SEAL — Tying Up Information Integration and Web Site Management by Ontologies

[2]Alexander Maedche, [1,3]Steffen Staab, [1,2,3]Rudi Studer, [1]York Sure, [1]Raphael Volz
maedche@fzi.de
{staab,studer,sure,volz}@aifb.uni-karlsruhe.de

[1]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS/
[2]FZI Research Center for Information Technologies,
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
http://www.fzi.de/wim/
[3]Ontoprise GmbH, Haid-und-Neu-Str. 7, 76131 Karlsruhe, Germany
http://www.ontoprise.de

Abstract

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 (SEmantic portAL) is a conceptual model that exploits ontologies for fulfilling the requirements set forth by these two properties at once. The ontology provides a high level of sophistication for web information integration as well as for web site management. We describe the SEAL conceptual architecture as well as its current implementation in KAON.

1 Introduction

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* [10]. The conceptual architecture developed for this purpose now generally consists of a three layer architecture comprising (cf. [24])

1. heterogeneous **data sources** (e.g., databases, XML, but also data found in HTML tables),
2. **wrappers** that lift these data sources onto a common data model (e.g. OEM [18] or RDF [16]),
3. integration modules (**mediators** in the dynamic case) that reconcile the varying semantics of the different data sources.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

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, 5, 4, 12, 11, 17]). 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 [6, 19, 13].

Semantic Portal. The topic of this paper is SEAL (SEmantic PortAL), a framework for managing community web sites and web portals on an ontology basis. The ontology supports queries to multiple sources (a task also supported by semi-structured data models [11]), but beyond that it also includes the intensive use of the schema information itself allowing for automatic generation of navigational views[1] 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 [20] 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.

The SEAL conceptual architecture (cf. Figure 1; details to be explained in subsequent sections) depicts the general scheme. Approaches for web site management emphasize on the upper part of the figure and approaches for web information integration focus on the lower part while SEAL combines both with an ontology as the knot in the middle.

History. The origins of SEAL lie in Ontobroker [8], which was conceived for semantic search of knowledge on the Web and also used for sharing knowledge on the Web [3], also taking advantage of the mediation capabilities of ontologies [10]. It then developed into an overarching framework for search and presentation offering access at a portal site [20]. This concept was then transferred to further applications [2],[22] and constitutes the technological basis for the portal of our institution[2] (among others)[3]. It now combines the roles of information integration in order to provide data for the Semantic Web and for a Peer-to-Peer network with presentation to human Web surfers.

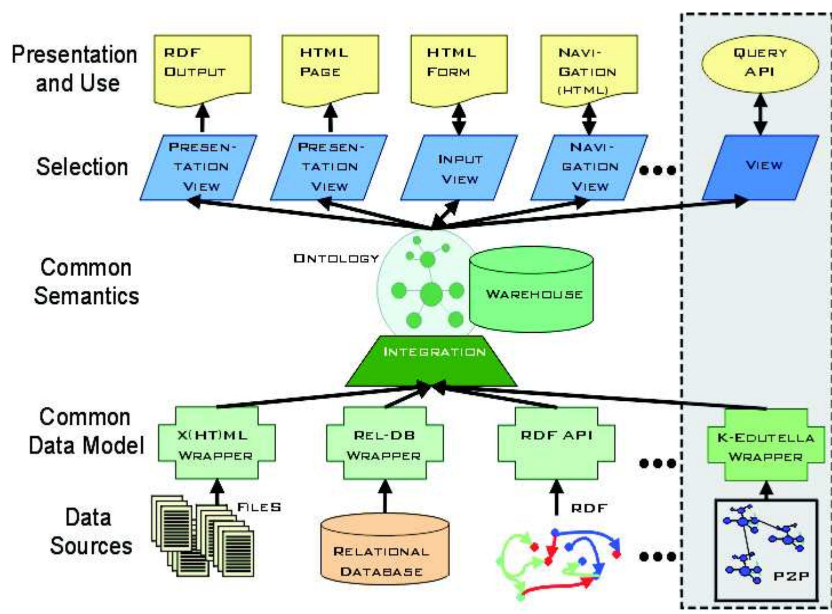

Figure 1: SEAL conceptual architecture

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

[2] http://www.aifb.uni-karlsruhe.de

[3] Also the web portal of of the EU-funded thematic network "OntoWeb" (http://www.ontoweb.org) and the KA2 community web portal (http://ka2portal.aifb.uni-karlsruhe.de)

