IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB12876

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Semantic Web-Enabled Web Engineering: The Case of Patterns

Pankaj Kamthan & Hsueh-Ieng Pai
Dept of Computer Science & Software Engineering, Concordia University, Montreal, Quebec, Canada H3G 1M8,
T: (514) 848-2424-3000, F: (514) 848-2424-2830, {kamthan, hsueh_pa}@cse.concordia.ca

ABSTRACT

Patterns are abstract entities of knowledge that are used throughout the lifecycle of a Web Application. In this paper, we give the motivation, prospects, and challenges of representing patterns within the Semantic Web framework. In doing so, we describe our experience of developing an ontology for Web Application patterns, OWAP. Examples of inferences and potential obstacles to ontological representation of patterns are presented.

1. INTRODUCTION AND MOTIVATION

The Semantic Web has recently emerged as an extension of the current Web that adds technological infrastructure for better knowledge representation, interpretation, and reasoning [3]. In order for the Web Applications to make a transition to the Semantic Web, the Web Engineering body of knowledge needs to adapt accordingly. In doing so, we must also preserve the successes of past and present practices in engineering these applications.

Patterns are distilled forms of such reusable knowledge gained from past experience and expertise in solving recurring problems at all levels of development [9]. In the past decade, patterns have been discovered in a variety of domains, including Web Applications [10], where they continue to serve as instruments of guidance and reference to Web engineers.

The Semantic Web can serve as a suitable vehicle for representation and communication of patterns [4]. In fact, ontologies provide a suitable avenue for representing the *knowledge* inherent in a collection of patterns [5]. Since patterns organized as an informal taxonomy provide limited possibilities for automated reasoning, some level of formalization is necessary. The Web Ontology Language (OWL) [1] is the successor of a number of initiatives for ontology specification languages for the Semantic Web with varying degrees of formality and target user communities. Motivated by these factors led to OWAP [6], an OWL ontology for typical structural patterns in a Web Application. In this paper, we outline some prospects and concerns that originate from our experience with the development of OWAP.

The rest of the paper is organized as follows. In Section 2, we give a brief summary of OWAP and highlight of some of its possible uses. Section 3 discusses the challenges of ontological representation of patterns from different perspectives. Finally, Section 4 presents concluding remarks.

2. OVERVIEW OF AN ONTOLOGY FOR WEB APPLICATION PATTERNS AND ITS POTENTIAL USES

The OWAP engineering process was human-centric, iterative, and involved the phases of planning, analysis, design, implementation, and testing [6]. Due to considerations of space, we only give a brief synopsis of this work.

The conceptual model of OWAP is based on the following decomposition scheme that encourages high cohesion and low coupling. The Web Application patterns are divided into two categories. The first category consists of patterns that describe the possible components that a Web Application can *physically* be composed of. For example, a menu is considered as a physical component. Now, a Web Application will normally not consist of all or an arbitrary combination of these patterns. Therefore, the second category consists of patterns that describe how the patterns can be organized *logically* so that a Web Application can be formed using patterns that make sense. For example, a Search Page would normally be composed of at least one text field and a start button to initiate the search. These categories are modeled as class hierarchies in OWL.

Each of the patterns has their unique defining properties that distinguish them from one another. As a collective, patterns are also related to each other in some manner (such as inheritance, composition, and so forth). These aspects are modeled using object and datatype properties in OWL.

For the sake of illustration, we elaborate on the MenuComponent, a nontrivial concept in OWAP. The MenuComponent groups together concepts related to menu. In particular, MenuItem is a MenuComponent, and a Menu is a MenuItem. This is shown in Figure 1 using the Unified Modeling Language (UML) Class Diagram for visual compactness.

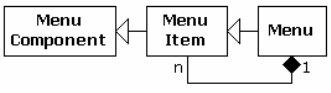
The reason to design Menu as a subclass of MenuItem is to enable complex menu patterns such as the well-known Fly-Out Menu.

OWAP was implemented using OWL DL, one of the three sub-languages of OWL, which provides the right balance for ontological representation of patterns from its foundations in well-understood declarative semantics and available tool support. This allowed us to organize and process patterns in an "intelligent" manner. It also enabled us to make "interesting" inferences, including derivation of facts not literally present in the ontology but entailed by the semantics, and answer certain commonly asked questions. We consider two such cases here.

Example 1. A Hot List is a term commonly referred to a compilation of Web Sites that are considered as favorite or important by a user. Looking at the known uses of a pattern can help a Web engineer appreciate the potential of that pattern. This motivates the question "*What are the known uses of the Hot List pattern?*", the answer to which could be inferred from OWAP using an OWL DL reasoner.

Note that since Hot List is a pattern, it must have at least three distinct known uses (by the "Rule of Three"). This fact can be modeled in OWL DL by defining cardinality constraints. However, there is no systematic way in HyperText Markup Language (HTML), Extensible Markup Language (XML), or in a usual database, to represent such a constraint.

Figure 1. Menu Component Design



Example 2. Given a specific pattern, an engineer may want to find out other patterns that are related to it in some way. This could be helpful to the engineer in estimating the cost of using a pattern, and consequently that of viability of use. This inspires the question "*What patterns is the Travel Web Site pattern composed of?*", the answer to which could again be inferred from OWAP using an OWL DL reasoner.

