

The Gnowsis Semantic Desktop for Information Integration

Leo Sauermann

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern, Germany
and
Knowledge-Based Systems Group, Department of Computer Science,
University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern
`leo.sauermann@dfki.de`

Abstract. To integrate office appliances, there are different standards, consisting of data formats and communication protocols. The WWW and *Semantic Web* standards are already designed for worldwide integration and can be transferred to office integration. We present our vision of the *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. The architecture is based on a Semantic Web Server running as Desktop service. Existing desktop applications (email client, browser, office applications) are integrated. The semantic glue between them is expressed with ontologies. This architecture will enable us to create tools for information management faster and cheaper. Based on the local Semantic Desktop Servers, teams of knowledge workers can set up *peer-to-peer* connections. *Distributed Organisational Memories* can be based on Semantic Desktops. The *gnowsis* framework is an open source project led by the DFKI that realizes parts of this vision. Gnowsis was used to test our ideas and allow others to experiment.

1 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 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* are now shipped with XML support and 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*. A translation of the Semantic Web to the topic of Desktop information systems is the next step.

In this paper we present the principles of the *Semantic Desktop* and how they can help in system architecture and information architecture. We took ideas from building web services and translated them to the desktop scenario. This allows us to treat desktop information management with the same approach as web-based information management. Existing systems can be coupled with a Semantic Desktop.

First, we will give a short introduction to the Semantic Web Framework. Then we will present the principles we envision for the Semantic Desktop on. Following is a description of the open source framework *gnowsis* that we created to test the principles. 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 a conclusion.

1.1 Semantic Web

Briefly, the Semantic Web is *an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.* [15].

It is a vision of an improved kind of Web with enhanced functionality which will permit semantic-based representation and processing of Web information. The *World Wide Web Consortium* (W3C) has proposed a series of technologies that can be applied to achieve this vision. The *Resource Description Framework* (RDF)¹ provides a common framework for expressing the semantic information so it can be exchanged between applications without loss of meaning. RDF is based on the idea of identifying things using Web identifiers (URIs), and describing resources in terms of properties and property values.

2 Semantic Desktop

In 2003 the Semantic Desktop idea was first introduced in a master thesis [12]. In this section we will describe the motivation for the Semantic Desktop and the principles. In the following section we will discuss the implications for *Intelligent Office Appliances*.

The data accessed by knowledge workers comes from different, existing applications. Emails, documents, contact information (address books) and calendar

¹ <http://www.w3.org/RDF/>

information are such obvious data sources. 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.

2.1 Everything is a Web Resource

The first step to a Semantic Desktop is a change of philosophy. Today the view is that files are stored on desktop computers. In the Semantic Desktop view, we can see the same files as web documents stored on a web server that is by coincidence also the personal computer (PC) of a knowledge worker. Seen through the Semantic Web glasses resources could be 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 resource, as every email, photo, address book entry, and all other information we find on a typical PC.

This is a mental step that we have to make: all digital information items stored on a PC can be seen as web resources. Based on this philosophy, we can use the techniques of the *Resource Description Framework (RDF)* on the resources. The next steps will give more detail.

2.2 A Local Http Server

The focal point of the Semantic Desktop integration is a local http server. This server has two general tasks. First, it hosts several web applications that publish the data and functions of the different applications. Second, software agents can contact the server to access the data of these applications.

2.3 Resources are Identified by URLs

One of the pillars of the WWW is the principle of the *Uniform Resource Identifier* – the URI [14]. First they were used to identify resources on the web. But they can also be used to identify anything. For instance, ISBN numbers (the numbers used to identify books) can be written as URIs. We will focus now on a subset of URIs called Uniform Resource Locator (URL). These are expressed in this form:

Example 1. A typical URL
[protocol] : [server] / [path]

An example would be an URL to a document in an intranet system. In example 2 we see the identifier (and locator) of a document. From reading parts of it, we can infer that this may be the contract of projectX of a company.

Example 2. An intranet resource

```
http://intranet.test.com/documents/projectX/contract.pdf
```

For the identification of local Semantic Desktop resources, URLs can be used. As the *http* protocol can be used to access local resources most URLs will therefore begin with the *http* protocol. The server in the URL will be the hostname of the localhost. Normally these hostnames are cryptic names given by network administrators. We prefer to name the computers in reference to their users, so "johndoe.acme.com" may be the hostname of John Doe's computer. The different kinds of documents and resources are then expressed in the path of the URL. For files, the filename and path can be part of a URL, for other resources there are always unique properties of a resource that work. The mapping of a resource into a URL can be done in different ways, one is proposed by C. Bizer in his D2RQ project [5]. We invented our own scheme that is discussed in detail in [12]. An example URL is shown below in example 3. It identifies a file from the "my documents" folder on a computer called John Doe which belongs to somebody with this name. The name of the file is cv.pdf and we assume that this is a curriculum vitae.