2 Web Information 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 [24, 11]. 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 **data sources** of the Web as input: First of all, to a large part the Web consists of static HTML pages, often semi-structured, including tables, lists, etc. We have developed an ontology-based **HTML wrapper** that is based on a semi-supervised annotation approach. Thus, based on a set of predefined manually annotated HTML pages, the structure of new HTML pages is analyzed, compared with the annotated HTML pages and relevant information is extracted from the HTML page. The HTML wrapper is currently extended to also deal with heterogeneous XML files. Second, we use an automatic XML wrapping approach that has been introduced in [9]. The idea behind this wrapping approach is that these XML documents refer to an DTD that has been generated from the ontology. Therefore we automatically generate a mapping from XML to our data model so that integration comes for free. Third, data-intensive applications typically rely on relational databases. A relational database wrapping approach [21] maps relational database schemas onto ontologies that form the semantic basis for the RDF statements that are automatically created from the relational database. Fourth, in an ideal case content providers have been registered and agreed to describe and enrich their content with RDF-based metadata according to a shared ontology. In this case, we may easily integrate the content automatically by executing an **integration** process. If content providers have not been registered, but provide RDF-based metadata on their Web pages, we use ontology-focused metadata discovery and crawling techniques to detect relevant RDF statements.

Our generic Web information integration architecture is extensible, as shown in Figure 1. In particular, we are currently working on connecting and integrating data sources available via enhanced **Peer-2-Peer (P2P)** networks. P2P applications for searching and exchanging information over the Web have become increasingly popular. The **Edutella**[4] approach builds upon the RDF metadata standard aiming to provide an RDF-based metadata infrastructure for P2P applications, building on the recently announced JXTA framework.

It is important to mention that in our current architecture and implementation we mainly apply **static** information integration building on a warehousing approach. Means for **dynamic** information integration are currently approached for Peer-2-Peer networks and within our relational database wrapper.

3 Web Site Management

One difficulty of community portals lies in integrating heterogeneous data sources. Each source may be hosted by different community members or external parties and fulfills different requirements. Therefore typically all sources vary in structure and design. Community portals like (in our case) the web site of our own institute require coherence in hosted information on different levels. While the information integration aspect (see previous section) satisfies the need for a coherent structure that is provided by the ontology we will now introduce various facilities for construction and maintenance of websites to offer coherent style and design. Each facility is illustrated by our conceptual architecture (cf. Figure 1).

Presentation view. Based on the integrated data in the warehouse we define user-dependent presentation views. First, as a contribution to the Semantic Web, our architecture is dedicated to satisfy the needs of software agents and produces machine understandable RDF. Second, 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.

[4] http://edutella.jxta.org

Input view. 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 [14] 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 (cf. Figure 3).

Navigation view. To navigate and browse the warehouse we automatically generate navigational structures 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. Therefore especially users that are unfamiliar with the portal are supported to explore the schema and corresponding content.

(General) View. In the future we plan to explore techniques of handling updates on these views.

4 Technical Architecture

The technical architecture of SEAL is derived from the architecture of KAON, the Karlsruhe Semantic Web and Ontology Infrastructure[5], whose components provide the required functionalities described in the previous sections. The architecture of KAON is depicted in Figure 2. KAON components can roughly be grouped into three layers.

The data and remote services layer represents optional external services, which can be used in the upper layers, e.g. reasoning services for inferencing and querying, or connectors to the Edutella Peer-To-Peer network, and alternative storage mechanisms for the data in the previously mentioned warehouse.

