Semantic Description of Documents in Enterprise Knowledge Infrastructures

Ronald Maier, René Peinl

Chair for Management Information Systems
Martin-Luther-University Halle-Wittenberg
Universitaetsring 3, D-06108 Halle (Saale), Germany
ronald.maier@wiwi.uni-halle.de, rene.peinl@wiwi.uni-halle.de

Abstract: Organizations have increasingly knowledge-intensive business processes that require a wealth of electronic resources which are scattered across a number of systems but are implicitly linked to each other. Meta-data annotations to these electronic resources can support users in retrieving related documents which is a key to provide advanced knowledge services in an enterprise knowledge infrastructure. The improved retrieval of documents ultimately aims at increasing productivity of knowledge work,. This paper discusses all necessary standards, languages and tools required to create, store and retrieve meta-data from the perspective of an enterprise knowledge infrastructure which should provide all means to do this within its integration layer.

1 Introduction

Organizations these days have a wealth of electronic resources scattered across a number of systems. These resources are implicitly linked to each other by having the same authors, referencing the same processes or projects or discussing the same topics. Because these links are only implicit, users get no support in retrieving all documents relevant for the tasks they are working on. This is particularly true in the case of knowledge-intensive processes for which tasks, workflows, required electronic resources and cooperation partners can only be described vaguely in advance.

Knowledge management systems have been proposed to overcome these limitations and offer advanced, personalized knowledge services that are combined into knowledge management instruments, such as experience management, lessons learned, good/best practices or communities [Ma04]. If such systems target an entire organization and reuse existing systems as much as possible they are called enterprise knowledge infrastructures (EKI). An EKI is a comprehensive ICT platform for knowledge sharing, collaboration and learning with advanced, integrated knowledge services that are personalized for participants networked in communities and foster the implementation of KM instruments in support of knowledge processes targeted at increasing productivity of knowledge work ([MHP05], p. 73). Figure 1 shows an ideal layered architecture for such an infrastructure. Users can access knowledge services using different server- and client-side technologies on a variety of devices. Knowledge services are used to exchange implicit (collaboration) and explicit knowledge (publication and discovery) as well as to

support individual training (learning). Integration services provide a semantically unified view on different information sources available in the corporate Intranet or the Internet. These information sources are made available through infrastructure services for storage, access, messaging, processing and security on the lowest layer.

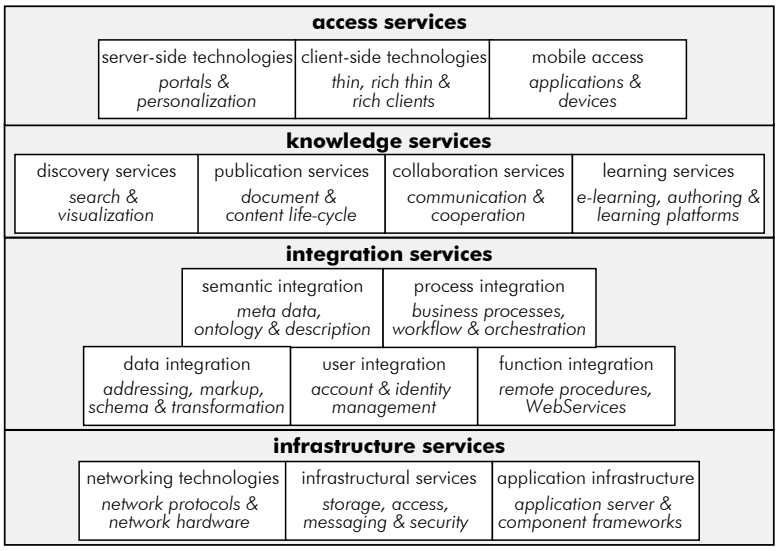

Figure 1: Architecture of an Enterprise Knowledge Infrastructure (cf. [MHP05], p.76)

This paper addresses the core integration layer of such an infrastructure that provides access to the heterogeneous data and knowledge sources of an organization in a semantically integrated way, so that advanced knowledge services can be built on top. The integration layer consists on the one hand of function-oriented integration services (function and process integration) and on the other hand of data-oriented integration services (data, user and semantic integration). Data-oriented integration services are the focus of this paper. The electronic resources mainly used in knowledge-intensive processes in organizations are semi-structured documents which have to be semantically described using common meta-data standards and semantically rich content and ontology description languages. Goal of this paper is to review current meta-data standards, languages and tools and their applicability for integration services in an EKI. To achieve that we reflect on the use of the terms document, meta-data and ontology in EKIs (section 2), review meta-data standards (section 3) and study state-of-the-art languages and tools to create, store and retrieve meta-data (section 4). Finally, section 5 compares the offerings of these approaches to the requirements posed by EKIs.

