Adaptive portal framework for Semantic Web applications

Michal Barla, Peter Bartalos, Mária Bieliková,
Roman Filkorn, Michal Tvarožek

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

Abstract. In this paper we propose a framework for the creation of adaptive portal solutions for the Semantic Web. It supports different target domains in a single portal instance. We propose a platform environment where the ontology models and adaptivity are among first-class features. Adaptivity is supported by the personalized presentation layer that integrates software tools for automatic user characteristic acquisition. A significant contribution of the design lies in our method for automatic form building from the domain ontology and automated CRUD pattern support by object-ontology mapping. We evaluate the framework in two domains – online labor market and scientific publications.

1 Introduction

Many current information systems need a suitable way of communicating with users by means of a user-friendly (graphical) user interface. Consequently, many systems adopted a web-based user interface, which can be accessed via a thin client, such as a generic web browser. This introduces new challenges since the architecture, design and overall approach to engineering of web-based applications differs from traditional desktop thick client applications.

While typical web applications offer specific services to users, web portal solutions aim to provide a single point of access for personalized services, information sharing and collaboration support. Furthermore, portals serve as gateways to other content and services provided either locally or more often as distributed applications. Thus, system integration plays a very important role where interoperability is becoming paramount. Unlike traditional desktop applications, web portals often employ a diverse range of middleware, specialized methods and tools to integrate and process information from various sources. Consequently, portal solutions strive not only for maximal flexibility and variability but also for shared semantics to which the Semantic Web principles may be applicable [10]. These however are not yet supported by the state of the art portal frameworks.

Web-based information systems in general, and portal systems in particular can also be viewed from the client users' perspective where the overall design, functionality and user-friendliness of the user interface is important. Personalized

approaches such as adaptive hypermedia have been proposed to solve common problems like the "lost in hyperspace" syndrome and information overload, while social approaches were proposed for collaboration and information sharing.

In this paper we propose a framework for the creation of adaptive web-based portal solutions for integrating different web applications. We strongly focus on reusability, component-based design, personalization and interoperability taking advantage of ontologies, adaptive navigation and presentation.

2 Related work

There are many commercial quality portal solution products from a variety of top rank software vendors, e.g. Microsoft SharePoint, Sun Java System Portal Server, IBM WebSphere or BEA WebLogic. Although definitely varying in specific technology particulars, the list of supported features and off-the-shelf components is overwhelming. They offer consistent solutions and share many characteristics such as security, enterprise information and services integration, documentation, steep learning curve, ease and comfort use and administration. Similarly, open-source Apache Foundation projects such as Cocoon, Struts, Tapestry or Jetspeed are examples of commonly used, technically mature, reusable portal, albeit less sophisticated, frameworks for fast web application development.

Some state of the art methods in web application development, based on model drive approaches, include HERA [14], WebML [8], SHDM/OOHDM [9], UWE [5]. Ideally, these are aimed at designing web applications, which are well understood and where the respective models can be (easily) defined. However, they do not directly address the integration and common aspects of different distributed web applications and/or data sources into a single portal instance.

The idea of using ontologies in portal solutions for the Semantic Web has already been examined in several works. OntoPortal uses ontologies and adaptive hypermedia principles to enrich the linking between resources [4]. The AKT project aims to develop technologies for knowledge processing, management and publishing, e.g. OntoWeaver-S [6], which is a comprehensive ontology-based infrastructure for building knowledge portals with support for web services.

The SEAL [11] framework for semantic portals takes advantage of semantics for the presentation of information in a portal with focus on semantic querying and browsing. The semantic web portal tool OntoViews [7] is designed for publishing of RDF content on the web and provides the user with a content-based search engine and link generation/recommendation based on relationships between ontological concepts. SOIP-F [13] describes a framework for the development of semantic organization information portals based on "conventional" web frameworks, web conceptual models, ontologies as well as additional metadata.

