Overview and Outlook on the Semantic Desktop

Leo Sauermann, Ansgar Bernardi, Andreas Dengel

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern, Germany
{leo.sauermann|ansgar.bernardi|andreas.dengel}@dfki.de

Abstract. In this paper we will give an overview of the Semantic Desktop paradigm, beginning with the history of the term, a definition, current work and its relevance to knowledge management of the future. Existing applications and research results are listed and their role as building blocks of the future Semantic Desktop described. Based on the analysis of existing systems we propose two software architecture paradigms, one for the Semantic Desktop at large and another for applications running on a Semantic Desktop. A view on the context aspect of the Semantic Desktop and the Knowledge Management aspect is given. Based on the current events and projects, we give an outlook on the next steps.

1 Introduction

The Semantic Desktop will be the driving paradigm for desktop computing in the area of the Semantic Web. Based on the needs and expectations of users today the software industry will evolve to a future way of computing, semantic desktop computing.

The main task at hand is to **transfer the Semantic Web to desktop computers**, and this transfer will not only consist of the technology, but also of the philosophy and the people involved. Developers that today concentrate on services for the Semantic Web (and find tools and examples) will need a complete RDF and ontology based environment to create applications on desktop computers. End users will benefit from these applications, as they integrate and also communicate better—based on ontologies and Semantic Web standards—than today's desktop applications.

1.1 The background and goals of the Semantic Desktop community

In 1945, Vannevar Bush wrote the now famous article "As we may think" [1], where he described the visionary system called "Memex". The definition that he gave was important for many systems to follow:

> Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

Bush based his ideas solely on analog devices, running on punch cards and using microfilm as storage. Today we notice how his vision becomes reality, the personal computer is very close to what Bush had in mind. Not all books and records are stored in a PC, but we are close to it. The idea of *trails*—paths of resources that build a personal look on a topic—were taken up by system like lifestreams [2]. Still, there is work left to create the *intimate supplement to memory* - in 1960, Ted Nelson described a system called *Xanadu* in his article "*As We Will Think*" [3]. Xanadu is a predecessor of hyperlink systems, the core idea was to link information items and, in a second phase, make them tradeable as a basis of information society. Nelson also coined the term "Hypertext". Although different implementations and prototypes of the Memex were built, it never ignited the revolution that was intended by Nelson. In 1992 the World Wide Web launched, created by Tim Berners-Lee. The Web grew at a very fast rate and changed society; information is used in a different way than in the pre-web era. Before the web lifted off, Berners-Lee programmed the *Enquire-Within-Upon-Everything* system. *Enquire* was a personal information management tool to store information about people, projects, hardware resources and how they relate to each other. It was created out of a certain need:

> What I was looking for fell under the general category of documentation systems – software that allows documents to be stored and later retrieved. This was a dubious arena, however. I had seen numerous developers arrive at CERN to tout systems that "helped" people organize information. They'd say, "To use this system all you have to do is divide all your documents into four categories" or "You just have to save your data as a Word Wonderful document" or whatever. I saw one protagonist after the next shot down in flames by indignant researchers because the developers were forcing them to reorganize their work to fit the system.[4, p. 17]

These were the requirements that led to a distributed version of Enquire that we know today as the World Wide Web [5]. The interesting fact is, that the Web had its revolution in the distributed world but the topic of personal information management remained the same. The field of "documentation systems" is still a vivid arena with many competing companies. The problem of metadata and labeled links was identified and is now tackled by the Semantic Web Initiative [6].

1.2 Today's state of the Semantic Desktop idea

In 2003, facing the fact that the Semantic Web was not universally used, we analyzed the field and found that the major projects aimed at large and distributed organizations, but the end user was only supported by Haystack or Protege, which both were complicated. Nearly all information we saw on web pages and in electronic documents had been created by people using personal computers. The PC was the place where most personal data is stored and the major interface to the web. Information stored on a server was usually manipulated through interfaces that are executed on a PC, be it a web browser or web authoring tool. The use of ontologies, classifications and global identifiers in normal desktop applications did not happen. Tim Berners-Lee also realized that the end user applications were missing and requested in several talks that we start building useful applications. From this perspective we stated [7]:**If the goal is to have a global Semantic Web, one building block is a Semantic Desktop, a Web for a single user.**

The term “Semantic Desktop” itself was coined by Stefan Decker and picked up by Leo Sauermann in 2003, to create a term that creates a mutual understanding for the similar ideas. Stefan Decker and Martin Frank stated the need for a “Networked Semantic Desktop” [8] in 2004 and sketched the way to the events today. Decker recognized that several new technologies had emerged which could dramatically impact how people interact and collaborate: The Semantic Web, P2P computing, and online social networking. He presented a vision of how these different thrusts will evolve and produce the Networked Semantic Desktop, which *“enables people and communities to directly collaborate with their peers while dramatically reducing the amount of time they spend filtering and filing information”*. His roadmap to the *Networked Semantic Desktop* is laid out as follows: [8]

– In a first phase, Semantic Web, P2P, and social networking technologies are developed and deployed widely.
– In the second phase, a convergence between the existing technologies brings Semantic Web technology on the desktop leading to the Semantic Desktop. In parallel, Semantic Web and P2P are incorporated and lead to Semantic P2P. Social networking and Semantic Web lead to ontology driven social networking.
– In a third phase, the social, desktop and P2P technology fully merge to a *Social Semantic Desktop*.

