Journal of Universal Computer Science, vol. 8, no. 6 (2002), 662-672
submitted: 4/2/02, accepted: 3/5/02, appeared: 28/6/02

Managing User Focused Access to Distributed Knowledge

Rudi Studer
(Institute AIFB, University of Karlsruhe, Germany
FZI Research Center for Information Technologies, Germany
Ontoprise GmbH, Germany
studer@aifb.uni-karlsruhe.de)

York Sure
(Institute AIFB, University of Karlsruhe, Germany
sure@aifb.uni-karlsruhe.de)

Raphael Volz
(Institute AIFB, University of Karlsruhe, Germany
FZI Research Center for Information Technologies, Germany
volz@aifb.uni-karlsruhe.de)

Abstract: Community web sites exhibit the property that multiple content providers exist. Of course, any portal is only as useful as the quality and amount of its content. Developing original content is time consuming and expensive. To offset the cost, we present a novel framework, viz. SEAL (SEmantic portAL), that builds on Semantic Web standards. We illustrate our approach with examples from the OntoWeb community portal. Community web sites exhibit two dominating properties: They often need to integrate many different information sources and they require an adequate web site management system. SEAL exploits ontologies for fulfilling the requirements set forth by these two properties. Ontologies provide a high level of sophistication for web information integration as well as for web site management.

Key Words: knowledge portal, information integration, ontology

Category: H.0

1 Introduction

Supporting communities in sharing and exchanging knowledge is an important aspect of Knowledge Management. This holds e.g for communities of practice being organized within enterprises or being organized by a collection of cooperating enterprises or for scientific communities that are spread all over the world and thus urgently need support in sharing their knowledge. In that context, knowledge portals [14] play a part in offering means for providing and accessing knowledge on a semantic level. In essence, knowledge portals exploit ontologies for achieving a conceptual foundation for all functionalities that are offered by the portal. We have developed the SEAL (SEmantic PortAL) framework for developing and managing knowledge portals. SEAL exploits Semantic Web technologies to offer mechanisms for acquiring, structuring, integrating, sharing and accessing distributed knowledge between human and/or machine agents [10, 8]. Up-to-now, SEAL put emphasis on supporting the acquiring and structuring of knowledge by semantic annotation [7] and the automatic generation of navigational views and mixed ontology and content-based presentation.

The topic of this paper is the application and extension of SEAL for realizing the OntoWeb community portal (http://www.ontoweb.org). OntoWeb is an EU IST thematic network that propagates ontologies in the context of eBusiness and Knowledge Management and that currently has more than one hundred members from research and industry. The OntoWeb knowledge portal will be used as a case study throughout the paper. In the process of setting up the OntoWeb portal we recognized rather soon that the process of knowledge provisioning and publishing has to be supported by an appropriate workflow in order to be able to control what content is put into the portal by whom. Only then, the high quality of content that is expected by the OntoWeb users can be guaranteed. Therefore, the SEAL framework has been extended by methods and tools for defining and handling a publishing workflow. Such a workflow represents an important constituent of the overall approach for managing a running knowledge portal to make user focussed access to the OntoWeb portal maintainable.

The paper is structured as follows. First, we describe the main components and functionalities of our SEAL framework. In Section 3 we outline the scenario that is set up by the OntoWeb portal and derive from that scenario the new requirements that have to be met by the SEAL framework, especially the publishing workflow. The definition of the publishing workflow and its realization as part of the SEAL framework are described in Section 4. We conclude with a discussion of related work and an outline of open research problems.

2 SEAL — The core approach

The recent decade has seen a tremendous progress in managing semantically heterogeneous data sources. Core to the semantic reconcilation between the different sources is a rich conceptual model that the various stakeholders agree on, an *ontology* [4]. The conceptual architecture developed for this purpose now generally consists of a three layer architecture comprising (cf. [15]) (i) heterogeneous **data sources** (e.g., databases, XML, but also data found in HTML tables), (ii) **wrappers** that lift these data sources onto a common data model (e.g. OEM [12] or RDF [9]), (iii) integration modules (**mediators** in the dynamic case) that reconcile the varying semantics of the different data sources. Thus, the complexity of the integration/mediation task could be greatly reduced.