A lot of work has already been done in the field of semantic web portals. Existing approaches take extensive advantage of ontologies, web services and different navigation and presentation models. However, while support for personalization (via presentation adaptation to user context) was already addressed in some approaches, they do not offer fully automatic semantic user action logging. Our

approach takes advantage of semantic server-side logging which supports and augments the successive user characteristic estimation.

Furthermore, issues concerning the evolution of open information spaces should be addressed with respect to effective portal development and maintenance with the aim of reducing workload when developing new portal solutions or maintaining existing ones in changing environments. Our automated support of CRUD patterns contributes to this issue.

3 Adaptive portal solution architecture

The proposed framework for the creation of adaptive web-based portal solutions has two major goals: to be able to support different target domains in a single portal instance, and to set up a platform environment where the ontology models and adaptivity will be among first-class features. A portal created using the framework stands as an integration platform for different existing or newly developed web applications, which are available via a single access point, and which can be either independent or interconnected.

A target domain is represented by a domain specific model. It captures and specifies its characteristic concepts, structures, relations, behavior and constraints. In order to easily change models one has to focus on a meta-model. Ontologies constitute a way to manage and process the model and its meta-model in a consistent and uniform way. While being able to manipulate the entities at the instance level of the ontology, the inference mechanisms may take both levels in consideration and the result may improve and alter either the model or the meta-model of the particular target domain.

From this point of view, even two consecutive versions of the same ontology may be considered as two different models and a suite of tools and inference rules may be able to process the data between these two instances. In such a way our framework is able to adapt to changes to its own (meta-)model.

Our design reflects the following requirements:

- Adaptivity and adaptability of the system's presentation and functionality.
- Built-in automatic user modeling based on user action logging with semantics and automatic user characteristic estimation.
- Reusability and generic design suitable for multiple application domains.
- Extensibility with additional tools for specific tasks and overall flexibility with respect to tool orchestration.
- Tolerance towards changes in the domain and user ontologies.

In our design we take advantage of MVC-based frameworks, component-based web development and XML processing, which are based on the pipes and filters architectural pattern, what makes them specifically suitable for RDF/RDFS and OWL processing. One such framework is the open-source web development framework Apache Cocoon (`http://cocoon.apache.org/`), which we used as the underlying portal framework for our solution. Figure 1 depicts an overview

of the portal architecture that extends the basic functionality of Cocoon with additional software components in order to fulfill the aforementioned requirements.

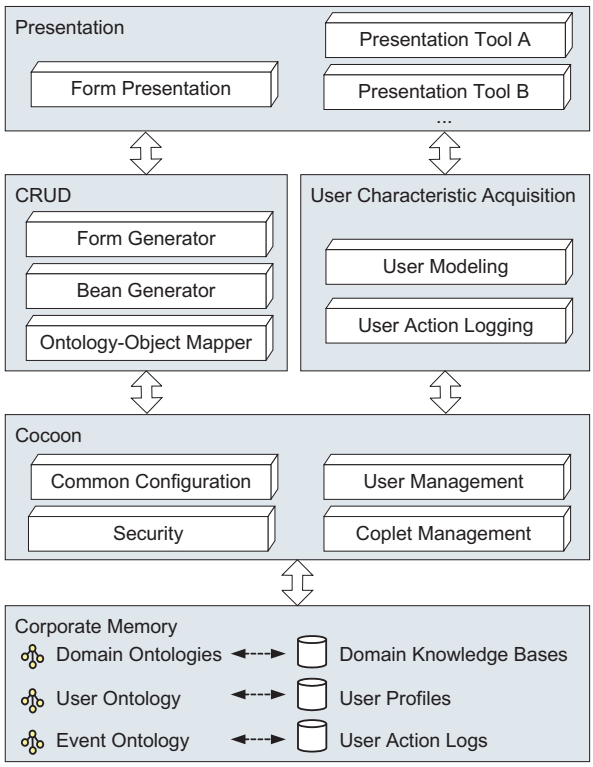

Fig. 1. Overview of our adaptive semantic portal framework architecture.