Based on the previous publications [8, 7, 9] we could define a Semantic Desktop in the following way:

Definition 1. *A Semantic Desktop is a device in which an individual stores all her digital information like documents, multimedia and messages. These are interpreted as Semantic Web resources, each is identified by a Uniform Resource Identifier (URI) and all data is accessible and queryable as RDF graph. Resources from the web can be stored and authored content can be shared with others. Ontologies allow the user to express personal mental models and form the semantic glue interconnecting information and systems. Applications respect this and store, read and communicate via ontologies and Semantic Web protocols. The Semantic Desktop is an enlarged supplement to the user’s memory.*

1.3 The near future

From our point of view, we have achieved most of the goals of the first phase and are currently in the second phase. Our task is now to weave the existing and stable parts of the Semantic Web into desktop computing, P2P, and Social Networking. In this paper we will address the aspects of a single Semantic Desktop system, the role of a Semantic Desktop in a networked environment was already addressed by Decker et al. [8]. The Semantic Desktop is a global project involving researchers and industry from different technical fields.

To create a focal point for the Semantic Desktop, the European IST Project NEPOMUK was initiated by a consortium lead by the DFKI. Bringing together researcher partners from NUI Galway, EPFL Lausanne, DFKI Kaiserslautern, FZI Karlsruhe, L3S Hannover and ICCS-NTUA Athens with practitioners from companies like HP, IBM, SAP, Mandriva, Thales, PRC Group and others, this project will build a community of experts. NEPOMUK bundles academic, industrial and open source community efforts to create a new technical and methodological platform: the *Social Semantic Desktop*. It enables users to build, maintain, and employ inter-workspace relations in large scale distributed scenarios. New knowledge can be articulated in semantic structures and be connected with existing information items on the local and remote desktops. Knowledge, information items, and their metadata can be shared spontaneously without a central infrastructure. NEPOMUK will realize a freely available open-source integration framework with a set of standardized interfaces, ontologies and applications. Collaboration with the open source community and integration with major open source products is intended and will ensure the broad acceptance of NEPOMUK technology—thereby activating a sustainable open source movement with viral spread-out. A number of case studies apply, adapt, and test NEPOMUK's solutions in various knowledge-work scenarios. NEPOMUK's standardized plug-in architecture combined with usage experiences opens up manifold business opportunities for new generic or domain-specific products and services. Using the methodology that spread the World Wide Web – *open standards, open source reference implementations and continuing communication with the global developer community* (as described in [4]) – the Semantic Desktop community at large will gain momentum through this project.

2 Semantic Desktop building blocks

To provide such a system to end users, a few prerequisites are required. In this section we start describing research projects that address the topic of an integrated Semantic Desktop and then we give examples of tools that are available today as building blocks for the future Semantic Desktop. An outlook will be given on the features users can expect and the relevance to personal knowledge management.

2.1 Integrated projects

The first research project using the term was the *Gnowsis Semantic Desktop* [7] by Leo Sauermann, co-author of this paper. The work was a diploma thesis and deals with the details of integrating desktop data sources into a unified RDF graph, also addressing the problem of how to identify resources with URIs. You will find an introduction to the field in this work and a prototypical user interface, introducing the terms "link and browse" as a desktop metaphor. The project is now continued both as an open-source project and is reused framework for other research projects, namely *EPOS* [10] and *@Visor*.

Similar to the gnowsis work, but on the web-services world is the *SECO: mediation services for semantic Web data* project aiming at integrating web sources [11]. It describes an infrastructure that lets agents uniformly access data that is potentially scattered across the Web. The results can be transformed to the desktop, as we have done in another paper submitted to ISWC [12]. In the field of data integration, also the architecture by Bizer and Seaborne [13] about adapting SQL sources should be mentioned. A product by the Microsoft corporation called *Information Bridge Framework* [14] aims in the same direction for conventional data sources: they can be included into office documents via so called *SmartTags*. The framework implements a client–server based approach, the server provides a metadata service that integrated several enterprise web services and other data sources (like CRM systems). The client can be normal office applications, that are extended by plugins: a client gathers current context and keywords from open documents and loads related information from the server.

A view on the Semantic Desktop was given by Stefan Decker and Martin Frank in 2004, their paper called “The social semantic desktop” [8]. It focuses on the technology threads that are available and have to be combined to create the Semantic Desktop. The need for the system and the solutions it will provide are outlined. A possible roadmap is drawn, as mentioned above.

A major research project concerning an integrated approach in our field is the *Haystack* system by Quan et al. [15] from the MIT Computer Science and Artificial Intelligence Laboratory. It is an integrated approach to let an individual manage her information in a way that makes the most sense to her. It is a replacement for many applications including word-processors, email clients, image manipulation, instant messaging and other functionality. They provide a complete semantic programming environment, from user interface to database. One disadvantage was that the prototype system had performance problems in 2003. These have been identified and addressed in the upcoming *Hayloft* project.

MyLifeBits by Microsoft Research is a lifetime store of multimedia data, based on the assumption that all information a single person reads and hears can soon be stored on a portable device. Every day a person consumes audio, video, text and other media. If a hypothetical disk of one terabyte per year is available, it would be possible to store all this multimedia on it. The MyLifeBits paper describes a concept how to manage this huge amount of media, how to classify and retrieve the data [16].

Ontooffice by ontoprise—a corporation close to semantic web research—is a desktop product that brings together the contents of a semantic web server and Microsoft Office applications. The scenarios are similar to those of SmartTags and the Information Bridge Framework.

From the **open source scene**, several projects are aiming at a semantic desktop environment, one such a project is *Chandler* managed by the osa-foundation and lead by Mitch Kapor (who designed Lotus Agenda). It is a Personal Information Manager (PIM) intended for tasks like composing and reading email, managing an appointment calendar and keeping a contact list. It simplifies information sharing with others, and calls itself an *Interpersonal Information Manager*.

