The Semantic Desktop - a Basis for Personal Knowledge Management

Leo Sauermann
Knowledge Management Department, DFKI GmbH,
Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern, Germany
and
Knowledge-Based Systems Group, University of Kaiserslautern
leo.sauermann@dfki.de

Key Words: Semantic Desktop, Semantic Web, personal knowledge management, information integration
Category: H.4.3, D.2.11

1 Introduction

Abstract: Knowledge Management software is software that integrates. Existing Data sources, process flows, application features from office appliances have to be brought together. There are different standards, consisting of data formats and communication protocols, that address this issue. The WWW and *Semantic Web* are designed to work on a worldwide scale and define those standards. We transfer the web standards to the desktop szenario, a vision we call *Semantic Desktop* – a Semantic Web enhanced desktop environment. Central is the idea of taking know-how from the Semantic Web to tackle personal information management. Existing desktop applications (email client, browser, office applications) are integrated, the semantic glue between them expressed using ontologies. We also present the *www.gnowsis.org* open source project by the DFKI that realizes parts of this vision. It is based on a Semantic Web Server running as desktop service. It was used in experiments and research projects and allows others to experiment. Knowledge management applications can be built on top of it, reducing the implementation cost. [1]

2 Introduction

Today we wish to perform knowledge work *anytime and anyplace*. This wish is inspired by the availability of the internet and by the experience we have by using services available on the WWW. So there is an ongoing shift from desktop-based systems to web-based systems. A problem here is the duality of web and desktop applications. Editing and creating information is usually done in desktop applications (email clients, address books, or word processing software). Documents are downloaded or received by email, edited and then sent

[1] This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW C01, Project EPOS: Evolving Personal to Organizational Memories).

on to others or posted on the web again. The duality is more and more replaced with a coherence. Office applications like *Microsoft Office* and *Open Office* can export their data in html. Web applications are used in office scenarios to realize organizational memories, search functions, collaboration environments. Software is either built to run on the web or it is *web-enabled*.

We want to push this coherence even further. Building information management systems would be much simpler if data on desktop computers could be treated like web resources. The state of the art in web architecture is the *Semantic Web*, "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [3]. An introduction to the Semantic Web has already been given by Dr. York Sure on the I-Know 2004 [17]. A translation of the Semantic Web to the topic of Desktop information systems is the next step.

In this paper we present the merge of Semantic Web and Desktop information systems, under the term *Semantic Desktop*, and how it can help in building information architectures. First, we will present the principles we envision for the Semantic Desktop. Following is a description of the open source framework *gnowsis* that we created to test the principles. This implementation is open for a broader community of researchers and practitioners to build on the ideas we present and to extend them. Typical information management software like topic map tools, mind maps, or wikis can be integrated to the Semantic Desktop. Based on the principles and the experience in the gnowsis project, we will discuss the effect on knowledge management and application integration. At the end we suggest how the Semantic Desktop can be used in related research areas and give an outlook on our future plans.

3 Semantic Desktop

In 2003 the Semantic Desktop idea was first introduced in a master thesis [15]. In this section we will describe the motivation for the Semantic Desktop.

The data accessed by knowledge workers comes from different, already existing applications. Emails, documents, contact information (address books) and calendar information are such obvious data sources. These are the vital resources that any personal information management system supports. There have been different approaches to embed these data sources into a Semantic Web scenario. Haystack [8] is a well known project to let individuals manage their information in a very free and integrated way. But the architecture is based on a centralized database and the integration of 3rd party applications is not straightforward, whereas in real life scenarios various information systems have to be integrated. The *Semantic Desktop* is targeted on integration. We want to see data from SQL databases, Office Applications and other common office appliances integrated - based on the Semantic Web Standards. To do this, we have to take several steps.

3.1 Everything is a Resource

The first step to a Semantic Desktop is a definition of the data entities. The term *resource* is used in the Semantic Web for information entities. Seen through the Semantic Web glasses these are images, documents, videos, emails, and other items we find on the web. We translate this term now into the desktop scenario. So every file on the desktop can be seen as a semantic web resource, as every email, photo, address book entry, and all other information we find on a typical PC. The advantage of this

3.2 Integration of Resources

The perspective of semantic web resources gives us a different view on information integration - we no longer integrate data from a heterogenous set of data-sources but instead concentrate on using semantic web resources. For that, some basic requirements have to be fulfilled.