We once again note the advantages of an ontological representation. Even though patterns in a catalog can be related to one another via linking mechanism in say HTML or XML, these links are structural constructs based on author discretion that do not carry any special semantics. Therefore, their correctness can not be automatically and rigorously verified. For example, even though a pattern does not use itself (as pattern relationships are non-reflexive), it is (mistakenly) possible to link a pattern to itself. Such errors can be avoided in an ontological representation. An ontology can also make hidden dependencies explicit. For example, since the isComposedOf relationship between patterns is defined as a transitive property in OWAP, if a pattern P_1 isComposedOf pattern P_2, and the pattern P_2 isComposedOf the pattern P_3, an OWL DL reasoner can automatically infer that the pattern P_1 isComposedOf the pattern P_3. The inferred results can be further refined using other properties of concepts/relationships.

3. CHALLENGES IN ONTOLOGICAL REPRESENTATION OF PATTERNS

The virtual nature of patterns poses unique knowledge representational challenges as compared to other, more tangible, domains. The experience with OWAP exposed issues in the ontology specification language used, and the authoring and testing tools deployed.

OWAP presents an opportunity to stretch the boundary of OWL to the limit. We demonstrate this using two short examples.

Example 3. In Web Application patterns, it is not uncommon to find spatial/temporal or optional relationships between concepts. However, the current definition of OWL DL (and even OWL in general) does not provide support for representing such information. So, for example, even though one can precisely express the items of a menu, it is difficult to specify in an ontology that, say, the ContactUs item is to the "right" of the Search item, that Help is the "rightmost" item, or that all these items are "close" to each other. Similarly, it is difficult to specify in an ontology that a Web Page instance is linked to another Web Page only "occasionally."

Example 4. Since radio buttons are about making a selection from a given set of choices, it makes sense to express the restriction that a RadioButtonGroup is composed of at least two radio buttons. However, imposing such a cardinality constraint in the ontology is only possible if we go beyond OWL DL (to OWL Full). This in turn leads to loss of computational guarantees (and brings on the risk of undecidability).

The success and quality of an ontology intimately depend on the tools used in its realization. The primary environment for authoring OWAP was Protégé-2000 [7], and it was used for basic syntactic checking, for consistency checking, and for viewing the inferred class hierarchy of the ontology. However, we also had to overcome certain idiosyncrasies that it presented: it was at times resistant to certain necessary modifications in the ontology and did not preserve the structure of markup of an existing file. The reasoner used to derive inferences from OWAP (such as in Examples 1 and 2) was Racer [2], which provides a complete and fairly stable support for OWL DL. It was able to answer most of the queries within a few seconds. However, as the number of instances in OWAP becomes large, a response for a query that involves the use of a transitive property (such as in Example 2) can take a few hours. This is prohibitive in a decentralized environment such as the Semantic Web.

4. CONCLUSION AND FUTURE WORK

For Web Engineering to benefit from the potential of the Semantic Web, its entities of knowledge such as patterns need to adjust accordingly. Indeed, ontologies provide the critical semantic layer in the Semantic Web-based framework for representation of patterns [4]. Ontological representation based on a sound logical basis can make implicit knowledge inherent in patterns explicit. The underlying choice and use of ontology specification language, authoring tool, and reasoner, *collectively* play an important role towards the success of an ontology project. We hope that our experience with OWAP will serve as guide to those who plan for a similar undertaking.

There are a few research directions that emanate from this work. It would be useful to investigate possible extensions of OWL DL to model non-structural relationships among patterns while still preserving the decidability of the ontology. As the number of ontologies grows, the need for a *systematic* approach to the quality assurance and evaluation of OWL ontologies arises. The alignment of ontologies for patterns with ontological efforts in hypermedia and Web design methodologies [8] would also be of interest. We intend to pursue these avenues in the future.

REFERENCES

1. Dean, M., Schreiber, G. (2004). OWL Web Ontology Language Reference. W3C Recommendation. World Wide Web Consortium (W3C).
2. Haarslev, V., Möller, R. (2001). Description of the Racer System and its Applications. Proceedings of the 2001 International Workshop on Description Logics (DL 2001), Stanford, USA, August 1-3, 2001, 132-141.
3. Hendler, J., Lassila, O., Berners-Lee, T. (2001). The Semantic Web. Scientific American, 284(5), 34-43.
4. Kamthan, P. (2005). A Framework for Representation of Software Patterns. Proceedings of the IBIMA 2005 Conference on Theory and Practice of Software Engineering for the 21st Century (TPSE 2005), Cairo, Egypt, December 13-15, 2005.
5. Kamthan, P., Pai, H.-I. (2006). Knowledge Representation in Pattern Management. In: Encyclopedia of Knowledge Management. Schwartz, D. (Editor). Idea Group Publishing.
6. Kamthan, P., Pai, H.-I. (2006). Representation of Web Application Patterns in OWL. In: Web Semantics and Ontology. Taniar, D., Rahayu, J. W. (Editors). Idea Group Publishing.
7. Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., Musen, M. A. (2001). Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems, 16(2), 2001, 60-71.
8. Plessers, P., De Troyer, O. (2004). Web Design for the Semantic Web. Workshop on Application Design, Development and Implementation Issues in the Semantic Web, New York, USA, May 18, 2004.
9. Rising, L. (1999). Patterns: A Way to Reuse Expertise. IEEE Communications Magazine, 37(4), 34-36.
10. Van Duyne, D. K., Landay, J., Hong, J. I. (2003). The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley.