Desktop Gateway: Semantic Desktop Integration with Cloud Services

Aleksandar Kareski
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Rugjer Boskovik 16, P.O. Box 393, Skopje, FYR of Macedonia
a.kareski@gmail.com

Milos Jovanovik
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Rugjer Boskovik 16, P.O. Box 393, Skopje, FYR of Macedonia
milos.jovanovik@finki.ukim.mk

Dimitar Trajanov
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Rugjer Boskovik 16, P.O. Box 393, Skopje, FYR of Macedonia
dimitar.trajanov@finki.ukim.mk

ABSTRACT

With the development of IT technologies, the amount of information that the user receives and the sources of information which he or she uses, increased dramatically. There is a need of an intelligent desktop application, which will integrate different information sources from both the desktop and the web. To address this problem, we present a solution for a semantic desktop application, which relies on the use of ontologies to annotate and organize data, extracted from any application or website, into one environment. This paper gives an overview of the semantic desktop paradigm and its relevance to the knowledge management of the future. We also show how the semantic desktop application can connect with the cloud services for an enhanced user experience.

Keywords

Semantic desktop; personal information management; cloud computing; enhanced user interface; social semantic desktop; semantic web services

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval - information filtering, retrieval models, search process.

1. INTRODUCTION

There are a lot of choices a user faces in today's computing world; most of them are too complicated. There are hundreds of features, dozens of user preferences, unresponsive programs, inscrutable error messages, crowded toolbars, and general disrespect for the safety of the user's data. All of these are problems which plague most of today's software.

Conference'6, September 19–21, 2013, Thessaloniki, Greece.

On the other hand, in order to complete a given task, the user has to open numerous applications and websites. The situation within the organizations is even more complicated. The employees have to switch between various systems, such as HRM, CRM, ERP, etc. This frequent switching demotivates the employees and has a significant impact on the organization. Because of this, the user loses focus on his or hers task and wastes precious time jumping through the applications in order to find the information he or she is looking for. Additional time is lost due to the registration and log-in requirements of applications, as well as remembering many passwords for various authentication systems. Our solution will simplify all of this, by using one application to get all information that the users need, and remembering only one password.

In this paper we will present a solution for integration of all user sources of information, from both the desktop and the Web, in one place. We call this solution Desktop Gateway. With it, data from menu systems and applications is combined into one solution, which interacts with the operating system and gives the user an easy-to-use interface. It also provides a simple and natural solution to the problems of information retrieval. With just one click on a word or any on-screen text from any application, the system will extract the relevant information from the context and deliver it to the user.

The main goal of Desktop Gateway is to provide a client-side connection with public and private services on both the desktop and the Web. Its features will interpret the data and detect the context, by using external ontologies. Based on the extracted context, the user can use the possible actions on the Cloud applications and other web services. The interface is simple and easy to use, and is based on text search.

2. RELATED WORK

According to the constant development of computer software, application development is now focused on user friendly interfaces and applications which use dynamic and effective methods to access the information from lots of data storages.

GNOME Do[1] is an intelligent launcher tool which makes performing common tasks on the user's computer simple and efficient. It does not only allow the user to search for items in his or hers desktop environment (e.g. applications, contacts, bookmarks, files, music), it also allows him or her to specify actions to perform on search results (e.g. run, open, email, chat, play). GNOME Do provides instantaneous, action-oriented desktop search results, which adapt to reflect the habits and preferences of the user. Our solution uses a similar approach, but it searches on both desktop and web resources. GNOME Do provides the user with familiar graphical depictions of search results, in contrast of other search tools which present search results as flat, homogeneous lists. Our solution, Desktop Gateway, focuses on good UI as well, and groups the search results in several tabs, according to the search result type.

Enso[2] is a language-based interface, which performs actions based on the commands typed in by the user. It has its own set of commands that can help the user accomplish specific types of tasks, in any application and the OS. For example, selected text (numbers) can be calculated inside Notepad, with the 'calculate' command. The result of the calculation is automatically added back into Notepad. This hot-key launching concept is also used within our solution, where we use it to get text and images from any desktop application for further processing.

The OpenCalais[3] web service allows automatic annotation of content with rich semantic metadata [1], including entities such as people, companies, events and facts. It analyses the content using methods such as natural language processing and machine learning, in order to discover the entities, events and facts within the text. The metadata returned as a response from the web service is an RDF construct which is centrally stored. The metadata gives the user the possibility of building maps, networks or graphs by linking documents to people, geographic places and companies. We use OpenCalais within our solution for determining the context of the input data.

Babylon-Enterprise[4] gives the users a one-click access to enterprise information from ERP, CRM, HRM and other information systems, and allows them to drill-down to the data in its source application. It works within any Windows application, and enables both online and offline access to information. Similar to Babylon, our solution uses a private information system, but also uses public web resources as sources of information.

The Gnowsis semantic desktop project [11] aims to develop a semantic desktop environment which supports P2P data management based on desktop services. Gnowsis uses ontologies for expressing semantic associations and uses RDF for data modeling. Within our solution we use a similar approach, but we are concentrated on external ontologies and cloud services. However, the emphasis of Gnowsis is more on the flexible integration of a large number of applications, than on the organization and manipulation of data, which is a focus used in our approach.

One of the main projects which use the integration approach is the Haystack [10] system developed by the Computer Science and Artificial Intelligence lab at MIT. Haystack aims to create, manipulate, and visualize arbitrary RDF data, in a comprehensive platform. For visualization, it uses an ontological / agent approach, where user interfaces and views are constructed by agents using predefined ontologies. The user manages his or hers personal information at most suitable way. It integrates many applications like text processors, email clients, image manipulation applications, and instant messaging. A complete semantic environment is created from the user interface, all the way to the database. A disadvantage of the system is the performance problem.

Stuff I've Seen [6] is based on a simple, but powerful idea – people usually want to find information which they have already seen; therefore, it provides an indexed space that includes not only the information on the user laptop, but also on web pages he or she visited [2]. A similar concept is used by our solution, which caches the search results. This increases the speed of the search for information and data which are frequently looked up.

The Universal Labeler [8] is a unified scheme for labeling all kinds of personal information, e.g., electronic documents, paper documents, email messages or web references. The hierarchy of subprojects and tasks is displayed in a window that not only manages the UL, but also gives direct access to other applications (e.g., email or calendar). This hierarchy is created by a *drag-and-link* operation, in such a way that users drag only a fragment of a document to the hierarchy, and the remaining information is hyperlinked to that fragment. Each node can own planning-oriented properties or behaviors, such as *remind me by* or *due dates*, that will be displayed, for example, in the user's calendar. A hierarchy provides a view over the user's personal information. The UL approach does not attempt, however, to manage the storage of information items, like we do.

Semanta [13], [9] is a prototype application for semantic email and represents an add-on for email agents. The SMail model is encapsulated in the semantic email ontologies. The connection between these ontologies and the semantic desktop ontologies is a step forward to emerging desktop and email information. Semanta uses these ontologies to enhance email messages with semantics. Message annotations can be automatically executed by a speech extraction web service, or manually, by the use of a wizard. Metadata are transported together with the email contents with the help of RDF MIME extensions of the email header. Defining such models with formal work patterns enables formal semantics for email collaboration.

Personal information application (PIA) [2] is an ontology-based framework consisting of three loosely-coupled layers: the application layer, the domain layer, and the resource layer. The first layer supports directly the specific user's activities, such as travel planning to attend a conference. The ontology that models the application provides a view over the ontologies at the domain level. Those ontologies are not specialized to meet the requirements of a particular user, but are standardized and engineered to meet wide use requirements in applications such as air travel, hotel booking, or event registration. At the resource layer, they have addressed the organization of data including the use of file descriptions and domain ontologies as annotations and the extensive use of data associations. The

[1] http://cooperteam.net/
[2] http://www.humanized.com/enso/
[3] http://www.opencalais.com/
[4] http://www.babylon-enterprise.com/

extensive annotation and the associations among data create a network of data and metadata that can be traversed using the concept of semantic navigation and lend itself well to the formulation of expressive database-like queries. The specification of a PIA is performed using a tool called the PIA designer, with which the users define the displays, the queries, and the channels. User studies were conducted to evaluate the prototype of the PIA designer by computer science majors (albeit with limited knowledge of data modeling or query languages) and received positive feedback. It was, however, recognized by the participants that queries and channels are more suitable for expert users than for casual users. Our solution also uses layered architecture which is extended with the cloud applications and external web services.

Semantic Sky [14] presents a software platform developed using semantic web technologies, which provides the users with a unified and simple composite approach to the different services they use, with a simple flow of information from one infrastructure to another. The system is able to automatically discover the context in which the users are working, and offer them the actions that can be used within the context. The users can completely focus on their tasks in their work environment, and get relevant information and possible actions in that context. This system automates the execution of the users' tasks, which leads to improvements in their productivity, information exchange and efficiency. Unlike Semantic Sky, Desktop Gateway is focused on the advanced interface for desktop and web integration by using external services for semantic annotation.

3. OUR APPROACH

Desktop Gateway is a semantic desktop application with a strong focus on integration. Besides data from desktop applications and folder hierarchy, it also integrates data from the web. It uses an advanced user interface which interacts with the OS and provides a pleasant user experience. The desktop application is enhanced with the functions of the operating system, semantic web services and external web services. The context from the various sources is combined and analyzed within the application, before the end result is presented to the user. For each entity which is recognized in the context, a list of possible actions is suggested to the user. Depending on the type of the entity and the context, various actions are available.

The application is based on four main components: a source of information, semantic resource entities, types of the semantic resource entities, and proposed actions for the entities. Besides the main components, the application uses semantically annotated web services on the Cloud which were developed on our university, the OpenCalais semantic web service and lots of other external web services. The concept of the application is in accordance to the conclusions of the overview of the Semantic desktop paradigm by Sauermann, Bernardi and Dengel [12].

The application uses two types of sources of information: public and private. Social networks, desktop user data, e-mail accounts, web calendars and other private sources on the web and on the users PCs are used as private sources of information. Public sources of information include web encyclopedias, word translators, geographical maps, etc. This component is extensible with other sources, but the integration depends on the API of the new source. Using the Social Semantic Desktop approach by Decker and Frank [4], the current version of this system incorporates the private services of Facebook, Gmail, Google Calendar, public services such as Geonames, Wikipedia and Microsoft Translator, as well as the public semantic search web service Open Calais and private semantically annotated web services within the Semantic Sky cloud. The folders and files structure on the desktop computer is also used as private source of information.

The semantic resource entities are the core component of the application. They are created according to the context which is identified from the corresponding services and sources of information. The semantic web services from the Semantic Sky system help in the process of determining the context of the input text and determine the type of the resource entities by semantic search in the ontology, which is also situated on the Cloud. Semantically annotated RDF data types from the web service are then processed through the application. The OpenCalais semantic service also helps in determining the context of the search string by using its OWL ontology. The web service returns RDF data which contains information about the RDF resource type of the objects that are recognized with the semantic search of the text. RDF data is then processed in our application for determining the type of the recognized objects and adding actions to those objects. Every object has a relevance index which determines the relevance of the object in the sentence, and a social tag which shows the categories that are related to the object. Other services return basic types of resource entities, such as person, contact information, hyperlinks, places, etc., which are directly used in the appropriate module. Based on the semantic type of these resources, the candidate actions are found. The actions are actual web or REST-based services from the service repository of the application, or their composition.

There are two types of actions in the application: public and private. Public actions are performed on resource entities which are recognized with the public services - Wikipedia, Google Maps, Microsoft translator, etc., and they use public web services (open a link in the browser, show a place on the map, show textual information, translate a sentence, etc.). Private actions use semantic web services from the Semantic Sky system for the specific objects. According to the type of the objects different semantically annotated actions can be evoked on the Semantic Sky cloud (SaaS – software as a service). Other private actions which are used with the basic types of resource entities are sending email, writing a message on a Facebook wall, viewing calendar event details, etc.

On Figure 1 we depict the process of interaction with the applications and the operating system, the gathering of information from various sources, the method of context extraction and the execution of the proposed actions. On the left side of the picture, the process of activation of the application and retrieving the selected text through the Clipboard of the OS is shown. In the middle, the process of context extraction, with the help of the sources of information on both the desktop and on the web, is presented. In the bottom of the picture, the applicable actions for the identified data and context are shown.