2 Documents in Organizations

Organizations handle an increasing amount of electronic resources. The types of data that have to be considered have been extended from structured data as can be found in database systems to semi-structured data typically found in e.g., document management

systems (DMS), file servers, content management systems or email servers. As compared to structured data, semi-structured data has not been managed equally well in most organizations. Recently, many organizations have realized the potential cost savings and valuable insights that can be gained from a more systematic management of semi-structured data with the help of an EKI [MHP05]. A large number of terms have been coined for semi-structured data, e.g., content, (digital) asset or document.

Document. A document is a legally sanctioned record (e.g., purchase order) or a transitory record (e.g., meeting notes) of a business transaction or decision that can be viewed as a single organized unit both from a business and from a technical perspective. It is composed of a grouping of information objects, also called content, plus format information. Documents can be (1) elementary documents, e.g., a text file or a fax message, (2) compound documents, e.g., a text file with embedded graphs, tables or pictures or (3) container documents, e.g., a collection of elementary or complex documents organized around a work flow in a folder or zip file ([KM97], p. 12). Documents have business value and thus can be considered as assets. Document types can be distinguished along a number of dimensions, for example:

- *Coding*: a project report stored as a text file is an example for *coded information* (CI), whereas the same report scanned from paper and archived as an image file would be *non-coded information* (NCI).
- *Structure*: documents containing *structured data* like the sales numbers for the last quarter have to be distinguished from *semi-structured data* like a project report.
- *File format and type*: *file formats* like JPEG, PNG and GIF are implementation-specific and more fine granular than *file types* like bitmap graphics and vector graphics or, in a further abstraction, text, image, audio and video *contents*.
- *Purpose*: since most documents used in EKIs are compound documents, a purpose-oriented classification seems more appropriate. Presentations, scientific articles and experience reports are examples of such classes, or knowledge types such as lessons learned, good/best practices, questions and answers or learning objects.

The file format is not sufficient to determine the content or purpose of a document, e.g., an XML file can be a text processor document, a spreadsheet, or a scalable vector graphic (SVG). EKIs primarily deal with semi-structured, compound documents containing coded information for different purposes. The document type has great impact on the requirements for meta-data description, e.g., a full text search may lead to a feasible result for a text document, but not for an image.

Meta-data are data about data. An EKI contains documents as well as meta-data which give further information about their content and associations. There are a number of reasons to assign meta-data to documents, e.g., increased accessibility and smarter retrieval, better retention of context, versioning, complying to legal and security requirements as well as improving system performance and economics [GS02]. Meta-data can be used to describe any kind of data from structured to unstructured. The structure itself already is a form of meta-data and usually provides information about the name of the data element, its data type and its basic relation to other data elements (e.g., an XML Schema for an XML document). Element names are often not sufficient.

Additional meta-data is needed that either describes the content (e.g., keywords, domain) or the context of the data especially for semi-structured data. The context can be further subdivided into creation context (e.g., author, creation date) and application context (e.g., customer, intended use) or even finer according to the document life-cycle into creation, storage, retrieval, application and archiving context [GS02]. Meta-data can be informal (e.g., free text description), semi-formal (e.g., structured according to a user-invented structure) or formal (e.g., structured and compliant to a standard). Summing up, three types of meta-data can be identified (in extension of [GS02]): (1) *Content meta-data* relate to what the object contains or is about and are intrinsic to an information object. (2) *Context meta-data* indicate aspects associated with the object's creation and/or application and are extrinsic to an information object (e.g., who, what, why, when, where and how aspects). (3) *Structure meta-data* relate to the formal set of associations within or between individual information objects and can be intrinsic or extrinsic. *Technical meta-data* are a special form of structure meta-data and capture file format-specific data such as compression algorithm or resolution of included pictures.

Ontology. Knowledge modeling aims at a formal description of (documented) organizational knowledge that can be processed by computers and help to exchange and share knowledge with EKI [MHP05]. "An ontology is a (1) formal, (2) explicit specification of a (3) shared (4) conceptualization" ([Gr93], p. 199). (1) An ontology has to be formal which requires that the ontology should be machine-readable. (2) Explicit specification means that the concepts and relationships as well as the constraints on the use of concepts are defined openly and not left to the interpretation of the ontology's users. (3) Shared refers to the requirement that the conceptualizations made in an ontology have to be agreed upon by a group of people that intend to use the ontology. (4) Finally, conceptualization is an abstract model, a representation of a domain or phenomenon which investigates the concepts of that domain or phenomenon relevant to the ontology's users.