The *Fenfire* [17] project is at an earlier stage, dealing with the problem of visualising and editing RDF graphs in a uniform way. It is a completely based on RDF and implements various user interface metaphors. Parts of the system are published, others are kept closed because of patent issues.

Another approach was taken by Joe Geldart in his bachelor thesis about the *frege* system [18]. He describes a minimal implementation of an RDF desktop communication framework on which a few example applications are implemented. The thesis tackles the core ideas and finds a minimal and efficient solution.

From these numerous examples, which only give starting points for the interested reader, we see that the field of the Semantic Desktop is already advanced and that different – sometimes competitive, other times complementary – approaches exist.

2.2 Tools

The active community produced a variety of tools that are used in the projects or that are end user applications. We will now categorize these tools based on a scheme similar to one developed for [19] and build a table that gives and overview, see table 1. Two main categories are assumed—first are the *grounding technologies*, the basic building blocks of system technologies and Semantic Web technologies. Second are *information interaction* tools providing users with interfaces to author and browse information. A third category are *ontologies* and ontology related tools.

The grounding technologies consist of storage, search and communication facilities. Storage and search are repositories that hold RDF and ontology data in a persistent way and to allow semantic search or fulltext search on the data. Known projects here are *Jena* [20], *Kowari* [21], *RDF Gateway* [22], or *Sesame* [23]. The support for full-text search is sometimes a feature of the repositories, if not it can be implemented through projects like *Apache Lucene*. For the Semantic Desktop we face several problems with repositories. First, multiple incompatible interfaces are implemented by the systems; therefore state that we need standardized interfaces for storage servers. The upcoming SPARQL standard [24, 25] will provide us with these. Although these repositories are in common use, they are far away from perfection. A description of problems with performance and ease of use can be found in the YARS project description by Harth et al. [26]; they tested the read and write performance of common open source RDF repositories and found major deficiencies. One store was not installable at all. *Communication technology* needed to receive and send messages is today provided by e-mail, instant messaging and peer-to-peer systems. It is possible to use these technologies to send semantic messages, as shown in [27] for semantic email or for the Jabber protocol (a standard for instant messaging) in the *Nabu* project [28], a semantically enhanced Jabber server. On the Semantic Desktop, these existing communication ways will be used to send semantic messages.

The shown storage, search and communication technologies will be used to store and communicate data that is expressed using *ontologies*. Users will work with several ontologies and the information expressed in these ontologies will come from heterogenous sources. A crucial factor will be the integration of ontologies by ontology mapping. Common ontologies we find on desktop computers today are Dublin Core, FOAF, iCalendar and more. We expect that through diversification and selection (an evolutionary, community process) a combination of many popular ontologies will be used on the future Semantic Desktop.

The user experience will be determined by *information interaction* software. Common applications here are ontology editors, domain specific applications, browsers and personal knowledge management tools. *Protégé* by Stanford Medical Informatics is a popular ontology editor, *PhotoStuff* by the Mindswap group is a photo annotation tool. *Tidepool* is another photo editing tool, with a commercial background; together with the website *Storymill.com* users can annotate and publish photos. The many RSS readers that are available today can be seen as domain specific applications—they focus on news and information syndication. On the personal information management side, we find *Microsoft Outlook* or *Lotus Notes* in many companies. Another good example is FRODO Taskman [29] which realizes a fully RDF based semantic workflow engine. We expect semantic personal knowledge management tools in the future, that can integrate heterogenous sources taken from the Semantic Desktop.

3 How to build a Semantic Desktop

In this section we will describe how the parts for a Semantic Desktop can be assembled together and what new features have to be implemented. Starting with the new requirements that come with the Semantic Desktop and how these requirements can be fulfilled, we then move on to well known features that are already implemented. But before we go into details, we have to step back and take a look at the way people think and express their mental models, so that we understand how the Semantic Desktop can support this.

3.1 Respect personal mental models

Because we do not perceive our environment as a continuum without any intrinsic boundaries, we categorize documents as belonging to named classes with certain inherent properties. We can verify this by an experiment where a number of persons should categorize a new computer science book or journal article into, e.g., the ACM Computing Classification System (CCS) [30].

Now, let us transfer this idea to the Semantic Desktop where we generate, receive and organize documents. Because of the nature of our brain to classify and store (and perhaps a *hunter-gatherer* mentality), we populate our workspace (and websites, corporate fileshares, etc.) with documents needed to satisfy the daily requirements of our work. This leads to the thesis that all documents which are available on our individual workstation are somehow related to our **individual background**, to the ongoing tasks and running processes we are involved, or to our personal interests. Further, the documents capture information about concepts we make of the world: persons, places, projects, topics, etc. These concepts are highly subjective but can be expressed using basic application features like the filesystem's folder structure or enhanced formalizations like OWL ontologies or taxonomies. Documents can be classified using these structures, manually by the user who decides how to classify a document at hand by reading it, understanding it and correlating it to a mental model or automated by using text classifying engines like "brainfiler" [31, 32] or GATE [33]. Hence there exists an interaction between mental models and formal ontologies, mental models find their match in the formal, symbolic representation of ontologies.

Although the directories at individual workspaces are highly subjective, we take into consideration that collaborators usually have a **common background**. In [34] it is shown how a shared background and an awareness of a coworker's activities and mental states contribute to establishing and maintaining communication. This common background has to be expressed using a formalization that addresses the similarities among participating collaborators. If the participants work in a similar topic, then the common background of ie "biology" may be available in a public ontology, expressed by domain experts, preferably formalized in OWL. Using them allows a sender to describe a message in a category that the receiver will understand, because the same category exists on both computers.