The middleware layer provides a high-level API for manipulating ontologies and associated data and hides the actual manner of storage and communication from all clients. Thus clients cannot distinguish between working on the local file system (provided by the RDF API) or working on a multi-user aware server which stores data in a relational database. The middleware also provides interfaces to QEL, the query language used within the Edutella network, which is not only used to communicate queries within the peer-to-peer network but also used to query the warehouse.

The application and services layer groups applications that use services from the underlying layers. Currently these are one hand, stand-alone desktop applications built using the Ont-O-Mat application framework or portals built using the KAON portal maker, which provides the features discussed in section 3. Ont-O-Mat applications are built as plug-ins that are hosted by the Ont-O-Mat application framework. This approach guarantees maximum application interoperability within Ont-O-Mat.

Finally, core to KAON is the domain ontology itself, which is represented in RDF Schema[23] - the data model at hand for representing ontologies in the Semantic Web. It provides basic class and property hierarchies and relations between classes and objects. Historically SEAL leverages the mapping of RDF Schema model to F-Logic[15] introduced in [7] to provide views (in form of logical axioms) and a query mechanism. This allows us to rely on the reasoning services offered by OntoBroker [8] or SiLRi [7] .

5 Creating a SEAL-based Web Site

The creation of a SEAL-based web site is a multi-step process. The genesis starts with the creation of the ontology, which provides a conceptualization of the domain and is later used as the content model of the portal.

Step 1 – Ontology design: Here, several tools come in handy, within KAON Ont-O-Mat SOEP provides an editor with strong abilities regarding the evolution of the ontology. OntoEdit is a commercial tool that additionally allows to provide F-Logic axioms to refine the ontology.

[5] http://kaon.semanticweb.org

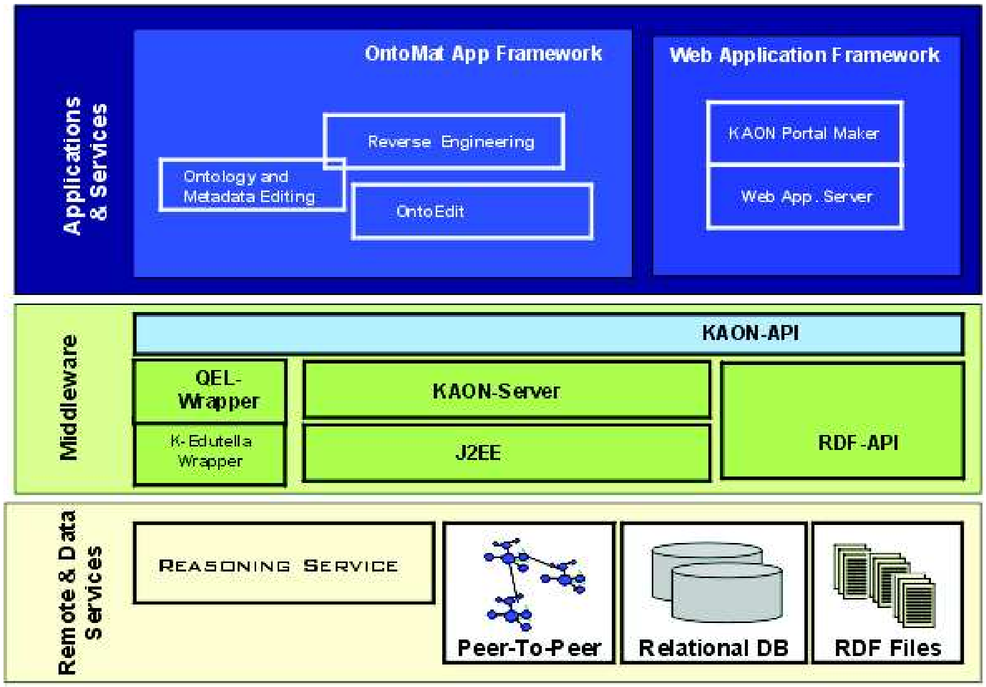

Figure 2: KAON architecture

Step 2 – Integrating Information: The next step towards the final web site is providing data. Here, we take a warehousing approach to amalgamate information coming from heterogeneous data sources.