Corporate memory. We store data in the *Corporate Memory* repository which stores the domain, user and event ontologies (Figure 1, bottom). We use a domain ontology to capture and formally specify domain specific data – concepts, structures, relations, behavior and constraints characteristic for a particular application domain. A user ontology is derived from the domain ontology to define users, their characteristics and preferences towards specific domain concepts. We employ an event ontology to capture the semantics of user actions during system operation for their successive processing in the user modeling process.

Cocoon extensions. The core Cocoon extensions include (Figure 1, center):

– *User Management* used for creating and altering of user accounts.

– *Common Configuration* of individual tools, which is used to access data in the *Corporate Memory* repository.
– *Security*, which ensures that only authorized users access protected resources.
– *Coplet Management* used to customize the overall portal interface, i.e. to add, remove or edit the layout and use of individual coplets (i.e., Cocoon servlets corresponding to specific GUI parts) and skins.

CRUD support. A significant contribution of our design is the *CRUD* component. It supports form generation from the domain ontology and the automated *CRUD* pattern (Figure 1, top left) as means of improving reusability for different application domains. *CRUD* organizes the persistence operations of an application into Create, Retrieve, Update and Delete operations that are implemented by a persistence layer, and includes the generation of form descriptions for Cocoon (*Form Generator*), the generation of the underlying JavaBeans (*Bean Generator*) and the associated mapping and persistence of JavaBeans (*Ontology-Object Mapper*) in the ontological repository [2].

User characteristics acquisition. We employ the personalized presentation layer architecture proposed in [12] that facilitates *User Characteristic Acquisition* – a two stage process consisting of sever-side and client-side *User Action Logging* and *User Modeling*. The process takes advantage of a set of software tools, integrated into the portal framework, that form a configurable user modeling chain which transforms user actions into a user model that can be used by all tools integrated in the portal.

The *User Action Logging* stage produces logs with semantics which are processed using a rule-based approach [1] resulting in user characteristics stored in an ontology-based user model. Every presentation tool in the portal is responsible for the logging of its respective events and their semantics by means of a common logging service and thus contributes to user characteristics acquisition process.

Presentation. The *Portal* tool is used to aggregate output from individual adaptive *Presentation tools*, which support adaptation based on user context, and assist in the creation of comprehensive user action logs. The user context itself contains different types of data, e.g., a user model describing user characteristics and an environment model describing the client device or client connection.

In particular, we utilize *Form Presentation* tools that take advantage of CRUD pattern support to provide users with personalized form filling functionality for specific domain concepts. For navigation in the domain ontology we use an adaptive faceted browser and a cluster navigation tool that supports visual navigation in clusters of domain concepts. We also employ several search tools that allow the user to specify different search criteria and ranking algorithms.

4 Evaluation

We successfully employed the proposed portal solution in two projects dealing with different domains. Using the framework we created a portal *Job Offer Portal* (JOP) used in research project NAZOU (`http://nazou.fiit.stuba.sk`) in the domain of online job offers. JOP offers its users several ways of navigation through the information space using different presentation tools, which work with the ontological database produced by a chain of data harvesting tools that acquire and process data from the Internet [3].

The whole system utilizes multiple data processing chains. Starting with the data sources, users can submit new job offers using a set of forms generated by the framework. On the other hand, a set of automatic wrapping and web crawling tools collects (structured) documents (tools WrapperGenerator, WebCrawler, RIDAR – Relevant Internet Data Resource Identification). To support information retrieval, approaches like clustering (tools Clusterer, ASPECT – Probabilistic document clustering) and criteria and top-k ordering (tools CriteriaSearch, SQEx – Semantic Query Expansion Tool, TopK aggregator, IGAP – Induction of Generalized Annotated Programs) are employed. The data and search results presentation is performed by JOP – the primary adaptable user interface, which integrates individual presentation and user modeling tools (tools Factic - Faceted browser and ClusterNavigator) and user modeling (tools Click, LogAnalyser, SemanticLog).

Another portal, called *Publication Presentation Portal* (P3) was created in research project MAPEKUS (`http://mapekus.fiit.stuba.sk`). It uses metadata about scientific publications downloaded from digital libraries and aids users in finding relevant ones by adapting the presented information.