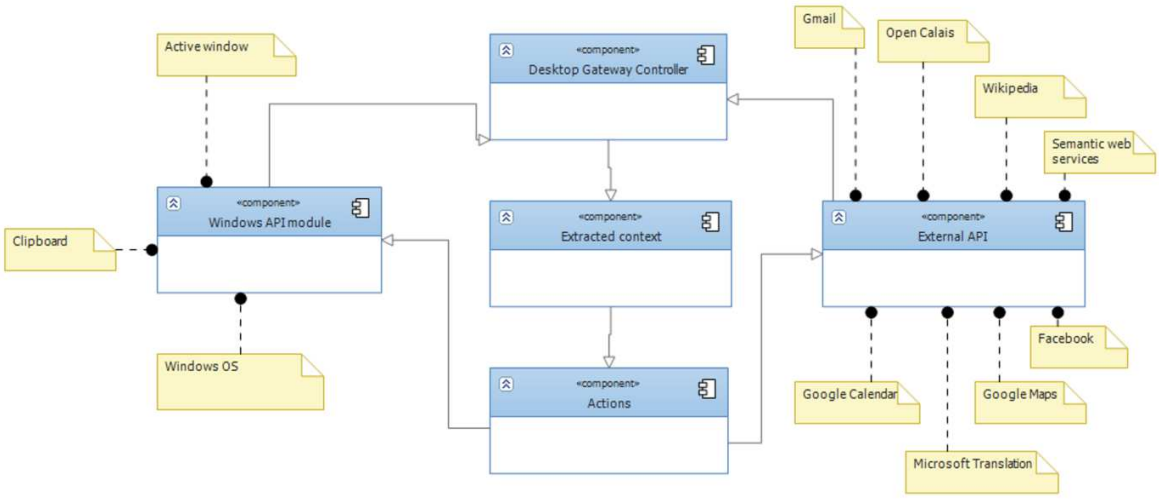

Figure 1. Interaction with the applications and the operating system.

4. IMPLEMENTATION DETAILS

The development of the application was divided in a few phases. The first phase was the development of a desktop application which interacts with the operating system. Tools for fetching text from any active window or part of the OS and module for copying the selected text from the Clipboard were implemented in this phase. In the second phase, a connection was established with the external sources of information (social networks, email, encyclopedias, the Cloud, etc.) by implementing APIs and web service references. The last phase was a generalization of all of the components in a social semantic desktop application. An intuitive and user-friendly UI was created to fit all modules on a separate tab.

Desktop Gateway runs on a Windows OS and is implemented with the Microsoft.NET platform; it uses the win32.dll, kernel32.dll and user32.dll libraries for OS interaction. SOAP and REST web services are used for gathering data and executing actions.

The application is divided in 3 layers:

- Communication layer – In this layer the web service interfaces are situated, as well as the internal and external API classes. This layer is used to improve the data transfer (gathering and sending data) between the desktop application, on one side, and the Cloud services, web services and other external resources on the other side. The communication layer is based on the Semantic desktop collaboration infrastructure by Decker [3].

- Controller layer - This layer is responsible for parsing the data gathered in the communication layer into structured objects, which are recognized from the social tags of the RDF data. The context of the search text is determined by the RDF resource data which is processed in the application. Structured object are presented to the graphical interface layer and saved in the local cache for further use. The context of the data is the foundation of the Semantic Desktop [7].

- Graphical interface layer – Presents the extracted information to the end user and receives the user input. It uses its advanced GUI to interact easily with the user. It is hotkey-activated and works in background when it is not being used. Search results are divided into separate tabs, sorted by the type of the data.

Part of the search results are saved in the cache, which stores data in a local MS SQL database. The cache speeds up the process of data searching for the common search queries. While external services need a few seconds to retrieve data, cached information are loaded immediately.

The interaction of the Desktop Gateway with the sources of information is accomplished with external APIs. Every source of information has a library which is referenced in the source code of the project. These libraries are used as interface for the physical connection with the remote servers. Such libraries implemented in the project are CalaisDotNet for the Open Calais API, Facebook API for Facebook, Google.GData for the Google contacts, Google Calendar and Google Maps, Geonames API for geographical landmarks, the Microsoft Translator SOAP web service, and the Wikipedia SOAP web service reference. A specialized API was built for the interaction with the web services from the Semantic Sky system. The OAuth standard is used for private API authentication of the users. With this, the user doesn't have to enter passwords for every private source of information which he or she uses, every time the application is started. Passwords are safely stored and encrypted.

5. WINDOWS API INTERACTION

The Windows OS has a lot of functions that are available in the Windows API, which allow direct interaction with the OS and executing OS-level commands.

The Desktop Gateway application interacts with the operating system and with the applications which are installed on it. It works in background and is activated by hotkeys. Global hooks

are used to detect hotkeys, to get the selected text from clipboard, and to activate the application window. The SetWindowsHookEx and CallNextHookEx functions from the user32.dll library are used to set the hook on the application initialization, and get the next activation of the hook.