Similarly, in recent years the information system community has successfully strived to reduce the effort for managing complex web sites [1, 2, 5, 11]). Previously ill-structured web site management has been structured with process models, redundancy of data has been avoided by generating it from database systems and web site generation (including management, authoring, business logic and design) has profited from recent, also commercially viable, successes [1]. Again we may recognize that core to these different web site management approaches is a rich conceptual model that allows for accurate and flexible access to data. Similarly, in the hypertext community conceptual models have been explored that im- or explicitly exploit ontologies as underlying structures for hypertext generation and use (e.g. [3]).

SEAL (SEmantic PortAL)[1], our framework to building community web sites, has been developed to use ontologies as key elements for managing community web sites and web portals. The ontology supports queries to multiple sources (a task also supported by semi-structured data models [5]), but beyond that it also includes the intensive use of the schema information itself allowing for automatic generation of navigational views[2] and mixed ontology and content-based presentation. The core idea of SEAL is that Semantic Portals for a community of users that contribute *and* consume information [13] require web site management *and* web information integration. In order to reduce engineering and maintenance efforts SEAL uses an ontology for semantic integration of existing data sources as well as for web site management and presentation to the outside world. SEAL exploits the ontology to offer mechanisms for acquiring, structuring and sharing information between human and/or machine agents. Thus, SEAL combines the advantages of the two worlds briefly sketched above.

3 OntoWeb Scenario

The EU thematic network "OntoWeb – Ontology-based information exchange for knowledge management and electronic commerce" aims at bringing together researcher and industrials to "enable the full power ontologies may have to improve information exchange in areas such as: information retrieval, knowledge management, electronic commerce, and bioinformatics. It will also strengthen the European influence on standardization efforts in areas such as web languages (RDF, XML), upper-layer ontologies, and content standards such as catalogues in electronic commerce" (cf. [16]). One of the tasks of the OntoWeb partners is to create a portal for this community serving as a platform for communication between partners and also between partners and other members of the Word Wide Web.

Portal approach. The OntoWeb portal (cf. Figure 1) is structured according to an ontology which serves as a shared basis for supporting communication between humans and machines. The general goal of our approach is the semi-automatical construction of a community portal using the community's metadata

[1] Cf. [10] on the history of SEAL.

[2] Examples are navigation hierarchies that appear as `has-part`-trees or `has-subtopic` trees in the ontology.

Figure 1: www.ontoweb.org – The OntoWeb portal

to enable information provision, querying and browsing of the portal. For this purpose we could reuse the framework as explained in Section 2, but we also had to provide new modules for content management resulting in the extended architecture depicted in Figures 2 and 3. The use of core SEAL modules is explained in the following, new ones follow subsequently. The process model is introduced in Section 4.

3.1 Use of core SEAL modules

Integration. One of the core challenges when building a data-intensive web site is the integration of heterogeneous information on the WWW. The recent decade has seen a tremendous progress in managing semantically heterogeneous data sources [15, 5]. The general approach we pursue is to "lift" all the different input sources onto a common data model, in our case RDF. Additionally, an ontology acts as a semantic model for the heterogeneous input sources. As mentioned earlier and visualized in our conceptual architecture in Figure 3, we consider different kinds of **Web data sources** as input. However, to a large part the Web consists of static HTML pages, often semi-structured, including tables,

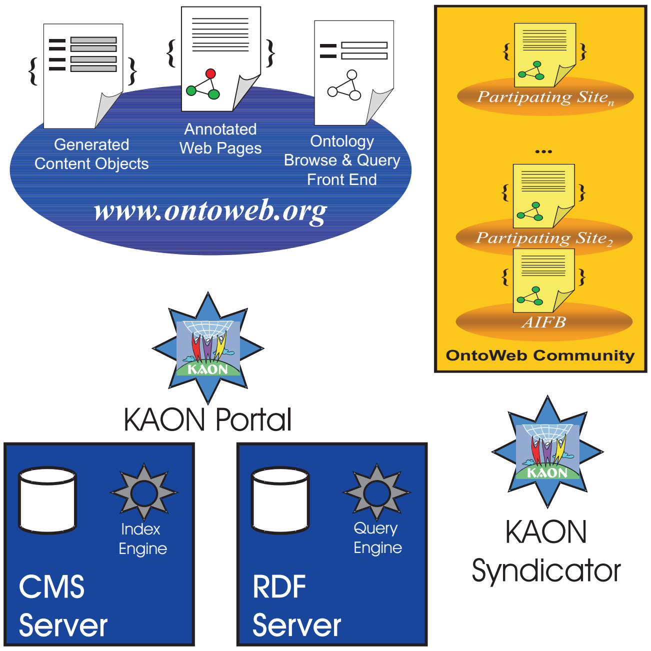

Figure 2: OntoWeb architecture

lists, etc..

