# PIMO Population and Semantic Annotation for the Gnowsis Semantic Desktop

Norberto Fernández[1,2], Leo Sauermann[2], Luis Sánchez[1], Ansgar Bernardi[2]

[1] Telematic Engineering Department. Carlos III University of Madrid.
Universidad 30, 28911, Leganés, Madrid, Spain.
[2] German Research Center for Artificial Intelligence, DFKI GmbH.
Erwin-Schrödinger-Str. 57, 67608, Kaiserslautern, Germany.

**Abstract.** The Semantic Desktop brings the ideas and the technologies of the Semantic Web into the personal computer desktop. As a prerequisite for applying Semantic Web technologies to a certain domain of knowledge an ontological model of the domain is required. In the Gnowsis Semantic Desktop, the PIMO (Personal Information Model Ontology) addresses this problem by providing a generic lightweight ontology whose classes model the main concepts involved in the daily activities of a person: places, organizations, persons, etc. But in order to be fully useful for a certain user, this generic model needs to be personalized and populated, adding more classes and concrete instances of the existent classes. As the process of manual population could be tedious and time consuming, in this paper we propose an alternative which tries to exploit the information that the user provides while performing Web searches. Apart from populating the PIMO, our approach is useful in resource annotation.

## 1 Introduction

Every day Internet users interchange e-mails, browse Web pages, download files of different types or create new ones as part of their daily life activities. All the information generated by these processes is heterogeneous in nature: the data is represented in different formats, handled with different tools, and is related to different topics or user activities. Managing his/her own information in an effective manner is thus difficult for the user.

The Semantic Desktop [1, 2] tackles this problem by bringing the ideas and the technologies of the Semantic Web [3] into the personal computer desktop. The main aim of the Semantic Desktop is to integrate the heterogeneous data sources of the user into a formal knowledge base, easier to manage and to share with other users.

Applying Semantic Web technologies to a certain domain of knowledge, like the personal desktop, requires the definition of ontologies modelling that domain in a formal, computer understandable manner. In that sense, the Gnowsis Semantic Desktop framework [4] exploits the results of the EPOS [5] project and provides a Personal Information Model Ontology, PIMO [6]. This PIMO consist of a small hierarchy of commonly used classes like *Location, Person, Organization, Project, Event, Topic*, etc (see figure 1).

In order to better reflect the interests of a certain user, the generic model provided by the PIMO needs to be personalized and populated, adding more classes and concrete instances of all the existing classes (that is, concrete persons, concrete organizations, etc.). As the process of manual population could be tedious and time consuming, in this paper we propose a system, SQA4Desktop, which tries to exploit the information that the user provides while performing common activities like Web search. Basically we try to exploit the fact that the users frequently look for information of interest on the Web by querying Web search engines. While doing so, they type in queries which refer to concepts the users are interested in: persons, organizations, papers, etc. For instance, a person wishing to travel to Italy would probably perform queries with the keyword *Italy*. Our system will take advantage of such information in capturing the interests of a certain user (for instance adding the concept *Italy* to the user model). As we will see, apart from populating the PIMO, our approach would be useful in resource annotation. In that sense, our current work continues the one introduced in [7] by integrating the SQAPS (Semantic Query-based Annotation, P2P Sharing) system into the Gnowsis Semantic Desktop framework and elaborating on the topic of user's interests capture that we left as future work in [7].

The rest of this paper is organized as follows: section 2 will describe our system, its working model and the characteristics of a beta implementation which is already available for download at [8]. Section 3 analyzes several proposals of Semantic Desktop frameworks. Our analysis will be centered on two aspects: what kind of user desktop models they provide and how they populate such domain models taking into account the interests of the user. Conclusions and future work in section 4 finalize this paper.

