval from Heterogenous Web Sources. In M. Bieliková, editor, *Proc. of IIT.SRC 2006*, pages 137–144, Bratislava, Slovakia, 2006.
14. M. Tvarožek. Personalized navigation in the semantic web. In V. Wade et al., editor, *4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 467–471, Dublin, Ireland, 2006. Springer, LNCS 4018.