System hook libraries allow keyboard and mouse activity detection, even when the application in which they are implemented works in the background or has no user interface. They throw the KeyEventArgs and MouseEventArgs .NET exceptions, which help acquire the needed information.

The hooks allow the application to work in the background and wait to be activated on a click of a hotkey. They also allow the selected text form the active window to be copied to the clipboard. When the application is activated, objects from the clipboard are copied in the application by the functions in the System.Windows.Forms.Clipboard class. The clipboard acts as a bridge between the processes. The data from any application is easily transferred to the Desktop Gateway application. The application gathers two types of information; the selected text and the surrounding text of the currently active window. This information helps in the process of determining the context of the data.

Many PC users use context menus for fast navigation through the windows. Our application has an integrated context menu which copies the path from the selected file or folder. The name of the window from which clipboard object originated is taken with GetWindowText function from the Win32 user32.dll library. The path and the name of the window are used in the process of understanding the context of the data.

6. USE CASE

Most of the applications today have complicated interfaces with hundreds of features, which makes their use rather difficult. The users have to navigate through complex menus, links and pages to find the information or actions they need. The goal of our approach is developing an application that will gather personal and relevant results to the search string as simply as possible. Our application has a very intuitive user interface which connects the operating system and the open applications with the Desktop Gateway. It is based on a simple text search with no additional features and options. Using the global hooks, the user doesn't even have to write the text. The selected text from any window is copied to the search box with a click on a hotkey. Just like Babylon Enterprise, the most powerful feature is its single click activation. Additional advantage to the application is the semantic data, which helps the process of context extraction. RDF data from the external ontologies determines the type and the context of the search text. The data is then processed through the layers of the application and the result is structured objects which are classified by its type in different tabs, where the user can look into it and use the available actions. Because it runs in the background, it can be activated whenever the user needs it

Detected resource entities have a list of specific associated actions. Depending on the type of the resource entity (person, place, subject etc.), different actions are available. Some of the actions are shown in Figure 2. The available actions allow an email to be sent to the persons detected in the context, with an optional attachment, or to write to the detected person's Facebook wall. For the detected geographic places, a short information summary is displayed and a link to a Wikipedia article is provided, for additional info. The places are also displayed on a Google Maps window within the application.

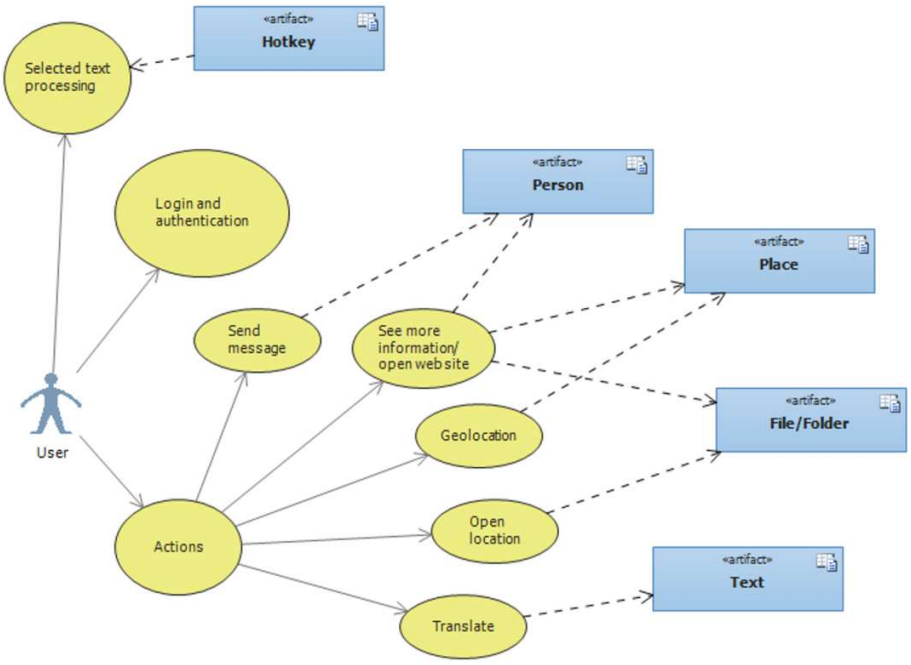

Figure 2. Actions associated with the selected object.

