SEAL — A SEMANTIC PORTAL WITH CONTENT MANAGEMENT FUNCTIONALITY

Marc Ehrig[1], Steffen Staab[1,2], Rudi Studer[1,2,3], York Sure[1], Raphael Volz[1,3]
[1]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
[2]Ontoprise GmbH, Haid-und-Neu Strasse 7, 76131 Karlsruhe, Germany
[3]Research Group Knowledge Management, FZI — Research Center
for Information Technologies, 76131 Karlsruhe, Germany
Telephone: +49 721 608-4750, Fax: +49 721 608-6580
E-mail: {ehrig,staab,studer,sure}@aifb.uni-karlsruhe.de

Abstract

"OntoWeb" is an European Union IST-funded thematic network for "Ontology-based information exchange for knowledge management and electronic commerce". The corresponding OntoWeb portal constitutes a Web-based research information system that is driven by some of the technologies which it reports about.
*In this paper, we present the core methodology underlying the OntoWeb portal,*viz. *SEAL (SEmantic portAL). In particular, we describe some of the core challenges that SEAL must meet. Because of the distributed nature of research information, SEAL has been developed as a methodology that integrates heterogeneous information from distributed resources. Because of the complexity of the application domain, SEAL is based on ontologies about research information that greatly contribute to the combined goals of low-effort information integration and user-friendly information presentation. Because of the high quality requirements obliged onto the OntoWeb portal, SEAL has been extended with content management functionality supporting portal editors in their process to rule out undesirable content.*

Keywords

Content Management, Knowledge Portal

1 Introduction

By its very nature, information about scientific research on the Web tends to be *distributed*, *heterogenous*, *volatile*, *interrelated*, and *focused around topics, persons, projects, and organizations*. There are plenty of structures on the Web that host research information, e.g. conference web sites, homepages of researchers, project web sites and web information providers (e.g. free providers like `http://www.ceur-ws.org` or providers-by-fee like `http://www.acm.org`). Typically, however, these existing structures do not make the context explicit under which their research information may be found. Thus, the provisioning of research information mostly remains ideosyncratic and access to it is at most as good as the information retrieval mechanisms that let you find research information by keyword search.

In order to overcome some of the difficulties associated with accessing research information by keyword search, we have developed an ontology-based approach. An ontology is an explicit specification of shared conceptualizations for a domain of interest. I.e. ontologies make assumptions explicit that a community of people shares about a particular subject. The use of ontologies for information exchange may help to put the many different pieces of research information available into a coherent, re-usable and re-configurable picture. Therefore, the core idea of our SEmantic portAL methodology (SEAL) consists of exploiting ontology structures for specifying the context of particular pieces of research information within one research community — for the purpose of information integration as well as for information presentation.

"OntoWeb" is an European Union IST-funded thematic network that propagates research related to ontology technologies and that, of course, has similar knowledge sharing needs as other research communities. Therefore, we are currently developing the *OntoWeb Portal* as part of the effort to develop ontology technology and nourish ourselves on it, too.

Developing the portal, we have seen the needs for ontology-based integration of information that we have also met when dealing with developing a Web presentation of our institute (just a small research community; cf. [9]). In addition, new challenges showed up calling for the combination of managing a portal for a highly-distributed community (about 100 partners spread all over Europe and beyond) at low costs and high quality. Thus, SEAL has been extended with content management functionality supporting portal editors in their process to rule out undesirable content.

In the following, we will first sketch the core SEAL approach that we had developed before the OntoWeb portal (Section 2). Then, we describe the scenario of the OntoWeb portal and some of its new requirements (Section 3) — including a revision of the existing architecture (compare with [9]). Thereafter, we describe the process model employed in the OntoWeb portal.

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* [5]. The conceptual architecture developed for this purpose now generally consists of a three layer architecture comprising (cf. [13]) (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 [11] or RDF [8]), (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 [2, 3, 6, 10]). 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 [2]. 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. [4]).