- *RDF metadata* User-supplied HTML and PDF documents have to be annotated with metadata based on the content ontology in order to be part of the SEAL portal. These documents can be located anywhere on the web and are made part of the portal using KAON Syndicator, a component that gathers the meta data contained in resources located on the web.

- *Database Content* Today most large-scale web applications present content derived from databases. KAON REVERSE is an application that provides visual means to map the logical schema of relational databases to the integrated conceptual model provided by the ontology [21]. The user-supplied mappings are then used to transform the database content to ontology-based RDF.

- *Peer-To-Peer* Also connectors to the **Edutella** peer-to-peer network[6], that provides an RDF-based metadata infrastructure for peer-to-peer applications, are currently constructed within KAON. SEAL portals can then be used to provide a web accessible interface to Edutella based Peer-To-Peer networks

Step 3 – Site design: We derive the previously mentioned navigation model and personalization model from the ontology. Currently no extensive tool support for these tasks exist. Both models are derived from the ontology using F-Logic queries that are provided by the site administrator.

Navigation model Beside the hierarchical, tree-based hyperlink structure which corresponds to the hierarchical decomposition of the domain, the navigation module enables complex graph-based semantic hyperlinking, based on ontological relations between concepts (nodes) in the domain. The conceptual approach to hyperlinking is based on the assumption that semantically relevant hyperlinks from a web page correspond to conceptual relations, such as `memberOf` or `hasPart`, or to attributes, like `hasName`. Thus, instances in the knowledge base

[6] http://edutella.jxta.org

may be presented by automatically generating links to all related instances. For example, on personal web pages there are, among others, hyperlinks to web pages that describe the corresponding research groups, secretary and professional activities (*cf.* Figure 3).

Step 4 – Web design: The derived models constructed in step 3 serve as input to the KAON Portal Maker, which renders the information in HTML. The implementation of KAON portal Maker adheres strictly to a model-view-controller design pattern. The ontology and the derived models are encapsulated by an abstract data model and the presentation of the information is created using template technologies like JSP, ASP or XSLT.

Default controllers are provided for standard application logic like updating data and generating links to other presentation objects. The reader may note that the default controllers can be replaced by custom-made controllers provided by the site administration.

KAON Portals also provides default templates that provide the most often used representations for information objects (like list-entries, forms for web-based data provision etc.) For instance, the AIFB portal includes an input template (*cf.* Figure 3, upper part) generated from the concept definition of person (*cf.* Figure 3, middle left) and a sheet like representation to produce the corresponding person web page (*cf.* Figure 3, lower part). These default templates can easily be customized for special purposes.