Presentation. Based on the integrated data in the warehouse we define user-dependent **presentation views**. First, we render HTML pages for human agents. Typically *queries for content* of the warehouse define presentation views by selecting content, but also *queries for schema* might be used, e.g. to label table headers. Second, as a contribution to the Semantic Web, our architecture is dedicated to satisfy the needs of software agents and produces machine understandable RDF. To maintain a portal and keep it alive its content needs to be updated frequently not only by information integration of different sources but also by additional inputs from human experts. The **input view** is defined by *queries to the schema*, i.e. queries to the ontology itself. Similar to [6] we support the knowledge acquisition task by generating forms out of the ontology. The forms capture data according to the ontology in a consistent way which are stored afterwards in the warehouse. To navigate and browse the warehouse we automatically generate navigational structures, i.e. **navigation views**, by using *combined queries for schema and content*. First, we offer different user views on the ontology by using different types of hierarchies (e.g. *is-a*, *part-of*) for the

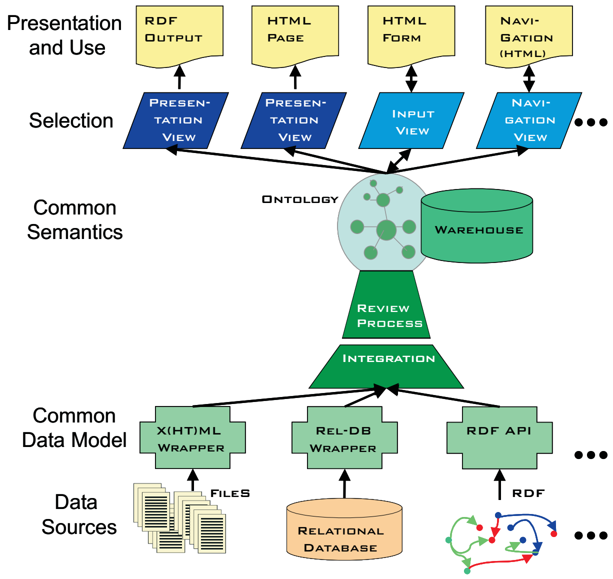

Figure 3: Extended conceptual SEAL architecture

creation of top level navigational structures. Second, for each shown part of the ontology the corresponding content in the warehouse is presented. For non-typed content such as documents we take several heuristics to offer navigation: First, all other objects that have the same physical location (folder on the web server) are assumed to be related, as the user put it at that exact location for a certain reason. Second, we use the metadata of the document to find similar objects using the objects' metadata, e.g. objects having the same subject, keywords or author. This provides a simpler way of exploring the content for users that are unfamiliar with the portal.

3.2 Implementation

In a nutshell, the upper two levels in the conceptual architecture of SEAL (cf. Figure 3) are implemented as KAON Portal (cf. Figure 2). It generates content objects and provides browsing as well as a query frontend. The replication of distributed knowledge into the RDF Server is done by the KAON Syndicator. Please note that only structured data is replicated and not, e.g., documents. The storage consists of (i) a content management system that allows for creation and management of documents (but not annotations), (ii) the RDF management system that stores ontologies and associated instance base annotations of the content management system. The OntoWeb Community provides metadata on their web sites which are syndicated with the KAON Syndicator. The workflow component described in the next section is provided by the CMF framework[3], an extension of the Zope web application server[4].

4 Process Model

As mention in Section 3 OntoWeb is an open community. Open communities pose additional constraints since data that is (re)published through the portal could be provided by arbitrary people. In order to guarantee quality of data in such an environment an additional model regulating the publishing process is required, which prevents foreseeable misuses. To support this requirement the established SEAL architecture was extended with a workflow component which regulates the publishing process. In the following we will begin with introducing the concept of a publishing workflow in general. Afterwards we explain how we instantiated this generic component in OntoWeb.