Hence the individual background is expressed using personal mental models, expressed as **personal concepts**; and the common background is represented by **common ontologies**. Both are formalized in RDF and preferably OWL and are used by the desktop application.

When people use computers to write down information, this information is never new. It is always created in a certain context, the individual and common background. As it is a mixture of existing information and a few new ideas, the Semantic Desktop should provide an environment where users can express new ideas and easily (preferably automatically) connect it to both personal concepts and common ontologies. We can call the background information that lead to the creation of the information resource X the **context** of the resource X. Respecting the *context of a resource* is a key feature of the Semantic Desktop. What is the user doing, what was the user doing in the last hour, day, year; what are topics relevant to the peers and the company of the user; and much more can be used to capture this context.

We also see that the **context may switch**: while most of the work of a user is around topic X (for example a project) there may be a certain time during the day (for example around noon) when the user switches to another context Y (that may be: what am I going to eat?). These context switches have to be detected and can be used. The goal of this proactive, context-sensitive assistance is that the user can keep on working as usual and the machine observes the actions of the user, automatically clustering and structuring the information at hand. Then, the system becomes a supplement to the memory of the user by doing some of the knowledge management work. Another aspect is, that the context capturing and context use is application independent. The problem Tim Berners-Lee describes should now be solved: "I saw one protagonist after the next shot down in flames by indignant researchers because the developers were forcing them to reorganize their work to fit the system" [4]. The Semantic Desktop is application-independent. The software doesn't force the user to adapt and instead adapts to the user and not only that—it also adapts to other software employed by the user.

Respecting the personal mental models can be summarized as: **do not assume one application alone representing the ideas of the user, but manage the personal concepts of the user in cooperation with other applications.**

3.2 Context and user observation solutions

The main challenge for context representation and reuse of context is the definition of a context model ontology for the personal knowledge management domain. In [35]

Schwarz explains a pro-active, context-sensitive assistance system to aid the user during her knowledge work, which is mostly about searching, reading, creating, and archiving of documents. This system was built as a research prototype in the EPOS project. Focus was to avoid distracting the user, therefore context gathering is realized by installable user observation plugins for standard applications such as Mozilla Firefox and Thunderbird.

The group around Wolfgang Nejdl published a paper on “Activity Based Metadata for Semantic Desktop Search” by Chirita et al. [36] describing a detailed ontology to represent the contextual information about several user activities, tested in a prototypical implementation. Relevant to context are e-mails and the way attachments are handled, the file hierarchy and how it resembles the users view of the world and the web browsing behavior of users. They propose an architecture to capture these contextual elements by metadata generators. The benefit for the user is that the context is used to enrich search results in desktop search. A practical implementation of this and other ideas is shown in the *Beagle++* prototype.

Another approach currently under evaluation at the DFKI in the eFisk project [37],[38] is to capture the reading behavior of the user with an eye-tracker. Using this technology, it is possible to capture on which parts of the screen the user is looking for how long. Combined with the currently displayed text, the system can recognize that the user looked a certain amount of time at a certain text. So we can assume that the text has been read and set metadata to value this text higher – during searching, we can rank read passages higher. This adds more information to the personal mental model of the user.

There are more projects aiming at capturing context information and representing it. We expect to see a common ontology for context information in the next years, that could connect these different approaches.

3.3 Searching the Semantic Desktop

Barreau and Nardi [39] analyzed the searching behavior on desktop computers in 1995 and identified two different search strategies when users are looking for information that is stored on their desktop computers: first, a path search is done, looking into folders and directories that could contain the document. If the path search does not succeed, fulltext search strategies are used. Today, desktop search engines are a major market and tools like *Google Desktop*, *Apple Spotlight*, *Yahoo! Desktop Search* or *Microsoft Windows Desktop Search* are products in a competitive market. The features provided in these free tools are satisfying to most users but far behind what is state of the art in commercial tools like *Autonomy* or *Convera* do on a company level and what is proposed in current research papers.

We expect that sophisticated information retrieval techniques will find their way to the Semantic Desktop. In fig. 1 we show a few technologies that are expected to be available. Starting from today’s fulltext search on the top-left we identify three directions towards Semantic Desktop search. In dark grey, to the top-right, text based technologies are listed. In light grey, to the lower-left, we see semantic technologies that benefit from metadata and explicit links between information items. These approaches are developed by Nejdl et al. [36, 40] and other researchers. Central are ontologies and

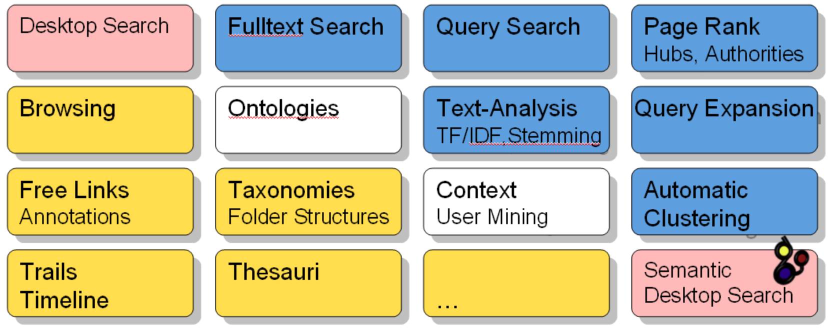

Fig. 1. Toward Semantic Desktop Search

context based approaches. In the figure we list some examples how to improve desktop search, but there are surely more ideas that contribute to the field. We intentionally left one field blank to represent the missing ideas. At the end, the combination of the listed technologies will improve the way users search, find, and experience information.