From the perspective of an EKI, ontologies are therefore developed to provide machine-processable semantics of electronic resources that are accepted by the members of the organization and facilitate organization-wide knowledge sharing and reuse.

3 Meta-data annotations

Integration of documents in EKIs with the help of meta-data requires a standard language for the serialization of meta-data annotations, a content-oriented standard to define the available meta-data fields and a standard language to formalize an ontology. The latter is used to define the domain and range of meta-data fields and relate meta-data on the type level as well as individual document objects or real-world objects on the instance level by reasoning about the defined concepts.

A number of institutions have developed standards and started initiatives to provide comprehensive frameworks for definition and exchange of meta-data, i.e. semantic information about documents, especially books, journals, images, photographs, audio and video files. Examples for institutions, standards and initiatives are the World Wide

Web consortium (W3C) with XML and the Semantic Web initiative, the International Standardization Organization (ISO) with standards for document exchange, e.g., the Motion Picture Experts Group (MPEG) 7 meta-data standard for images, audio and video files or the Topic Map standard as well as the Dublin Core standard for exchanging meta-data about text documents (cf. [DFH03], [SH05]).

Automated reasoning about documents is a complex task. Thus, the Semantic Web initiative breaks down the sub-tasks into a layered structure. From a meta-data point of view, this comprises the description of (1) the internal structure of documents defined with XML and XML Schema, (2) the scope or domain in which the specified names in the markup are valid, called a namespace, (3) how to query and translate a document that is an instance of one schema so that it conforms to another schema with XPath and XSLT. Based on these standards, semantic description is accomplished with the help of (4) statements that describe Web resources with the Resource Description Framework (RDF) and the RDF Schema language, (5) ontologies that show relationships between concepts used in the descriptions which are defined with the Web Ontology Language (OWL) and (6) rules as well as a logic framework that allow for advanced reasoning about documents and their descriptions (cf. [DOS03], [Fe04]).

Content-oriented meta-data standards focus on standardization of meta-data fields and can be serialized with the help of languages like XML and RDF. There are a number of domain-independent initiatives to standardize meta-data, e.g., Dublin Core [Hi05], Digital Object Identifier (www.doi.org), or the Text Encoding Initiative (www.tei-c.org). Additionally, there are a large number of domain-specific meta-data standards, e.g., in the areas of publishing, library, education, museum or multimedia. Examples are Learning Object Metadata [HD02], PRISM (www.prismstandard.org) or MPEG-7 [MKP02]. Standards can be compared according to e.g., comprehensiveness, flexibility, languages used for serialization, adoption rate or user friendliness.

An ontology can be used to relate the meta-data fields. Popular ontology languages include DAML+OIL, Ontolingua and OWL [Fe04]. Ontologies for an EKI can be developed on the basis of existing ontology types, like enterprise ontologies that define organizational structure, domain-task ontologies that define processes, domain ontologies that define relevant topics and common sense ontologies that define location and time concepts [GFC04]. Recently, more comprehensive specific ontologies have been proposed for a variety of domains. Of special interest for EKIs are publication descriptions using BibTeX in OWL (visus.mit.edu/bibtex/0.1/) or the AKT Portal ontology that describes academic researchers, their publications and projects (www.aktors.org/ontology/).

4 Creation, storage and retrieval of meta-data

An integration layer in an EKI has to offer services for (1) creating meta-data describing heterogeneous documents, (2) storing it either together with or separated from the documents in a repository and (3) retrieving it for inferencing to enable advanced knowledge services.

Creation: The creation of meta-data in most organizations is primarily accomplished manually. Often, the user is prompted to type in author, title and keywords describing a document before she can save it to e.g., a DMS. Even more inconvenient is the manual creation of an RDF file to annotate e.g., a Web resource, since there is no form popping up that needs to be filled, but the RDF-tags often have to be created manually and it may even be unclear which fields should be used. From an EKI view, a manual approach is not appropriate due to the amount of documents. There are some first steps towards (semi-) automated creation of meta-data which either use document-inherent structures and tags like DC-Dot (www.ukoln.ac.uk/metadata/dcdot/) that utilizes HTML tags to generate Dublin-Core conforming RDF annotations or sophisticated text-mining and language-processing algorithms to extract meta-data from the content like TextToOnto (sourceforge.net/projects/texttoonto). Some meta-data can be more easily extracted if the document is structured using an XML-format like DocBook (www.docbook.org) that already incorporates most Dublin-Core elements.