4.1 Publishing workflows

A publishing workflow is the series of interactions that should happen to complete the task of publishing data. Business organizations have many kinds of workflow. Our notion of workflow is centered around tasks. Workflows consist of several tasks and several transitions between these tasks. Additionally workflows have the following characteristics: (i) they might involve several people, (ii) they might take a long time, (iii) they vary significantly in organizations and in the computer applications supporting these organizations respectively, (iv) sometimes information must be kept across states, and last but not least, (v) the communication between people must be supported in order to facilitate decision making. Thus, a workflow component must be customizable. It must support the assignment of tasks to (possibly multiple) individual users. In our architecture these users are grouped into roles. Tasks are represented within a workflow as a set of transitions which cause state changes. Each object in the system is assigned a state, which corresponds to the current position within the workflow and can be used to determine the possible transitions that can validly be applied to the object. This state is persistent supporting the second characteristic mentioned above. Due to the individuality of workflows within organizations and applications we propose a generic component that supports the creation and customization of several workflows. In fact, each concept in the ontology, which – as you might recall – is used to capture structured data within a

[3] http://cmf.zope.org/
[4] http://www.zope.org/

portal, can be assigned a different workflow with different states, transitions and task assignments. As mentioned above, sometimes data is required to be kept across states. For example, envision the process of passing bills in legislature, a bill might be allowed to be revised and resubmitted once it is vetoed, but only if it has been vetoed once. If it is vetoed a second time, it is rejected forever. To model this behavior, the state machine underlying our workflow model needs to keep information that "remembers" the past veto. Thus, variables are attached to objects and used to provide persistent information that transcends states. Within our approach variables also serve the purpose of establishing a simple form of communication between the involved parties. Thus, each transition can attach comments to support the decision made by future actors. Also metadata like the time and initiator of a transition is kept within the system.

4.2 Workflows in OntoWeb

Figure 4 depicts the default workflow within OntoWeb. There are three states: private, pending, and published. If a user creates a new object[5] the object is in private state. If the user has either a reviewer or a manager role the published state is immediately available through the publish transition. For normal users such a transition is not available, instead the object can only be send for a review leading to the pending state. In the pending state either managers or reviewers can do the transition to the published state (by applying the transition "publish") or retract the object leading back to the private state. The reject transition deletes the object completely. When an object is in the private state, only the user who created it and users with manager roles can view and change it. Once an object is in published state the modification by the user who created it resets the object into pending state, thus the modification must be reviewed again. This does not apply to modifications by site managers.

5 Related work

Given aforementioned difficulties with managing complex Web content, several papers tried to facilitate database technology to simplify the creation and maintenance of data-intensive web-sites. Systems, such as ARANEUS [11] and AutoWeb [2], also take a declarative approach. In contrast to SEAL that relies on standard Semantic Web technologies these systems introduce their own data models and query languages, although all approaches share the idea to provide high-level descriptions of web-sites by distinct orthogonal dimensions. The idea of leveraging mediation technologies for the acquisition of data is also found in

[5] (currently only within the portal, the content syndicated from other OntoWeb member web sites and within the databases is "trusted". We assume that this kind of data already went through some kind of review.

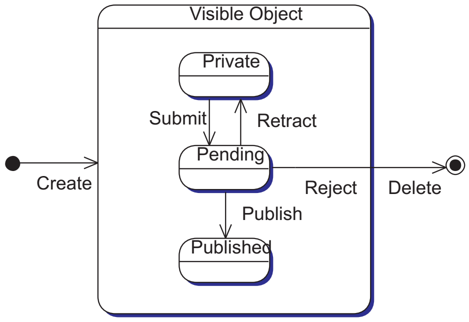

Figure 4: SEAL Publishing workflow

approaches like Strudel [5] and Tiramisu [1], they propose a separation according to the aforementioned task profiles as well. Strudel does not concern the aspects of site maintenance and personalization. It is actually only an implementation tool, not a management system. From our point of view the SEAL framework and it's application as the OntoWeb portal is rather unique with respect to the collection of methods used and the functionality provided.

6 Conclusion

In this paper we have shown the application of our comprehensive framework SEAL for building "SEmantic portALs". In particular, we have focused on three issues. First, we have described the general architecture of the SEAL framework. Second, we have presented our real world case study, the OntoWeb portal. Third, to meet the requirements of the OntoWeb portal, we extended our initial conceptual architecture SEAL by publishing workflows to make user focussed access to the OntoWeb portal maintainable.