3.4 User interface

Looking at the building blocks mentioned above, we find similarities in the users interfaces and in the architecture that the software is build upon. We abstract now from the concrete examples and describe patterns we found in the user interfaces and architectures that are used today.

A typical interaction sequence in such an application is as follows:

- User searches and finds the information of interest using search services or by opening known resources via a path and confirms to edit/view it.
- Remote or local repository is contacted for the data. It usually drills down to one RDF graph and one current resource to view and edit.
- Additional data from ontologies is loaded to understand the data. Inference engines are used to augment the loaded information.
- Related information is gathered, using the loaded graph and the current resource as a starting point. Related information comes from remote and local repositories. Ontologies, thesauri, text similarity, and context are used to find related information.
- User browses information and makes decisions. New facts are entered and the personal mental model changes.
- User stores changed information to a local or remote repository

This program workflow itself is simple, and simplicity is a key feature of useful software. Systems that went beyond the simple workflow faced problems of complexity. For example, the *gnowsis* system started as a mixture of database, inference engine, user interface, and data integration architecture. The high goals of gnowsis lead to a complex architecture and performance problems which again forced us to refactor the project and split it into reusable components (a process that is not finished yet). *Haystack* also consists of database, user interface and domain specific (email, instant messaging, picture editing) functions. *Haystack* offers useful features and is a well administered project, but the demands on computing power, memory and disk storage are high. Also, users faced with such complex systems need a long training time to understand the system and benefit from them.

Protégé gives an example of a clean architecture: provide a fast, extensible user interface for ontology editing and leave storage and inference to plugins and external services.

The following description gives a rough image what a typical Semantic Desktop application of today looks like. We expect totally new interaction models for the future that extend this model, as already the example applications extend the model in different ways. Visual examples are given in fig. 3. As a reference we took these applications: Mindraider, Gnowsis, Aduna Autofocus, Haystack, PhotoStuff, Protégé, Personal Brain (thebrain.com), Windows Vista.

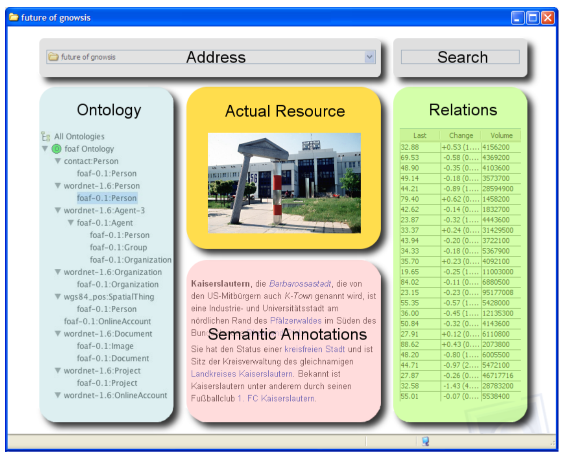

Fig. 2. a typical Semantic Desktop application user interface

We propose that the core parts of a user interface and application for information interaction are (see figure 2):

– An **adress bar** comparable to that of a web browser, where the user can easily enter the URI of the resource she wants to edit. Optionally, the address bar may also contain the address of a model/RDF graph that is currently edited.
– A single **fulltext search field** allowing searching for a resource like it is provided in Aduna Autofocus. Users expect that a plain text search field allow then to search on all possible resources and will, according to Nielsen [41], also demand such a search field.
– An visual area representing the **currently selected resource**. This is usually the center component and receives the focus of the user during editing. Visual feedback (color, font, etc.) about the currently selected resource is needed here.
– An area to add and change **annotations** of the currently selected resource. It may be part of the last point or a separate editor window. Such a component can be found,

e.g., in gnowsis or mindraider and will be provided in windows vista. Possible metaphors for it are wiki-like editors or forms.

– Additional **relations** of the current resource are also part of the user interface. These relations are often inferred based on factors like text similarity, related time or explicit links. Examples are given in mindraider, gnowsis, haystack or personal brain. They can help to ensure that all kind of information about a single resource can be presented to the user within a single window.
– An embedded **ontology browser** is also required. Respect that the ontologies are shared between applications and show the both the **personal concepts** and the **common ontologies** (as mentioned above). Users need the ability to relate the currently selected resource to the ontologies.

Because this kind of application would be monopolizing the user's attention, e.g. like an email programm, it would be best used in full screen mode. That leads to the conclusion that it has to be a sovereign posture program [42]. This is also enforced by the fact that such an application would be used very often and therefore dominates the users attention as a primary tool. The fact of having an sovereign posture points out that a semantic desktop application has to be designed for optimal use by perpetual intermediates (see axiom in [42, chapter 8]).

For future Semantic Applications, users will expect that the experience is similar to existing applications. An overview of existing applications is given in Figure 3. Based on the expectations of users we recommend: *when building Semantic Desktop applications, design the user experience in a way that can be recognized and understood by the users of today*.

3.5 Architecture of a Semantic Desktop application

Under the hood, we also find similarities in existing applications and generalize now to give the reader an insight to how today's applications are built. Separating user interface from database is a rule of thumb that can always help, the model–view–controller design pattern is also common. For Semantic Desktop applications, we find that a common pattern is to focus on the editing of a single resource, one after another, and support the usual actions of loading a file, editing it and storing it. In the semantic web, where the notion of files slowly shifts to the notion of RDF graphs, we propose an architecture that focuses on the editing of these graphs and resources inside graphs. The architecture, illustrated in 4, is aligned at the model-view-controller pattern:

– The *model* to show and manipulate is one RDF graph. It can be loaded from a local or remote repository and can also be stored remote or locally. Ontologies and related information are also models, but they are usually secondary data and seldom changed.
– The *controller* is application logic that is described using inference rules or program code. It is highly domain specific.
– The view (user interface) is already described above Fig. 2. It is also domain specific but conforms to common patterns.