Example 3. A Semantic Desktop Resource

```
http://johndoe.acme.com/myDocuments/cv.pdf
```

Note that the PDF document identified by the URL in example 3 can be accessed through the local *http* server as described in Section 2.2.

2.4 All Structured Data is Accessible as RDF

XML as a data format has already been a success in the field of application integration. RDF is a language that is based on a graph representation of data and can be expressed in XML. RDF has an advantage over XML: the meaning of the data is described in ontologies using RDF Schema or OWL, which allows application integration in a more effective way [10].

For the Semantic Desktop we require that all structured data (like the columns in a database or the fields in an address book) has to be representable as RDF. For any resource identified with an URI such an RDF representation can be obtained. For the CV of John Doe (Example 3) the RDF document would contain author, date of creation, keywords, file type and other data. See the following sections how to access the data.

2.5 Desktop Communication Based on Semantic Web Protocols

Inter-process-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. These are standards that enable us to build integrators.

For the Semantic Web, we still miss a general standard that can take the role of the protocol. There is ongoing research on this topic, a standardization group is the *Data Access Working Group*² of the W3C. Defining Web Service standards is also part of the SWWS³. There are many simple and promising approaches like URIQA⁴ that may outrun the big projects. We decided to use many ideas from URIQA to implement our prototype and wait for the outcome of the general discussion for a final protocol. But whatever protocol becomes the standard, it will provide better means for application integration that SOAP and CORBA offer today.

Semantic Web protocols will be used locally on the Semantic Desktop as a communication means for inter-process-communication.

2.6 Data is Described with Ontologies

As mentioned in 2.4, the RDF data representation allows us to add semantic meaning to our data. Our group has shown the benefits from using ontologies in an e-learning scenario [3]. Others call Ontologies the *Silver Bullet for Knowledge Management and Electronic Commerce*[10]. We are now facilitating ontologies to integrate desktop applications.

We require that all services describe their managed data using ontologies expressed as RDF-S [6] or OWL [7] descriptions. For most desktop data, these ontologies already exist and can be retrieved from web-sites like *schemaweb*⁵. Data from desktop applications can be represented in a vocabulary that one of these public ontologies describes. For instance email messages can be described using the EMiR ontology⁶. Metadata of John Doe's CV (Example 3) would be described using the Dublin Core ontology [1]. More details about the use of ontologies on the Semantic Desktop can be found in [12].

2.7 Security

Opening a web server on every desktop computer creates questions about security. 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.

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. How to realize this is still under evaluation.

² <http://www.w3.org/2001/sw/DataAccess/>

³ <http://swws.semanticweb.org>

⁴ <http://swdev.nokia.com/uriqa/URIQA.html>

⁵ <http://www.schemaweb.info>

⁶ <http://xmlns.filsa.org/emir/>

3 The Gnowsis Project as Reference Implementation

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 DFKI uses Gnowsis in several research projects. The project is hosted and documented at <http://www.gnowsis.org>. In September 2004 we released an alpha version. The framework is Java based and works 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.

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 also hosts a Semantic Desktop forum.

4 Consequences

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.

4.1 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 4. 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. The semantics of what a person is and what can be done with persons is not known.

In a Semantic Desktop project, the engineer has to cope with only one protocol, a Semantic Web protocol 2.5 and with only one data format, RDF 2.4. The description of the data format and the semantic is delivered with the data as ontology. To query information about all persons on a Semantic Desktop, the developer asks a Semantic Web query "return all local instances of class Person" 2.5. The result to this query is returned in RDF 2.4. The semantic concept of *a person* as a human being is already expressed in an ontology 2.6. In Figure 1 we see a comparison of involved technologies.