Storage: Basically, meta-data can be stored either inline, as part of the document, like in MS Word or Adobe PDF documents, or document-external, e.g., in a separate RDF file or in a relational database like many DMS do. XML-documents also allow to store RDF annotations inline using the XML namespace concept. Inline storage is especially advantageous when documents are exchanged between several EKIs, e.g., between a company and one of its cooperation partners or in a peer-to-peer scenario where documents are stored in a distributed environment. In this case, the sending EKI packs all document descriptions relevant for the target environment together with every exchanged document which can then be extracted by the receiving EKI. For searching large document collections, it is not efficient to store RDF data only inline or in separate files, so the need arises for a way to store RDF data in and retrieve it from a database. In general, either relational, object-oriented, XML-based databases or proprietary database formats can be used. In an EKI setting, relational databases might be preferred due to their dominance and the fact that common drawbacks for XML storage like missing whitespace preservation or breaking digitally signed contents do not seem to be an issue here. Thus, this approach is examined closer [Me01]:

One method would be to store all RDF triples in one table which results in denormalized data. Separate tables for resources, literals, namespaces and statements would dramatically decrease required storage capacity, but also decrease performance as a number of computation-intensive joins have to be made. The Jena toolkit uses the former approach, whereas Sesame is an example for a tool that implements the latter approach. Finally, one could also store RDF data in a database schema according to the RDF schema describing the structure of the RDF file. This potentially results in a large number of tables and makes it more difficult to retrieve statements independently from their RDF schema, but can also improve retrieval for a fixed and small number of schemas.

Retrieval: Established query languages like SQL, OQL or XPath/XQuery could be used in order to retrieve meta-data from the database, depending on the type of database management system used. However, there are a number of shortcomings that could be overcome with a new query language that explicitly supports the RDF triple structure

and other RDF language constructs. A number of proposals for such languages have been made, e.g., iTQL, RDFQL, RDQL, RQL, SeRQL, and SPARQL. Although these languages look similar, since they all imitate SQL[1], their capabilities are quite different.

Haase et al. evaluate a number of these languages and define the following requirements for an RDF query language [HB+04]: support for (1) RDF abstract data model, (2) formal semantics and inference, (3) XML schema data types for literals and (4) statements about resources. They further judge the languages according to their (5) expressiveness, (6) closure, (7) adequacy, (8) orthogonality and (9) safety. They conclude that especially grouping and aggregation, as well as sorting and optional matching are poorly or not at all supported. Also, RDF language elements like XML data types, containers and reification are only supported in a few cases. From an EKI perspective, language capabilities and industry support are important criteria. Stier's evaluations supervised by the authors [St05] as well as the updated results of Haase's research [Ha05] show that RDFQL scores better than other query languages. Nevertheless, it seems that either RDQL, due to its support by HP and implementation in several tools (e.g., Jena, RDFStore, Sesame, 3Store, RAP [W3C04]) or SPARQL due to its progress in the W3C standardization process [W3C05] will become widely accepted.

Tool support: There are a number of tools available that support RDF storage and retrieval, most of which are the results of academic research and are freely available [St05]. However, maybe as a result of that, only few tools are easy to use, most of them even lack a graphical user interface. Some remarkable exceptions are 4 Suite, Sesame, KAON and Kowari. Sesame supports RDQL as well as RQL and SeRQL, whereas most other tools only support one language. This is especially interesting for EKIs for flexibility reasons, as long as there is no clear standard yet. RDFQL support is only available in the commercial tool RDF Gateway from Intellidimension. A prototype implementation developed at the authors' department [St05] builds on top of Jena and enhances the toolkit with a Web-based client for retrieval as well as a Java-based graphical client with support for creating, storing and retrieving RDF from the database.