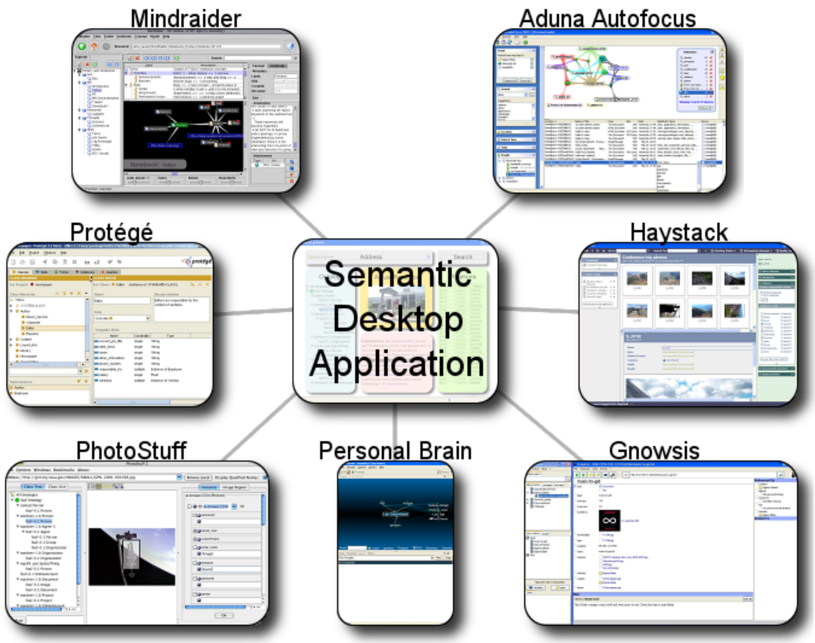

Fig. 3. Examples of existing user interfaces

3.6 Semantic Desktop enabling framework

To build information interaction applications which support above features, a basis architecture should be put in place. The diverse applications will need centralized services, so that not every application has to re-implement the wheel. These services will be part of a **framework** that runs as an invisible background server on a Semantic Desktop. Because they allow us to build user interfaces faster, we call them **enabling**.

Hence what services are elemental to a Semantic Desktop? This question is our concern in the *gnowsis.org* project, which serves as a prototype and test-bed for future applications. A few services are common technology today, the more complicated services are described below.

central RDF repository Even if the architectures differ – a central RDF repository is always there.

central search on the repository and documents a fulltext search and semantic (ontology aware) search service is needed

adapters It is agreed that existing data sources and applications have to be integrated. A detailed discussion on adapters can be found in [12].

ontologies The basis for information articulation and communication are common ontologies, their formal representation needs to be accessible to run inference and adapt user interfaces. We recommend to separate ontologies and make them available through dedicated services, so that developers clearly know what RDF graphs to use when the question comes to ontologies.

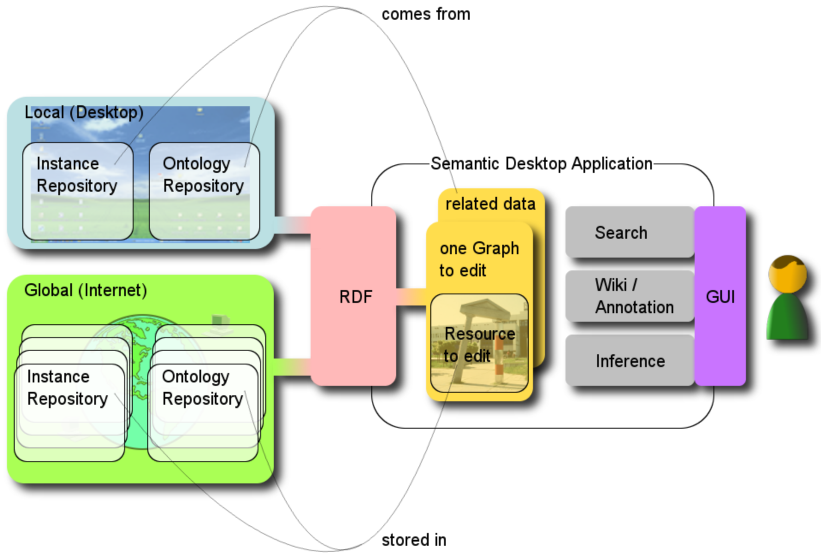

Fig. 4. A typical Semantic Desktop application architecture

context capture The quoted research projects suggest to observe the user behavior and user background via plugins to applications and the operating system. The plugins then report the actions relevant for the context to a service that stores the context and makes it available for other applications [35, 36].

pluggable architecture Service discovery and communication in the large are tasks tackled in the SWWS project[1]. Simpler structures can be used on the desktop suggested in [18].

These core capabilities are more or less available today. The next step will be to standardize their interfaces and provide stable implementations. The more difficult features (context and workflow) are still open but we expect to define the needed interfaces in the upcoming NEPOMUK project.

3.7 Merging the blocks—a Semantic Desktop

Above listing of existing and future developments leads to a description of an integrated system—the Semantic Desktop.