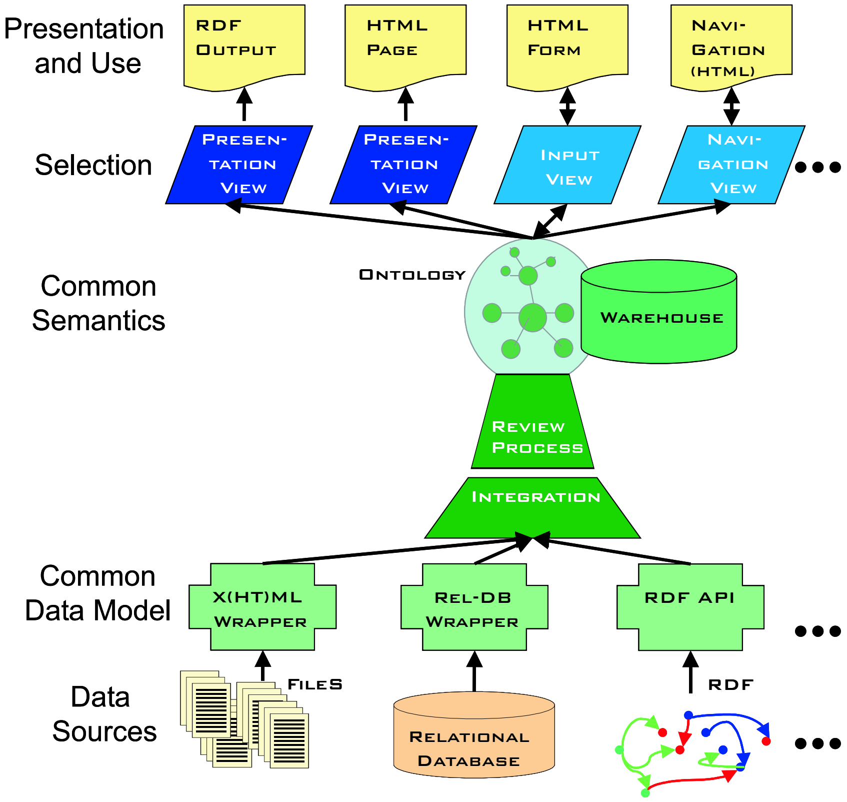


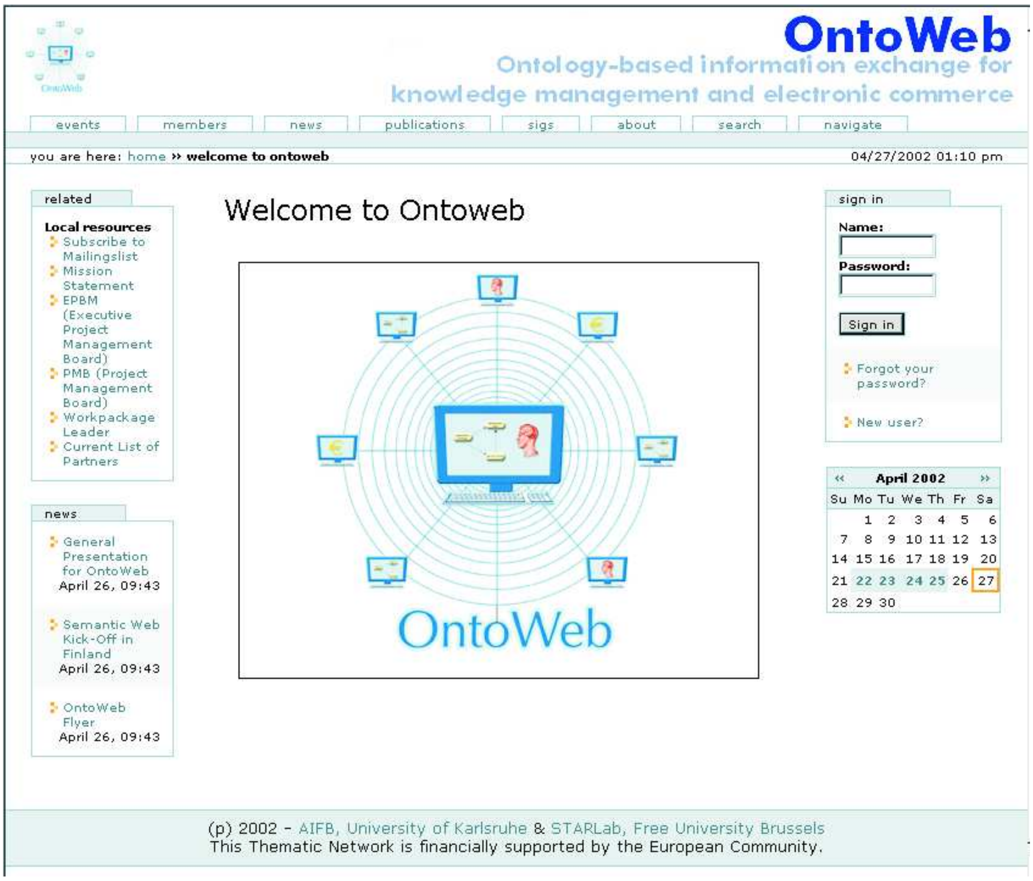
Figure 1: Extended conceptual SEAL architecture

SEAL (SEmantic PortAL, Figure 1)[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 [6]), 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 [12] 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.

[1] Cf. [9] on the history of SEAL.