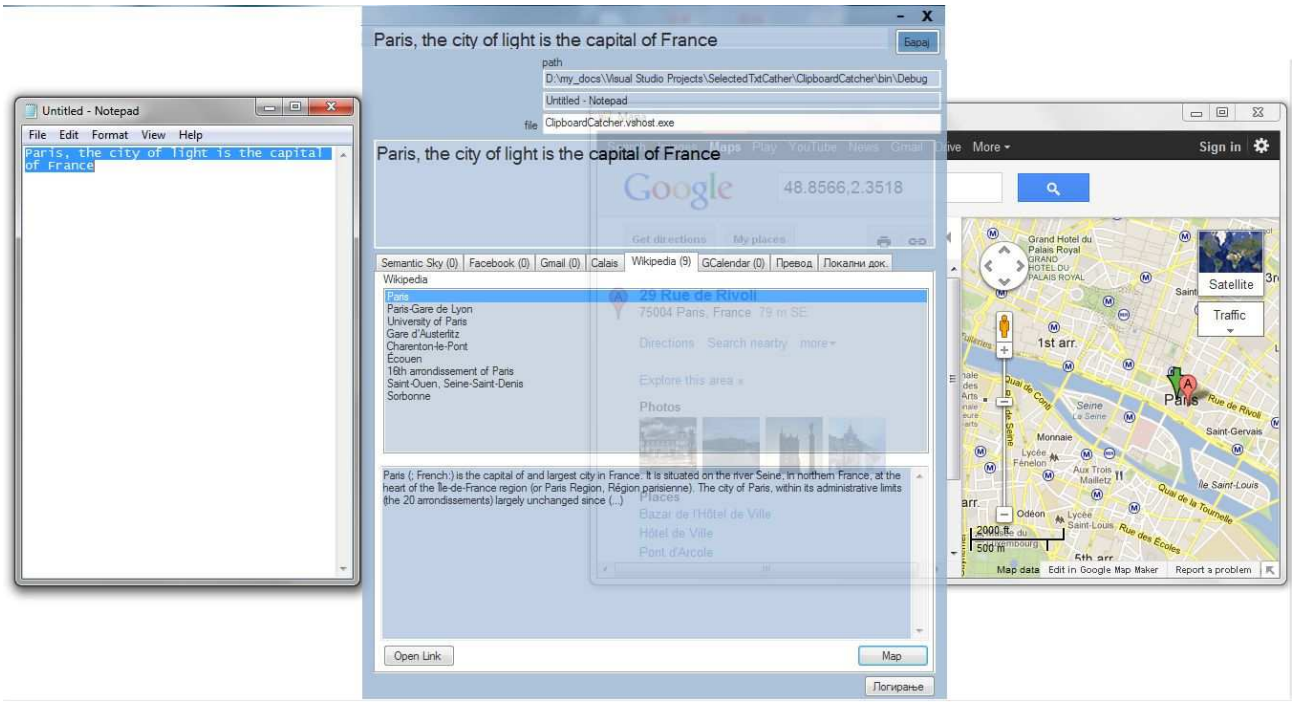

Figure 3. Screenshot of the application.

The files and folder which are detected can be opened directly in the folder in which they are situated. After starting the application, the user can login at the private sources of information he or she wants to use. The use of OAuth allows the user to enter them only once. The user credentials are securely saved and encrypted. When the application is active, the user can simply enter the text he or she wants processed in the text box. When the application is inactive, the user can select a text anywhere on the OS working environment and use the hot key to activate the application..

Figure 3 shows the activation of Desktop Gateway and capturing text from a Notepad window. After processing the text, the results are shown in a Wikipedia tab, with a short description and a map of the detected and identified country.

The Desktop Gateway application uses the benefits of the applications with advanced user interface like GNOME Do and Humanized Enso, it enhances the data with semantic annotations from the OpenCalais service and integrates a large number of web services into a single user friendly solution.

7. CONCLUSION

The basic idea of this paper was to bring the benefit of the semantic web technologies to the desktop, enabling people to use their desktop computers like a personal Semantic Web, where applications are integrated, and ideas are connected through ontologies. The enhanced desktop experience intends to bring all personal and relevant information, from both the desktop and the web, together into one familiar environment.

The Semantic Desktop paradigm intends to connect the Semantic Web with the users which work on their desktop computers. It allows ideas and knowledge to be saved and shared with others. It has a potential to bring in a new style of interaction in the personal computers which is not possible with the conventional technologies. In such knowledge-based environment, the desktop is defined in relation with the user and not with the hardware, operating system, applications and protocols which are used. However, more parameters are needed for shaping the semantic desktop to fit the needs of an individual user.