For the future, we see a number of new important topics appearing on the horizon. For instance, we consider approaches for ontology learning in order to semi-automatically adapt to changes in the world and to facilitate the engineering of ontologies. Currently, we work on providing intelligent means for providing semantic information, *i.e.* we elaborate on a semantic annotation framework that balances between manual provisioning from legacy texts (*e.g.* web pages) and information extraction. Finally, we envision that once semantic web sites are widely available, their automatic exploitation may be brought to new levels. Semantic web mining considers the level of mining web site structures, web site content, and web site usage on a semantic rather than at a syntactic level yielding new possibilities, *e.g.* for intelligent navigation, personalization, or summarization, to name but a few objectives for semantic web sites.

Acknowledgements

We thank our colleagues at StarLab, VU Brussels headed by of Robert Meersmann and at Institute AIFB, University of Karlsruhe, in particular Daniel Oberle, for fruitful discussions on the work reported here. This work has been funded under the EU IST-2001-29243 project "OntoWeb" and the EU IST-2001-33052 project "WonderWeb".

References

1. C. R. Anderson, A. Y. Levy, and D. S. Weld. Declarative web site management with tiramisu. In *ACM SIGMOD Workshop on the Web and Databases - WebDB99*, pages 19–24, 1999.
2. S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (WebML): a modeling language for designing web sites. In *WWW9 Conference, Amsterdam, May 2000*, 2000.
3. M. Crampes and S. Ranwez. Ontology-supported and ontology-driven conceptual navigation on the world wide web. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, May 30 - June 3, 2000, San Antonio, TX, USA*, pages 191–199. ACM Press, 2000.
4. D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, R. Studer, and A. Witt. Lessons learned from applying AI to the web. *International Journal of Cooperative Information Systems*, 9(4):361–382, 2000.
5. M. F. Fernandez, D. Florescu, A. Y. Levy, and D. Suciu. Declarative specification of web sites with Strudel. *VLDB Journal*, 9(1):38–55, 2000.
6. E. Grosso, H. Eriksson, R. W. Fergerson, S. W. Tu, and M. M. Musen. Knowledge modeling at the millennium: the design and evolution of PROTEGE-2000. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW-99)*, Banff, Canada, October 1999.
7. S. Handschuh and S. Staab. Authoring and annotation of web pages in CREAM. In *The Eleventh International World Wide Web Conference (WWW2002), Honolulu, Hawaii, USA 7-11 May*, 2002. To appear.
8. A. Hotho, A. Maedche, S. Staab, and R. Studer. SEAL-II - The soft spot between richly structured and unstructured knowledge. *Universal Computer Science (J.UCS)*, 7(7):566–590, 2001.
9. O. Lassila and R. Swick. Resource Description Framework (RDF). Model and syntax specification. Technical report, W3C, 1999. http://www.w3.org/TR/REC-rdf-syntax.
10. A. Maedche, S. Staab, R. Studer, Y. Sure, and R. Volz. Seal — tying up information integration and web site management by ontologies. *IEEE Data Engineering Bulletin*, 25(1):10–17, March 2002.
11. G. Mecca, P. Merialdo, P. Atzeni, and V. Crescenzi. The (short) Araneus guide to web-site development. In *Second Intern. Workshop on the Web and Databases (WebDB'99) in conjunction with SIGMOD'99*, May 1999.
12. Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proceedings of the IEEE International Conference on Data Engineering, Taipei, Taiwan, March 1995*, pages 251–260, 1995.

13. S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. In *WWW9 / Computer Networks (Special Issue: WWW9 - Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000)*, volume 33, pages 473–491. Elsevier, 2000.
14. S. Staab and A. Maedche. Knowledge portals - ontologies at work. *AI Magazine*, 21(2), 2001.
15. G. Wiederhold and M. Genesereth. The conceptual basis for mediation services. *IEEE Expert*, 12(5):38–47, Sep.-Oct. 1997.
16. Thematic Network EU IST-2000-25056 OntoWeb: Annex 1 - "Description of Work". Technical report, Information Societies Technology (IST) Programme, February 11 2001.