1. All resources have to be *identified using a URI* [4]. The URI can either be generated automaticall or by an algorithm (based on some metadata of the resource, ie location). In the latter case, different applications accessing the same resource have to employ the same algorithm to create the URI.

2. All structured data is accessible through the *Resource Description Framework* (RDF)[2], which provides a common framework for expressing semantic information so it can be exchanged between applications without loss of meaning. RDF is based on the idea of describing resources in terms of properties and property values.

3. The represented RDF information has to comply to *ontologies*. The semantic meaning of the data is described in ontologies using RDF Schema or OWL, which allows application integration in a more effective way [11].

These three rules are - *in theory* - very simple and are based on existing work. We require that all services describe their managed data using ontologies expressed as RDF-S [5] or OWL [13] descriptions. For most desktop data, these ontologies already exist and can be retrieved from web-sites like *schemaweb.info*[3]. Data from desktop applications can be represented accordingly. For instance email messages can be described using the EMiR ontology[4]. Our group has shown the benefits from using ontologies in an e-learning scenario [2]. Others call Ontologies the *Silver Bullet for Knowledge Management and Electronic Commerce*[11]. We are now facilitating ontologies to integrate desktop applications, the details about using ontologies on the Semantic Desktop can be found in [15].

[2] http://www.w3.org/RDF/
[3] http://www.schemaweb.info
[4] http://xmlns.filsa.org/emir/

The use of above three rules - *in practice* - leaves many issues open, as we can show in a simple example: to identify a local file, a standard way of generating the URI still has to be found - it is operating system dependent, and implementation dependent. One system might use `file://c/test.txt` while the other comes to `file://c/:/test.txt`. When files are renamed or their contents changes, it is not straightforward to create a URI that does not change.

3.3 Desktop Integration Based on Semantic Web Protocols

Inter-application-communication is an important part in integration systems. Applications have to communicate with each other to extract information and to update information. Common protocols to do this are DCOM on the Windows platform, SQL, CORBA, or SOAP.

For the Semantic Web, there is no agreed protocol to communicate, ongoing work about this topic is collected by the *Data Access Working Group*[5] of the W3C. Defining Web Service standards is also part of the SWWS[6]. There are many simple and promising approaches like URIQA[7] that may outrun the big projects. In the gnowsis project, we decided to use a Java based web server that provides access to the different services through our own protocols and URIQA.

Opening a web server on every desktop computer creates questions about *trust and security*. Users will be able to define fine-grained access rules for their resources. It should be possible to allow other people and social groups access to resources. The W3C is still working on the trust and security standards for the Semantic Web. When these standards are available, we plan to transfer them to the desktop.

These are the basic rules that are the foundation of our work. Implementing these rules may help in any project that builds on Semantic Web technology and they surely help integrating software.

4 The Gnowsis Project

Gnowsis is an Open Source project released under a BSD compatible license. It is created to prove the ideas of the Semantic Desktop and to have a reference implementation at hand for other researchers to test the system and build on it. The project is hosted and documented at `http://www.gnowsis.org`. It is Java based and runs on Windows, Linux and MacOs X. During development of the project, we learned many things about building real Semantic Web applications. Presentations of the System gave us feedback to improve. Our company, the DFKI research center, uses Gnowsis in several research projects.

[5] http://www.w3.org/2001/sw/DataAccess/
[6] http://swws.semanticweb.org
[7] http://swdev.nokia.com/uriqa/URIQA.html

The gnowsis project is also about building a community of researchers and developers that are engaged in Information Management projects with Semantic Web technologies. The gnowsis website hosts a Semantic Desktop forum. We met with other researchers on conferences to discuss the topic, with the project at hand as a good support for our arguments.

5 Application Areas

Based on the introduced principles and our experience from the gnowsis prototype we gathered information about the consequences that the use of the Semantic Desktop has. In this section, we will describe the consequences of such a Semantic Desktop for knowledge appliances.

5.1 Streamlining Integration

Today, application integration is handled through different communication protocols and a plethora of data formats. For instance let us take the task to build a software agent that works with information about persons.