5 Discussion

The integration layer in an EKI builds on semantic descriptions of documents to provide functionality to the knowledge services on the upper layer, such as semantically relating documents to each other or identifying experts based on authorship. Thus, the creation, storage, retrieval and processing of meta-data and associated ontologies is required. With XML, RDF, OWL and RDF query languages as well as the use of content-oriented meta-data standards, a significant part of the required integration services can be realized. However, the lacking standardization of RDF query languages together with missing capabilities of the proposed standards and insufficient tool support inhibits a broad implementation of EKI integration layers in organizations. Moreover, despite a number of content-oriented meta-data standards that seem well-suited for their designated

[1] There are also a number of other languages that use different constructs like Kaon-QL, N3, Triple and Versa.

domains, there is no broadly accepted standard that performs well in an EKI context. The modularization that is already designed for some standards seems to be a step in the right direction. We imagine a more flexible, modular meta-data annotation system with a few basic attributes for all documents together with a set of document type-specific attributes. The meta-data should be organized according to the identified categories and the dimensions time, topic, location, person, process and type [MS04]. In addition to that, ontologies should be used for every dimension and to define the relation between the dimensions to enable inferencing.

References

[DOS03] Daconta, M. C.; Obrst, L.; Smith, K. T.: The Semantic Web - a Guide to the Future of XML, Web Services, and Knowledge Management, Indianapolis, Ind. et al., 2003

[DFH03] Davies, J.; Fensel, D.; van Harmelen, F. (eds.): Towards the Semantic Web - Ontology-driven Knowledge Management, 2003

[Fe04] Fensel, D.: Ontologies: A silver Bullet for Knowledge Management and Electronic Commerce, 2nd edition, Berlin et al. 2004

[GS02] Gilliland-Swetland, A. J.: Setting the Stage - Introduction to Metadata, last change: 2002-02-13, URL: http://www.getty.edu/research/conducting_research/standards/intrometadata, last accessed 2004-12-07

[GFC04] Gómez-Pérez, A.; Fernández-López, M.; Corcho, O.: Ontological Engineering, London 2004

[Gr93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications, in Knowledge Acquisition 5 (2), p. 199-220, 1993

[Ha05] Haase, P: A Comparison of RDF Query Languages - queries and updated results, last change: 2005-04-11, URL: http://www.aifb.uni-karlsruhe.de/WBS/pha/rdf-query/, last access: 2005-06-12

[HB+04] Haase, P; Broekstra, J.; Eberhart, A.; Volz, R.: A Comparison of RDF Query Languages, in: McIlraith, S. A.; Plexousakis, D.; van Harmelen, F. (eds.): The Semantic Web - ISWC 2004 Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004, p. 502 – 518

[HD02] Hodgins, W.; Duval, E. (eds.): Draft Standard for Learning Object Metadata, online resource, date: 2002-07-15, URL: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, last access: 2005-06-14

[Hi05] Hillmann, D.: Using Dublin Core, online resource, date: 2005-05-26, URL: http://dublincore.org/documents/2005/05/26/usageguide/, last access: 2005-06-14

[KM97] Kampffmeyer, U.; Merkel, B.: Grundlagen des Dokumentenmanagements, Wiesbaden, 1997

[Ma04] Maier, R.: Knowledge Management Systems - Information and Communication Technologies for Knowledge Management, 2nd edition, Berlin et al. 2004

[MHP05] Maier, R.; Hädrich, T.; Peinl, R.: Enterprise Knowledge Infrastructures, Berlin et al. 2005

[MS04] Maier, R., Sametinger, J.: Peer-to-Peer Information Workspaces in Infotop. In: International Journal of Software Engineering and Knowledge Engineering, 14 (1), p. 79-102, 2004

[MKP02] Martínez, J. M.; Koenen, R.; Pereira, F.: MPEG-7: the generic Multimedia Content Description Standard, online resource, date: 2002-04, URL: http://www.chiariglione.org/mpeg/tutorials/IEEEMM_mp7overview_withcopyrigth.pdf, last access: 2005-06-14

[Me01] Melnik, S.: Storing RDF in a relational database, last change: 2001-12-04, URL: http://www-db.stanford.edu/%7Emelnik/rdf/db.html, last access: 2005-06-12

[St05] Stier, B.: Verwaltung von RDF/RDF Schema-Statements - Werkzeugvergleich und prototypische Implementierung, diploma thesis, Martin-Luther University Halle-Wittenberg, Halle, Germany 2005

[SH05] Stuckenschmidt, H.; van Harmelen, F.: Information Sharing on the Semantic Web, Berlin et al., 2005

[W3C04] W3C: RDQL - A Query Language for RDF, W3C Member Submission 9 January 2004, URL: http://www.w3.org/Submission/RDQL/, last access: 2005-06-12;

[W3C05] W3C: SPARQL Query Language for RDF, W3C Working Draft 19 April 2005, URL: http://www.w3.org/TR/rdf-sparql-query/, last access: 2005-06-12