In Figure 5 an overview is given on how the building blocks of a Semantic Desktop work together. It is an evolution of the gnowsis architecture as described in [7]. The Semantic Desktop grows on a ground of data and information, the information is stored invisible to the user, in a database system or a RDF repository. The Semantic Desktop itself can now be seen as a tree – the roots of this tree are the stored information items and ontologies, invisible, under the surface, stored in semantic storage systems. Here we also find “grounding technology”. Above the soil are the applications visible to the user. They are independent from the tree but can use the tree to access the information in

[1] http://swws.semanticweb.org/

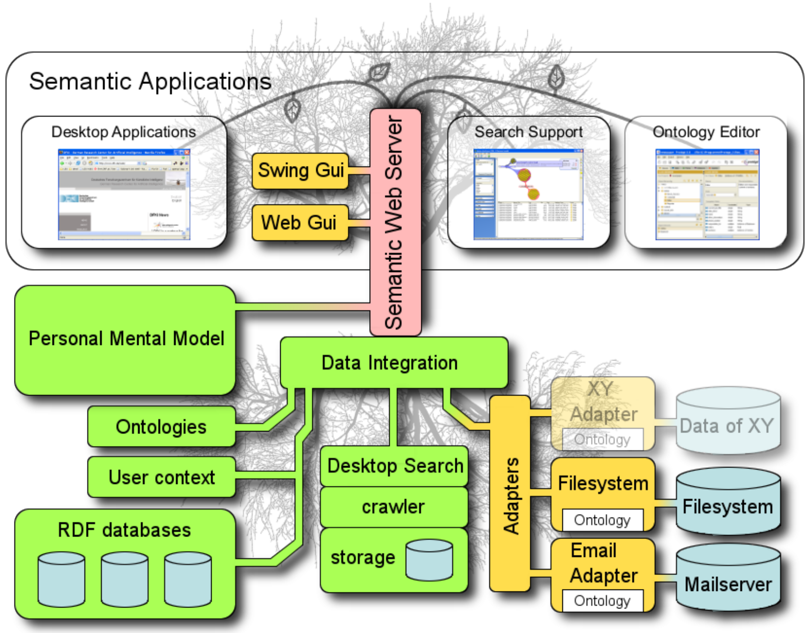

Fig. 5. Parts of a Semantic Desktop

the soil. The trunk of the tree, where it surfaces, consist of semantic web protocols and a server that gives access to the semantic services underneath. On this trunk, the branches and leaves grow, *information articulation and browsing* software. Applications can also connect to the tree and pick its fruits – use the information existing in the ground.

4 Summary and Outlook

The field we call today "Semantic Desktop" is both old (memex, hypertext systems and the web) and new (first publication with the term - 2003). A brief historical abstract was given, listing the projects and publications that form this field. The core idea is to bring Semantic Web technologies to the desktop, enabling people to use their desktop computers like a personal semantic web, where applications integrate and ideas are connected through ontologies. This idea was already addressed in several research projects and software products which are listed. We distinguished grounding technologies, ontologies and information interaction applications. These are the building blocks available today to build the Semantic Desktop of the future. We should now align our different ideas of the Semantic Desktop, for this we provided a definition of the term. We do not claim this definition to be final, but to be a starting point.

Also, we identified the need to standardize application programming interfaces and provide a background framework, that supplies enabling services. The user interface and the architecture of existing applications was presented and a view design patterns extracted, to provide developers with more indications where to start. The upcoming NEPOMUK project, proposed by a consortium of experts and lead by the DFKI, will help building a community of experts that develop and use the Semantic Desktop. Part of the project are free open source implementations to standardize the interfaces, provide developers with example applications and end users with useful Semantic Desktop applications.

The Semantic Desktop will connect the semantic web to individual people, working on their desktop computers. It will allow them to write down ideas and knowledge and to share these ideas with others.

Acknowledgement We want to thank Dominik Heim from the FH Kaiserslautern for the graphic design of the included illustrations and for the comments on user interaction and visual metaphors.

References

1. Bush, V.: As we may think. The Atlantic Monthly **176(1)** (1945) p101–108
2. Freeman, E., Gelernter, D.: Lifestreams: A storage model for personal data. SIGMOD Record (ACM Special Interest Group on Management of Data) **25** (1996) pp80
3. Nelson, T.: As we will think. On-line 72 Conference Proceedings **vol. 1** (1972) pp. 439–454
4. Lee, T.B.: Weaving the Web, The Past, Present and Future of the World Wide Web by its Inventor. Texere, London (2000)
5. Berners-Lee, T.: Frequently asked questions by the press. (http://www.w3.org/People/Berners-Lee/FAQ.html)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **89** (2001)
7. Sauermann, L.: The gnowsis-using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna (2003)
8. Decker, S., Frank, M.: The social semantic desktop. WWW2004 Workshop Application Design, Development and Implementation Issues in the Semantic Web (2004)
9. Sauermann, L.: The semantic desktop - a basis for personal knowledge management. In Maurer, H., Calude, C., Salomaa, A., Tochtermann, K., eds.: Proceedings of the I-KNOW 05. 5th International Conference on Knowledge Management. (2005) 294 – 301
10. Dengel, A., Abecker, A., Bähr, J.T., Bernardi, A., Dannenmann, P., van Elst, L., Klink, S., Maus, H., Schwarz, S., Sintek, M.: Epos - evolving personal to organizational knowledge spaces (2002) Project Proposal, DFKI GmbH.
11. Harth, A.: Seco: mediation services for semantic web data. Intelligent Systems, IEEE **Volume 19** (2004) 66 – 71
12. Leo Sauermann, S.S.: Gnowsis adapter framework: Treating structured data sources as virtual rdf graphs. In: Proceedings of the ISWC2005. (2005)
13. C. Bizer, A.S.: D2rq-treating non-rdf databases as virtual rdf graphs. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004). (2004)
14. Corp., M.: Information bridge framework. (http://msdn.microsoft.com/office/understanding/ibframework/default.aspx)
15. Quan, D., Huynh, D., Karger, D.R.: Haystack: A platform for authoring end user semantic web applications. In: International Semantic Web Conference. (2003) 738–753
16. Gemmell, J., Bell, G., Lueder, R., Drucker, S., Wong, C.: Mylifebits: Fulfilling the memex vision. In: ACM Multimedia December 1-6, Juan-les-Pins, France. (2002) pp. 235–238
17. et al., B.F.: the fenfire project. (http://fenfire.org/)
18. Geldart, J.: Rdf without revolution an analysis and test of rdf and ontology. Bachelor thesis, Department of Computer Science, University of Durham (2005)