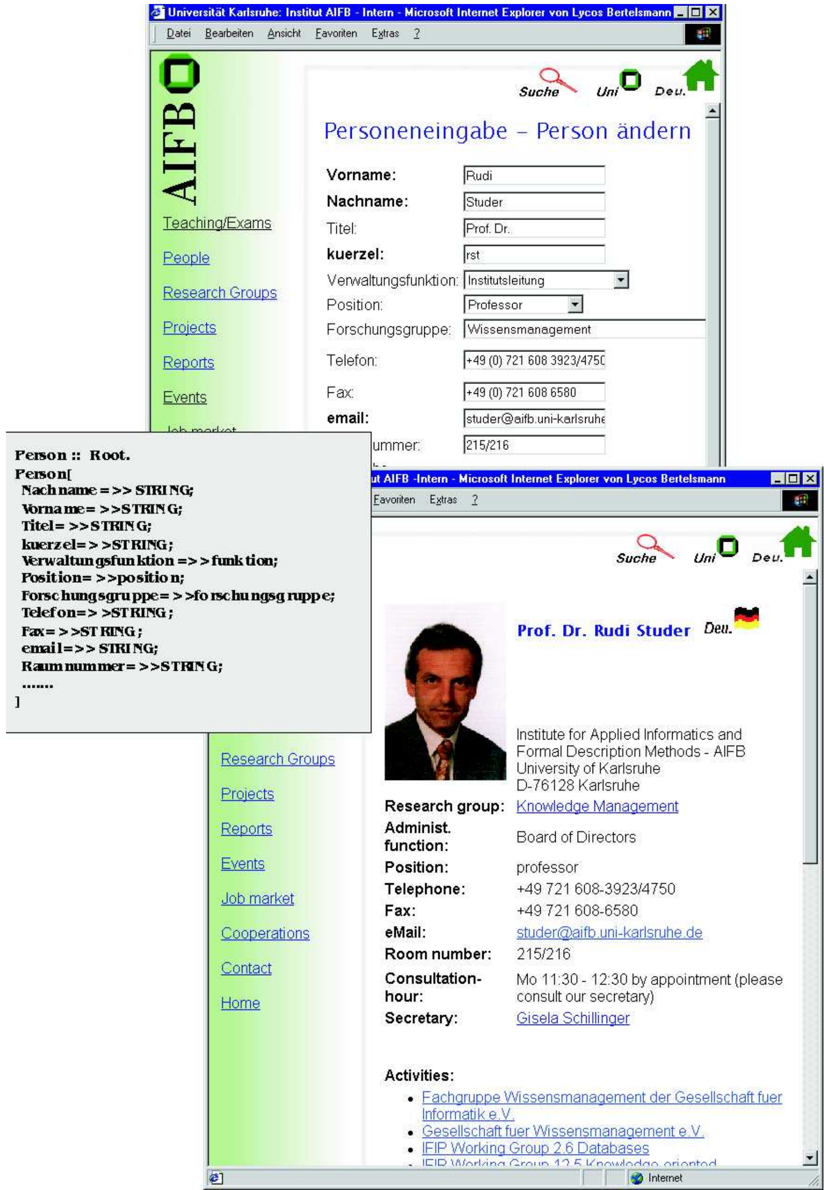

Figure 3: Templates generated from the web-site models

6 Discussion

The SEAL approach offers a comprehensive conceptual framework for Web information integration and Web site management. A crucial feature of SEAL is the use of an ontology as a semantic backbone for the framework. Thus, all functions for information integration as well as for information selection and presentation are glued together by a semantic conceptual model, i.e. a domain ontology. Such an ontology offers a rich structuring of concepts and relations that is supplemented by axioms for specifying additional semantic aspects. The ontological foundation of SEAL is the main distinguishing feature when comparing SEAL with approaches from the information systems community.

The STRUDEL system [11] is an approach for implementing data-intensive Web sites. STRUDEL provides a clear separation of three tasks that are important for building up a data-intensive Web site: (i) accessing and integrating the data available in the Web site, (ii) building up the structure and content of the site, and (iii) generating the HTML representation of the site pages. Basically, STRUDEL relies on a mediator architecture where the semi-structured OEM data model is used at the mediation level to provide a homogeneous view on the underlying data sources. STRUDEL then uses so-called 'site definition queries' to specify the structure and content of a Web site. When compared to our SEAL approach STRUDEL lacks the semantic level that is defined by the ontology. Furthermore, within SEAL the ontology offers a rich conceptual view on the underlying sources that is shared by the Web site users and that is made accessible at the user interface for e.g. browsing and querying.

The Web Modeling Language WebML [4] provides means for specifying complex Web sites on a conceptual level. Aspects that are covered by WebML are a.o. descriptions of the site content, the layout and navigation structure as well as personalization features.Thus, WebML addresses functionalities that are offered by the presentation and selection layer of the SEAL conceptual architecture. Whereas WebML provides more sophisticated means for e.g. specifying the navigation structure, SEAL offers more powerful means for accessing the content of the Web site, e.g. by semantic querying.