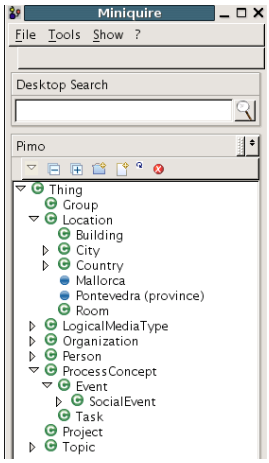


**Fig. 1.** The Gnowsis PIMO browser showing a part of the PIMO hierarchy

## 2 System description

As we have said in section 1 the SQA4Desktop system aims at providing a means for exploiting the process of querying Web search engines for populating the personal user information model and annotating resources. Basically the system requires from the user the annotation of his/her query by associating a concept or set of concepts to it, providing a computer friendly description. The concepts involved in this process need to be taken from an ontology or other semantic source, which in our case is the Wikipedia [9]. So the system invites to the user to provide some URLs of Wikipedia pages describing the concepts involved in the query. Then the user can associate the keywords and the concepts of the query to one Web resource, annotating it, and, at the same time, providing information over his/her interests.

In this section we will describe the SQA4Desktop system and its working model through a simple use case. A prototype implementation of the system is available for downloading and testing at [8]. It has been implemented as a service for the Gnowsis Semantic Desktop environment and has been tested with the most recent stable version of this environment (Gnowsis 0.9.0). It has been developed as a Web application using JSP Pages [10] for the graphical user interface. The Web search functionalities are implemented using the Yahoo Web Service APIs [11].

### 2.1 Use case

Alice wants to visit Italy on her next holidays, because some friends of her have told her that it is a really nice country. She has never been there, so she does not know so much about the country, and wants to look for information on the Web. Opening her Gnowsis desktop, she opens the SQA4Desktop Web search window in the Web browser and, as she is used to do, she simply types some keywords into a text field, for instance *Italy Rome*, and clicks on a button. By doing so, the system gets the keywords in the query, and sends them to a Web search engine, obtaining a list of results which are shown to Alice (see figure 2). As can be seen, each result is described by the title, a small abstract and a link, mimicking the way the information is currently shown by Web search engines, in order to increase usability.

The search interface of the system provides the basic functionalities that are common nowadays on Web search engines: Alice can now move forward and backward over the pages showing results of her query, can open any result by simply clicking on the title link and can reformulate her query until she finds what she is looking for. Additionally the SQA4Desktop interface provides a special link for each item in the result set. This link allows the user to annotate a resource, storing its information into the personal desktop.

After some browsing and checking of the results, Alice finally finds a resource of interest. As she has spent some time and effort browsing the results to find the interesting one, she decides to annotate the result of interest to keep its reference for the future. In order to do so, she clicks on the *Click here to annotate this Web*

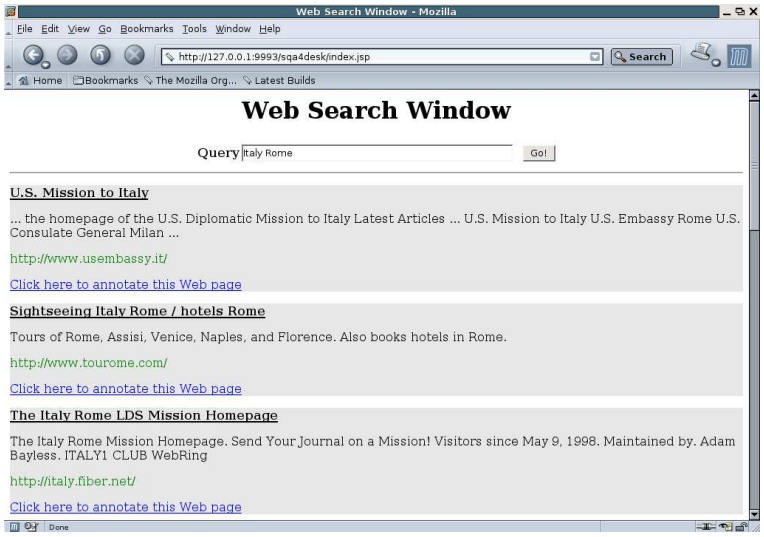


**Fig. 2.** Web Search Window

*page* link. By doing so, the system internally gets the keywords of the last query of Alice and looks in Wikipedia for entries related to that query. This is done by simply adding a restriction on the site where the results are being searched, that is, in our example the system will perform the following query: *site:wikipedia.org Italy Rome*. When the results of this query are collected from the search engine a new window is shown to Alice (see figure 3). This annotation window presents to her, among others, the following components:

- Information about the resource being annotated. The title, abstract and link of that resource are shown to Alice, providing her with the context for the process of annotation.
- A text field, where the terms of her previous query (that is, *Italy Rome*) are shown.
- The results of executing that query on the Wikipedia site, with the same format as the original results of the query that Alice had previously browsed. The only difference is that now the items in the result set do not have a *Click here to annotate...* but an *Add Concept* link, whose functionality we will describe below.
- Two buttons, *Look for Concepts* and *Annotate* that we will also describe below.

This new user interface allows Alice to perform the following operations:

**Annotation with the original query concepts** By clicking on the *Add Concept* link of a certain item, Alice states that the Wikipedia resource represented by such item is relevant to annotate the contents of the original Web

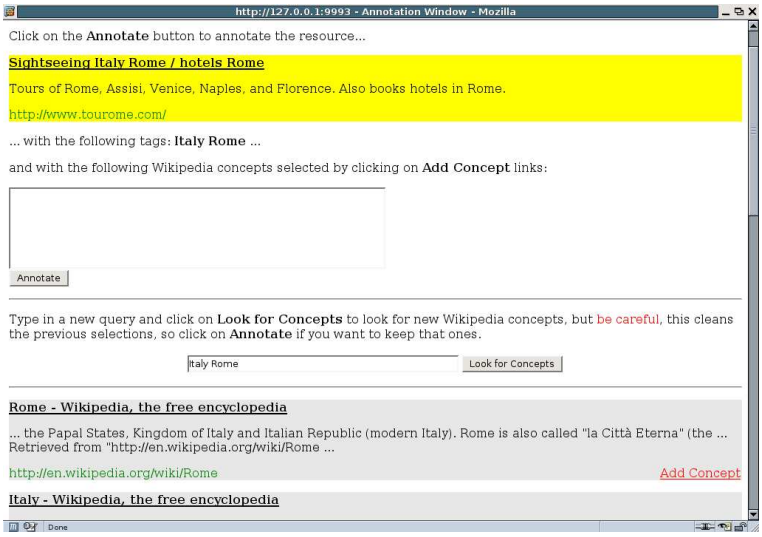


**Fig. 3.** Annotation Window

resource. Several items can be added (or deleted) to the relevant concept list, and in this sense the behavior of the system is similar to the one of a shopping cart on e-shopping. A text area shown in this window over the *Annotate* button contains the set of concepts that Alice is using in the annotation (has in her cart) allowing her to have a clear view of the annotation process.

**Annotation with other concepts** It can happen that the concepts being shown to the user are not the most adequate ones for annotating the resource. It can also happen that an active user would like to annotate the resource not only with the concepts in his/her original query, but with other ones. In order to deal with these situations, the annotation window of SQA4Desktop allows the user to replace the keywords of the original query with others. By clicking on the *Look for Concepts* button, the user can then look for Wikipedia resources associated to this new query. Using the same mechanism described above, relevant concepts for this new query are returned and are now available for use in annotation.

This process of typing in new queries, looking for relevant resources in Wikipedia, and adding these as annotations to the Web resource, can be thought as a kind of semantic tagging. Instead of tagging only with keywords, as in classical tagging systems like del.icio.us [12], the semantic tagging functionality of SQA4Desktop allows the user to tag the resource with Wikipedia URLs which represent concepts of interest. As a difference with the annotations derived from the user query, these semantic tags can be used to relate the Web resource with concepts which are not explicitly mentioned in the text contents of the Web resource.

**Tagging** If Alice does not select any Wikipedia resource as concept, an annotation can still be generated. This annotation associates the keywords in the last query *Italy Rome* to the Web resource. In that sense the application allows tagging in a similar way as well-known systems in the state of the art as del.icio.us.