Both created portals use ontology-based back-end and user modeling features provided by our portal framework. Both stand for an integration platform for various domain-specific tools and data processing workflows. The features of common ontologies and adaptivity significantly improve their overall quality.

5 Conclusion

We described the design of a framework for the creation of adaptive web-based portal solutions with support for both adaptability and adaptivity.

We take advantage of component-based design and built a working portal from a set of interconnected software tools that perform specific tasks. Furthermore, we employ ontologies in order to incorporate semantics shared across individual tools, data and metadata into the respective domain and user models in a consistent and uniform way. In this way our solution supports different target domains in single portal instance.

The automated form generation from the domain ontology and object-ontology mapping contributes to the flexibility and the easy reuse of the solution. Using these components we can flexibly react to domain ontology changes by changing the corresponding parts of the application automatically.

Acknowledgment. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-007104 and the State programme of research and development under the contract No. 1025/04.

References

1. M. Barla and M. Bieliková. Estimation of User Characteristics using Rule-based Analysis of User Logs. In *Data Mining for User Modeling Proceedings of Workshop held at the International Conference on User Modeling UM2007*, pages 5–14, Corfu, Greece, 2007.
2. P. Bartalos. An approach to object-ontology mapping. In Mária Bieliková, editor, *IIT.SRC – Student Research Conference 2007*, pages 9–16. Slovak University of Technology, Bratislava, Slovakia, 2007.
3. M. Ciglan, M. Babik, M. Laclavik, I. Budinska, and L. Hluchy. Corporate memory: A framework for supporting tools for acquisition, organization and maintenance of information and knowledge. In J. Zendulka, editor, *9th Int. Conf. on Inf. Systems Implementation and Modelling, ISIM'06*, pages 185–192, Perov, Czech Rep., 2006.
4. S. Kampa, T. Miles-Board, L. Carr, and W. Hall. Linking with meaning: Ontological hypertext for scholars, 2001.
5. N. Koch. Transformation techniques in the model-driven development process of uwe. In *ICWE '06: Workshop proceedings of the sixth international conference on Web engineering*, page 3, New York, NY, USA, 2006. ACM Press.
6. Y. Lei, E. Motta, and J. Domingue. Ontoweaver-s: Supporting the design of knowledge portals. In E. Motta et al., editor, *EKAW*, volume 3257 of *LNCS*, pages 216–230. Springer, 2004.
7. E. Mäkelä, E. Hyvönen, S. Saarela, and K. Viljanen. Ontoviews - a tool for creating semantic web portals. In S. A. McIlraith et al., editor, *Int. Semantic Web Conf.*, volume 3298 of *LNCS*, pages 797–811. Springer, 2004.
8. N. Moreno, P. Fraternalli, and A. Vallecillo. A uml 2.0 profile for webml modeling. In *ICWE '06: Workshop proceedings of the 6th international conference on Web engineering*, page 4, New York, NY, USA, 2006. ACM Press.
9. L. A. Ricci and D. Schwabe. An authoring environment for model-driven web applications. In *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*, pages 11–19, New York, NY, USA, 2006. ACM Press.
10. N. Shadbolt, T. Berners-Lee, and W. Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, May/June 2006.
11. N. Stojanovic, A. Maedche, S. Staab, R. Studer, and Y. Sure. Seal: a framework for developing semantic portals, 2001.
12. M. Tvarožek, M. Barla, and M. Bieliková. Personalized Presentation in Web-Based Information Systems. In J. van Leeuwen et al., editor, *SOFSEM 2007*, pages 796–807. Springer, LNCS 4362, 2007.
13. E. D. Valle and M. Brioschi. Toward a framework for semantic organizational information portal. In Ch. Bussler et al., editor, *ESWS*, volume 3053 of *LNCS*, pages 402–416. Springer, 2004.
14. K. van der Sluijs and G.J. Houben. A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning*, 16(1/2):64–76, 2006.