In addition to ongoing work to integrate Peer-to-Peer functions for accessing information on the Web, two topics are currently under investigation: first, the view concept that is implemented by the KAON framework does not support updates in general. Currently, only the simplistic input views provide means for updating the warehouse. Clearly, Web site users do expect to be able to update the site content. A second topic that needs further improvement is the handling of ontologies. Just offering a single, centralized ontology for all Web site users does not meet the requirements for heterogeneous user groups. Therefore, methods and tools are under development that support the handling and aligning of multiple ontologies.

The SEAL framework as well as the KAON infrastructure can be seen as steps for realizing the idea of the Semantic Web. Obviously, further steps are needed to transfer these approaches into practice.

Acknowledgements. We thank our colleagues and students at the Institute AIFB, University of Karlsruhe, at FZI Research Center for Information Technologies at the University of Karlsruhe and at Ontoprise GmbH for many fruitful interactions. Especially, we would like to thank our colleagues Siegfried Handschuh and Nenad Stojanovic for their contributions to the SEAL framework. Research reported in this paper has been partially financed by EU in the IST projects On-To-Knowledge (IST-1999-10132) and Ontologging (IST-2000-28293).

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] J. Angele, H.-P. Schnurr, S. Staab, and R. Studer. The times they are a-changin' — the corporate history analyzer. In D. Mahling and U. Reimer, editors, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland, October 30-21, 2000*, 2000. http://www.research.swisslife.ch/pakm2000/.

[3] V. R. Benjamins, D. Fensel, S. Decker, and A. G. Perez. $(KA)^2$: Building ontologies for the internet. *International Journal of Human-Computer Studies (IJHCS)*, 51(1):687–712, 1999.

[4] 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.

[5] S. Ceri, P. Fraternali, and S. Paraboschi. Data-driven one-to-one web site generation for data-intensive applications. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 615–626, 1999.

[6] 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.

[7] S. Decker, D. Brickley, J. Saarela, and J. Angele. A query and inference service for RDF. In *QL98 - Query Languages Workshop*, December 1998.

[8] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editors, *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–269. Kluwer Academic Publisher, 1999.

[9] M. Erdmann and R. Studer. How to structure and access XML documents with ontologies. *Data and Knowledge Engineering*, 36(3):317–235, 2001.

[10] 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):361–282, 2000.

[11] 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.

[12] P. Fraternali and P. Paolini. A conceptual model and a tool environment for developing more scalable, dynamic, and customizable web applications. In *EDBT 1998*, pages 421–435, 1998.

[13] C. Goble, S. Bechhofer, L. Carr, D. De Roure, and W. Hall. Conceptual open hypermedia = the semantic web? In *Proceedings of the Second International Workshop on the Semantic Web - SemWeb'2001, Hongkong, China, May 1, 2001*. CEUR Workshop Proceedings, 2001. http: //CEUR-WS.org/Vol-40/.

[14] 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.

[15] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843, 1995.

[16] O. Lassila and R. Swick. Resource description framework (RDF). model and syntax specification. Technical report, W3C, 1999. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax.

[17] 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)* , May 1999.

[18] 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.

[19] G. Rossi, A. Garrido, and D. Schwabe. Navigating between objects. lessons from an object-oriented framework perspective. *ACM Computing Surveys*, 32(30), 2000.

[20] 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.

[21] L. Stojanovic, N. Stojanovic, and R. Volz. Migrating data-intensive web sites into the semantic web. In *Proceedings of the ACM Symposium on Applied Computing SAC-02, Madrid, 2002*, 2002.

[22] Y. Sure, A. Maedche, and S. Staab. Leveraging corporate skill knowledge - From ProPer to OntoProper. In D. Mahling and U. Reimer, editors, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland, October 30-21, 2000*, 2000. http://www.research.swisslife.ch/pakm2000/.

[23] W3C. RDF Schema Specification. http://www.w3.org/TR/PR-rdf-schema/, 1999.

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