19. Lausen, H., Stollberg, M., Hernández, R.L., Ding, Y., Han, S.K., Fensel, D.: Semantic web portals - state of the art survey. Technical Report 2004-04-03, DERI (2004)
20. McBride, B.: Jena: Implementing the rdf model and syntax specification. In: Proc. of the Semantic Web Workshop WWW2001. (2001)
21. the kowari project. (http://kowari.sourceforge.net/)
22. Corp., I.: Rdf gateway semantic web server. (http://www.intellidimension.com/)
23. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: Proc. of the International Semantic Web Conference 2002. (2002)
24. (edt), K.G.C.: Sparql protocol for rdf. Working draft, W3C (2005) http://www.w3.org/TR/rdf-sparql-protocol/.
25. Prud'hommeaux, E., (edts), A.S.: Sparql query language for rdf. W3c working draft, W3C (2005) http://www.w3.org/TR/rdf-sparql-query/.
26. Harth, A., Decker, S.: Yet another rdf store: Perfect index structures for storing semantic web data with contexts. http://sw.deri.org/2004/06/yars/doc/summary last change Jan 2005, visit Aug 2005 (2005)
27. McDowell, L., Etzioni, O., Halevey, A., Levy, H.: Semantic email (2004)
28. Osterfeld, F., Kiesel, M.: nabu semantic jabber server. (http://nabu.opendfki.de)
29. Elst, L.v., Abecker, A., Bernardi, A., Lauer, A., Maus, H., Schwarz, S.: An agent-based framework for distributed organizational memories. In Bichler, M., Holtmann, C., Kirn, S., Müller, J.P., Weinhardt, C., eds.: Coordination and Agent Technology in Value Networks, Multikonferenz Wirtschaftsinformatik (MKWI-2004), 9.-11.3.2004, Essen, GITO-Verlag, Berlin (2004) 181–196
30. ACM: Acm classes 1998. (http://www.acm.org/class/)
31. AG, B.: the brainfiler text classification system. (http://www.brainbot.de)
32. Maus, H., Holz, H., Bernardi, A., Rostanin, O.: Leveraging passive paper piles to active objects in personal knowledge spaces. In: Proceedings of 3rd Conference Professional Knowledge Management: Experiences and Visions. (2005) 43–46
33. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. (2002)
34. Clark, H.H.: Using language. Cambridge University Press (1996)
35. Schwarz, S.: A context model for personal knowledge management. In: Proceedings of the IJCAII'05 Workshop on Modeling and Retrieval of Context, Edinburgh (2005)
36. Chirita, P.A., Gavriloaie, R., Ghita, S., Nejdl, W., , Paiu, R.: Activity based metadata for semantic desktop search. (2004)
37. Miller, T., Agne, S., Dengel, A.: eFISK – eine aufmerksamkeitsbasierte Schlüsselwort-Extr aktions- und Information Retrieval-Maschine. Abschlussbericht 15202-386261/659, Stiftung Rheinland-Pfalz für Innovation (2005)
38. Miller, T., Agne, S.: Attention-based information retrieval using eye tracker data. In: Proceedings of the Third International Conference on Knowledge Capture (K-CAP05). (2005) To appear.
39. Barreau, D., Nardi, B.A.: Finding and reminding: File organization from the desktop. (1995)
40. Nejdl, W., Paiu, R.: I know i stored it somewhere - contextual information and ranking on our desktop. 8th International Workshop of the EU DELOS Network of Excellence on Future Digital Library Management Systems (2005)
41. Nielsen, J.: Mental models for search are getting firmer. http://www.useit.com/alertbox/20050509.html see also the tutorial on Fundamental Guidelines for Web Usability at the User Experience 2005 conference (2005)
42. Cooper, A., Reimann, R.: About Face 2.0 - The Essentials of Interaction Design. Whiley publishing inc (2003)

	Type	**Today**	**Semantic Desktop**
Grounding Technology	Storage	* Jena * Sesame * RDF Gateway	storage supports SPARQL and semantic protocols
	Search	* Lucene * Desktop Search Tools	semantic search services
	Communication	* Jabber, IM * email * P2P networks	semantic messaging and P2P
Ontologies		* DC * FOAF * iCalendar * SKOS * Thesauri * PIM	* popular ontologies * ontology mapping tools * desktop ontologies
Information Interaction	Ontology Editing	* Protege * IsaViz * KAON	ontology editors present in all applications
	Domain Specific	* Tidepool/Storymill * PhotoStuff * RSS Readers	Semantic Desktop Applications
	PIM and Workflow	* Microsoft Outlook * Lotus Notes * Frodo Taskman	Semantic PIM, Semantic Workflow

Table 1. Building Blocks