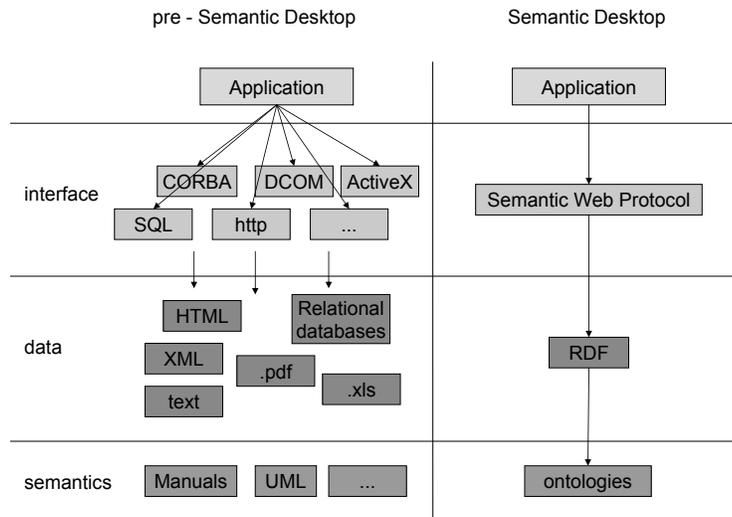


Fig. 1. Integration Effort Comparison

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

4.2 Peer-To-Peer Scenario

Peer-to-peer is a buzzword that has been stressed in recent years. The simple core of it is that several computers or applications access each other in a distributed way, trying to avoid central servers. The beginning of *peer-to-peer* was marked with very popular tools for file sharing like *Napster* or *Kazaa*. Today we can share our music in peer-to-peer systems using *Apple's iTunes* peer-to-peer functions. But all of these applications are internally based on web servers. *iTunes*, *Kazaa* and *Napster* publish their files using a modified http protocol in combination with specialized web services for distributed search. They also identify files with URLs when file transfers are initiated.

Based on the commercial social and popularity of these http-based peer-to-peer systems, it is obvious to say that this architecture is capable of connecting knowledge workers as well. The principles above 2.2 and 2.3 are a prerequisite for peer-to-peer functionality. With 2.5 and 2.6 a search functionality is given. And all these functions are at the heart of desktop computers.

The Semantic Desktop is therefore a stable platform for peer-to-peer networks. Simple and stable technology, proven in the commercial world to be scalable and effective, justifies our commitment to it.

4.3 Knowledge Management

Van Elst et. al. describe the role of *information technology* as an enabling factor for knowledge management [9]. In the same article, they describe a *Distributed Organizational Memory* (DOM) as the next logical step after building an *Organizational Memory* as previously described in [2]. One of the characteristics of Knowledge Management is that "KM has to respect the distributed nature of knowledge in organizations" [9].

The desktop and mobile computers of knowledge workers should be 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 the Semantic Desktop.

In the gnows prototype, we have build a knowledge management system consisting of a diary and an idea management software (a weblog/wiki). It is based on many predecessors [4], [11].

The advantage of building such a system on the Semantic Desktop ground is that any resources can be included. Image a mind mapping tool that allows you to draw maps of your information and your ideas. But when you put a project, a person or an email into the mind map, the element of the mind map will be linked to real resources. People from an address book, emails, projects can all be added to the system. If a company has a knowledge management 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.

4.4 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 [13]. 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 4.3, 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.

5 Summary and Outlook

We show the consequent use of Semantic Web standards on desktop computers. Taking this approach, we can open many possibilities for appliance integration and knowledge management.

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.

Today we have millions of web servers already running on the web. The information industry as a whole has experience with integration of web servers. Using web technology will simplify the integration development of desktop applications.

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. The dublin core metadata initiative. <http://dublincore.org/documents/dcmi-terms/>.
2. Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn, and Michael Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3):40–48, 1998.
3. Stephan Baumann, Andreas Dengel, Markus Junker, and Thomas 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. to appear.
4. Vannevar Bush. As we may think. *The Atlantic Monthly*, 176(1):p101–108, July 1945.
5. A. Seaborne C. Bizer. D2rq treating non-rdf databases as virtual rdf graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*, 2004.
6. R.V. Guha D. Brickley. Rdf vocabulary description language 1.0: Rdf schema. w3c recommendation 10 february 2004. <http://www.w3.org/TR/rdf-schema/>.
7. F. Harmelen D. L. McGuinness. Owl web ontology language overview w3c recommendation 10 february 2004. <http://www.w3.org/TR/owl-features/>.
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. 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.
10. D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, 2001.
11. 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.
12. Leo Sauermann. The gnowsis-using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna, 2003.
13. 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.
14. L. Masinter Tim Berners-Lee, R. Fielding. Rfc 2396: Uniform resource identifiers (uri): Generic syntax. <http://www.ietf.org/rfc/rfc2396.txt>, 1998.
15. Ora Lassila Tim Berners-Lee, James Hendler. The semantic web. *Scientific American*, 89, May 2001.