Example 1. A Person Agent. The agent is written in Java and accesses the Outlook address book to extract the acquaintances of the user. Building such an agent involves many steps. First, identify which address book software is used. If it is MS-Outlook then access to the data is through ActiveX or ADODB, standards that are per se not accessible in Java and require 3rd party packages. Then the data structures inside of Outlook have to be known. These data structures are described in an *online help* file.

In a Semantic Desktop project, the engineer has to cope with only one protocol, a Semantic Web protocol and with only one data format, RDF. The semantic meaning of the data is shipped with the system as ontology. To query information about all persons on a Semantic Desktop, the developer can run a Semantic Web query "return all local instances of class Person". The result to this query is returned in RDF 3.2. The semantic concept of *a person* as a human being is already expressed in an ontology. In Figure 1 we see a comparison of involved technologies.

This lowers the cost to build agents that integrate many applications. Because engineers do not have to know several protocols and data standards, agents can be created faster and cheaper.

5.2 Knowledge Management

Van Elst et. al. describe the role of *information technology* as an enabling factor for knowledge management[10]. In the same article, they describe a *Distributed*

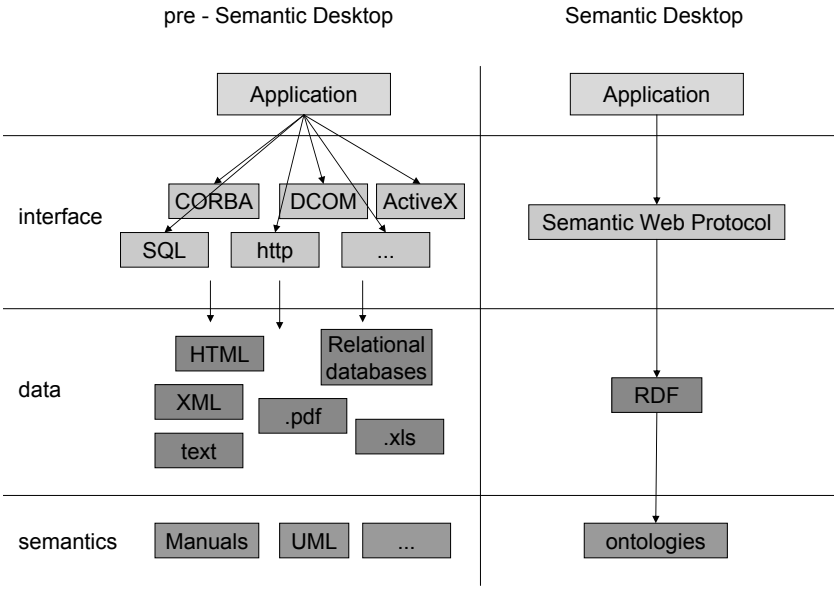

Figure 1: Integration Effort Comparison

Organizational Memory (DOM) as the next logical step after building an *Organizational Memory* as previously described in [1]. One of the characteristics of Knowledge Management is that "KM has to respect the distributed nature of knowledge in organizations" [10].

The desktop and mobile computers of knowledge workers are therefore part of a Distributed Organizational Memory. Not only as clients, but also as Semantic Desktop Servers, that allow the workers to share their knowledge. Knowledge management applications can be built on top of a network of Semantic Desktops.

In the gnowsis prototype, we have build a knowledge management system consisting of a *diary and an idea management software* (a weblog/wiki combination), inspired by many predecessors [6], [12]. The advantage of building such a system on the Semantic Desktop ground is that any resources can be included into the daily information work.

Cayzer already advocates the use of rich metadata in weblogs to create an "decentralized, informal knowledge management"[7]. On the semantic desktop, we can integrate existing desktop resource into the weblog entries. If people or events are cited in the weblog, they are automatically linked to the corresponding desktop resources from the address book or calendar. The connection between the weblog entry and the desktop resource is then available both from

the weblog and from the desktop application (the address book), allowing the user to navigate between these. The same functionality is also transfered to the wiki. Previous and ongoing work in this topic [9], [14] indicates that this approach eases the use of Semantic Web technology in practice. We enhanced an existing wiki to extract RDF information out of the links that are included in a wiki text. Thus, the information hidden in such sentences as "John was an attendee at the ConferenceMeeting" could be extracted and the noted person (John) could be linked to a resource (the ConferenceMeeting) using a qualified relation (attending).