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

3 OntoWeb Scenario



you are here: home » **welcome to ontoweb** 04/27/2002 01:10 pm

Welcome to Ontoweb

Local resources

- Subscribe to Mailingslist
- Mission Statement
- EPBM (Executive Project Management Board)
- PMB (Project Management Board)
- Workpackage Leader
- Current List of Partners

- General Presentation for OntoWeb
  April 26, 09:43
- Semantic Web Kick-Off in Finland
  April 26, 09:43
- OntoWeb Flyer
  April 26, 09:43

- Forgot your password?
- New user?

«	**April 2002**					»
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

(p) 2002 - AIFB, University of Karlsruhe & STARLab, Free University Brussels
This Thematic Network is financially supported by the European Community.

Figure 2: www.ontoweb.org – The OntoWeb portal

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. [1]). 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 2) 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 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 1 anf 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 [13, 6]. 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 1, 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, 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

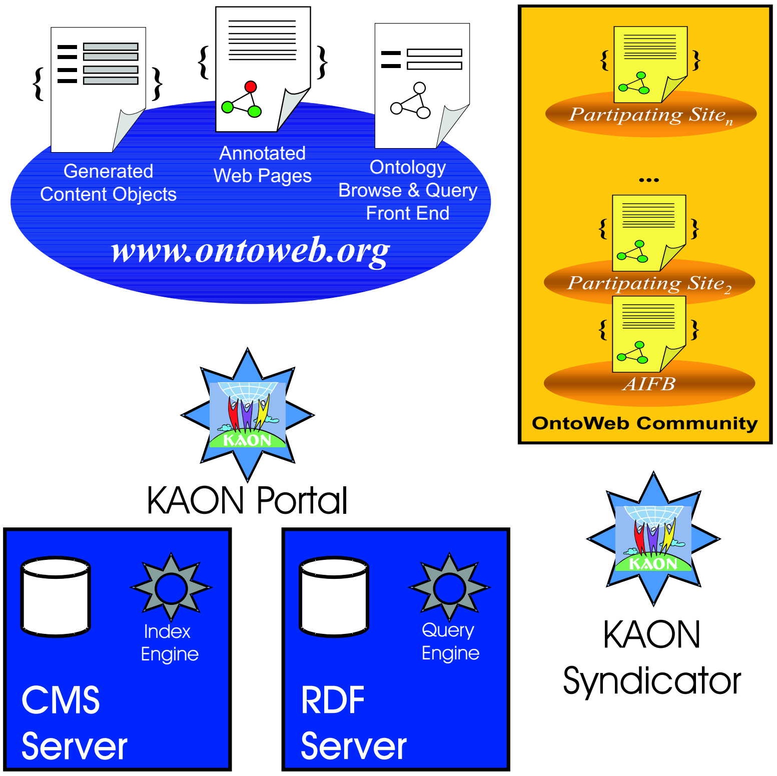


Figure 3: OntoWeb architecture

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 [7] 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 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 1) are implemented as KAON Portal (cf. Figure 3). 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].

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

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

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 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. In the private state the respective object is only visible to the user himself, the pending state makes it visible to reviewers. In the published state, a given object is visible to all (possibly anonymous) users of the portal. 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 [10] and AutoWeb [3], 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 approaches like Strudel [6] and Tiramisu [2], 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

[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.

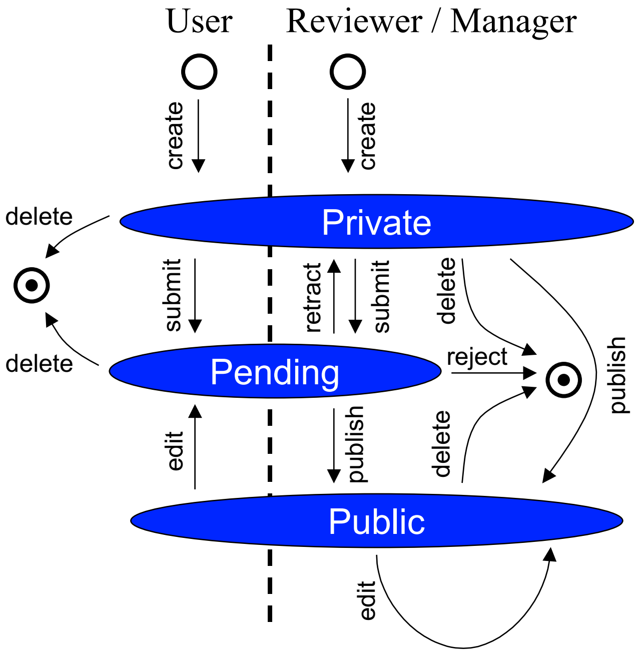


Figure 4: SEAL Publishing workflow

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] Thematic Network EU IST-2000-25056 OntoWeb: Annex 1 - "Description of Work". Technical report, Information Societies Technology (IST) Programme, February 11 2001.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] G. Wiederhold and M. Genesereth. The conceptual basis for mediation services. *IEEE Expert*, 12(5):38–47, Sep.-Oct. 1997.