This paper is presenting a solution which is an interaction of many active research fields, like the semantic web, semantic desktop, web services cloud computing, which will result with a web based infrastructure for collaboration through the desktop PC. The application should accomplish reduction of the searching- and filtering-time and provide easier access to vital information.

There is bright future for the Semantic Desktop and the Semantic Web. More and more resources are annotated semantically on the web, but also on the user PCs. This contributes to the growing number of new project which use semantic metadata and the growth of the use of ontologies. The use of better ontologies will produce more accurate and relevant results to our application.

The Desktop Gateway application intends to bring the Semantic Web and the benefits of the semantic web technologies to individuals working on their desktop

computers. It allows them to work more seamless on their PCs, saving valuable time in context switching between applications.

We believe that the Desktop Gateway also has a bright future in which it should be constantly updated. Possibilities of upgrading with new sources of information are almost indefinite. Upgrading of the search algorithm and parallel threads for gathering information will help for faster access to the search results. Another aspect that should be reviewed is the upgrade of actions and better interaction between the search results for enhanced user experience, by adding more ontologies. The heterogeneity of knowledge models and ontologies can be solved by representation of the knowledge with named graphs in Nepomuk Representation Language [14].

REFERENCES

[1] Alergus, E. A. 2011. *Using OpenCalais API in the context of Linked Data*, Faculty of Computer Science, Distributed Systems, Romania Babylon Enterprise.

[2] Cruz, I.F., and Xiao, H. 2008. A Layered Framework Supporting Personal Information Integration and Application Design for the Semantic Desktop. *The International Journal on Very Large Data Bases*, vol. 17, no. 6, pp. 1385 – 1406.

[3] Decker, S. 2006. The Social Semantic Desktop: Next Generation Collaboration Infrastructure. *Information Services and Use.* vol. 26, no. 2.

[4] Decker, S., and Frank, M. 2007. The Social Semantic Desktop. *In Proceedings of I-Semantics 2007*.

[5] Dourish, P, Edwards, W. K., Lamarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D. B., and Thornton, J. 2000. Extending Document Management Systemswith User-specific Active Properties. *ACM Transaction of Information System*.

[6] Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R., and Robbins, D.C. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 72-79. ACM*.

[7] Heath, T., Motta, E., and Dzbor, M. 2005. Context as Foundation for a Semantic Desktop. *4th International Semantic Web Conference.*

[8] Jones, W, Bruce, H., Foxley, A., and Munat, C.F., 2006. Planning Personal Projects and Organizing Personal Information. *Annual Meeting of the American Associationfor Information Science and Technology*.

[9] Moller, K., and Decker, S. 2005. Harvesting Desktop Data for Semantic Blogging. *Proceedings of the first Semantic Desktop Workshop, at the 4th International Semantic Web Conference*.

[10] Quan, D., Huynh, D., and Karger, D.R. 2003. Haystack: A Platform for Authoring End User Semantic Web Applications. *International Semantic Web Conference*.

[11] Sauermann, L., Grimnes, G.A., Kiesel, M., Fluit, C., Maus, H., Heim, D., Nadeem, D., Horak, B., and Dengel, A. 2006. Semantic Desktop 2.0: The Gnowsis Experience. *In Proceedings of the International Semantic Web Conference*.

[12] Sauermann, L., Bernardi, A., and Dengel, A.. 2005. Overview and Outlook on the Semantic Desktop. *In Proceedings of the 1st Workshop on The Semantic Desktop at the International Semantic Web Conference.*

[13] Scerri, S., Handschuh, S., and Decker, S. 2008. Semantic Email as a Communication Medium for the Social Semantic Desktop. *European Semantic Web Conference*.

[14] Sintek, M., Van Elst, L., Scerri, S., and Handschuh, S. 2007. Distributed Knowledge Representation on the Social Semantic Desktop: Named Graphs, Views and Roles in NRL. *European Semantic Web Conference, Innsbruck, Austria*.

[15] Trajanov, D., Stojanov, R., Jovanovik, M., Zdraveski, V., Ristoski, P., Georgiev, M., and Filiposka, S. 2012. Semantic Sky: a Platform for Cloud Service Integration based on Semantic Web Technologies. *Proceedings of the 8th International Conference on Semantic Systems, pp 109-116, ACM*.