But the use cases can be extended into another direction also, of monitoring knowledge management rules and expressing them. If a company has a policy that requires unused files to be moved in an archive, this policy can now also be implemented on emails, people, projects, etc. Knowledge management tools can concentrate on the policies and do not have to interact with native file formats.

5.3 User Observation and User Context

One of the goals of the ongoing EPOS project of the DFKI is gathering knowledge about the user's current context by observing the interaction of the user with the system. This is described in [16]. The envisioned system will observe the users work as well as his ways of information handling and automatically learn and identify his goals, intentions, structures, ontologies, and work processes. Towards the user, a sophisticated knowledge workspace shall act as an adaptive assistant proposing follow-up working steps and providing (how-to) information as well as relevant documents. In order to do so, the assistant needs to know about the users current context.

Building such an assistant would require many man-years of work. The observation mechanisms have to be included into every application. The goals, intentions, structures and ontologies would be hard to extract from the plethora of normal office resources. But when all documents are already described in detail within Semantic Web ontologies, this work is not required. If the knowledge workspace is already implemented by a KM tool like mentioned above 5.2, then the user's goals, structures and work processes are also accessible out of the box. The cost of developing such projects is greatly reduced by building on Semantic Desktop technology.

6 Summary and Outlook

The Semantic Web is still at the beginning but we think the Semantic Desktop will be an enabling technology. With it, users can benefit from the technology today. Building software that assists users in their daily information work is eased using streamlined integration mechanics.

To prove the capabilities of the Semantic Desktop, we work on the gnowsis open source framework, where we are implementing and testing the discussed ideas. We still wait for decisions by the World Wide Web Consortium on Semantic Web standards for querying and protocol. When these are mature, we will implement them and the Semantic Desktop may be the basis for future information integration.

References

1. A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3):40–48, 1998.
2. S. Baumann, A. Dengel, M. Junker, and T. Kieninger. Combining ontologies and document retrieval techniques. In *3rd International Workshop on Theory and Applications of Knowledge Management (TAKMA 2002)*, in conjunction with DEXA 2002, Aix-en-Provence, France, September 2002.
3. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 89, May 2001.
4. Tim Berners-Lee and L. Masinter R. Fielding. Rfc 2396: Uniform resource identifiers (uri): Generic syntax. http://www.ietf.org/rfc/rfc2396.txt, 1998.
5. D. Brickley and R.V. Guha. Rdf vocabulary description language 1.0: Rdf schema. w3c recommendation 10 february 2004. http://www.w3.org/TR/rdf-schema/.
6. Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):p101–108, July 1945.
7. Steve Cayzer. Semantic blogging and decentralized knowledge management. *Commun. ACM*, 47(12):47–52, 2004.
8. David Huynh Dennis Quan and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In *International Semantic Web Conference*, pages 738–753, 2003.
9. B.R. Djaloeis. Kaukolu: Building a semantic wiki. Master's thesis, University of Kaiserslautern, 2005.
10. Ludger van Elst, Andreas Abecker, Ansgar Bernardi, Andreas Lauer, Heiko Maus, and Sven Schwarz. An agent-based framework for distributed organizational memories. In M. Bichler, C. Holtmann, S. Kirn, J. P. Mller, and C. Weinhardt, editors, *Coordination and Agent Technology in Value Networks, Multikonferenz Wirtschaftsinformatik (MKWI-2004), 9.-11.3.2004, Essen*, pages 181–196. GITO-Verlag, Berlin, 2004.
11. D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, 2001.
12. Eric Freeman and David Gelernter. Lifestreams: A storage model for personal data. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 25(1):pp80, 1996.
13. D. L. McGuinness and F. Harmelen. Owl web ontology language overview w3c recommendation 10 february 2004. http://www.w3.org/TR/owl-features/.
14. Paolo Castagna Roberto Tazzoli and Stefano Emilio Campanini. Towards a semantic wiki wiki web. Technical report, Demo at the ISWC2004, 2004.
15. Leo Sauermann. The gnowsis-using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna, 2003.
16. Sven Schwarz and Thomas Roth-Berghofer. Towards goal elicitation by user observation. In *Proceedings of the FGWM 2003 Workshop on Knowledge and Experience Management*, Karlsruhe, 2003.
17. York Sure. Perspective on semantic web technologies. *Proceedings of I-KNOW*, 2004.