Let us assume that Alice is a moderately active user. She is not going to spend time in providing new queries and associated concepts for that ones, but she decides to provide concepts at least for the terms in her original query. So she selects two of the results shown in the annotation window (the first describing *Rome*, the capital of Italy, and the second describing *Italy*, the European country) and clicks on the *Annotate* button. By doing so she triggers the following internal process at the system:

1. The Wikipedia pages selected by Alice are downloaded and processed. This process extracts from each page a label and a small definition for the concept being described by the page. The language links, which relate the Wikipedia page with other Wikipedia pages dealing with the same concept in different languages, and the category links, which relate the Wikipedia page with the Wikipedia categories the page belongs to, are also extracted. All this information is stored in the user model.
2. It is checked whether the concepts represented by the Wikipedia pages are already stored into the user desktop or not. If a concept is not stored, the system proceeds to store it, which in practice means:
   - Generating a PIMO compatible identifier for the new concept. The PIMO specification [6] indicates that PIMO identifiers should be URIs from the user's PIMO namespace.
   - Adding this concept as an instance of the *Topic* class of the default PIMO. This is so because we do not know exactly what is the class in the PIMO the concept represented by the Wikipedia page belongs to. In any case once the concept is added, it is easy for the user to move the concept to a different class by using drag and drop in the Gnowsis PIMO browser.
   - Generate the human readable description of the resource. The Gnowsis system includes a Wiki, Kaukolu Wiki [13], allowing the user to edit the PIMO data in a Wiki way. The short description of the concept extracted from Wikipedia is copied into this local Wiki for later reference.
   - Including an *occurrenceRef* relation between the Wikipedia page and the new concept, stating that the Wikipedia page represents a resource that can be used in the task of identifying the new concept.

   If the concept was already there, we simply get the identifier of the old entry, as we will need it for the following processing.
3. Generate automatically a *occurrence* relation. This links the identifier of the concept (new or old) with the Web resource, stating that such Web resource mentions or is related with the concept.

4. Generate automatically a description of the document being annotated. Basically we store its title, its URL and the abstract originally provided by the Web search engine. This information could be used in the future to show results of queries like *show me the documents that are annotated with a certain concept.*
5. Insert a new annotation object into the user model, containing the following information: the identifier (URL) of the document being annotated, the identifier (URI) of the user who has performed the annotation, the text of the query used in the annotation process, a time stamp indicating the annotation creation time and in case the user had decided to provide them, the concepts associated to such query.
6. In case the user provides concepts in his/her annotation, we update the user profile by means of some triples that relate the user with the language/s of the Wikipedia pages representing the concepts. We use the language prefixes of the Wikipedia URLs (like *en* in *http://en.wikipedia.org/Rome*) to do so. This information could be useful for retrieval of annotations in the future: when connecting in a P2P manner different user desktops, every user would have access to annotations in several different languages. Knowing the languages that the user used to select for the concepts, we can filter the annotations retrieving only those using the same language set.

Once this process ends, Alice refreshes the Gnowsis PIMO browser. As she did not have entries in her PIMO neither for *Italy* nor *Rome*, the entries for the new concepts appear now as new instances of the *Topic* class. Using the PIMO browser, she can then drag the entries and drop them in a different place to change their class (for instance, moving *Rome* to the *City* class). By double clicking on one of these entries, for instance on *Rome*, Alice can inspect (and edit) the desktop entry for the concept (see figure 4). As can be seen, in this window also appear the relations of the new concept with the Web resource annotated (*Relations* in right hand side) and with the Wikipedia page associated to the concept (*Important Resources* below the description).

By clicking on the *open wiki* button in the *Topic* instance description, Alice can open the Kaukolu Wiki page describing the resource (see figure 5), where she can modify, if required, the description of the concept.

### 2.2 Discussion

After briefly describing the system functionalities and working model, in this section we want to introduce, with discussion purposes, some aspects that in our opinion deserve special attention:

**Why querying?** We have chosen the querying process as source of information for resource annotation and PIMO capture. Several reasons support this decision, like for instance:

- Querying is a common user activity.

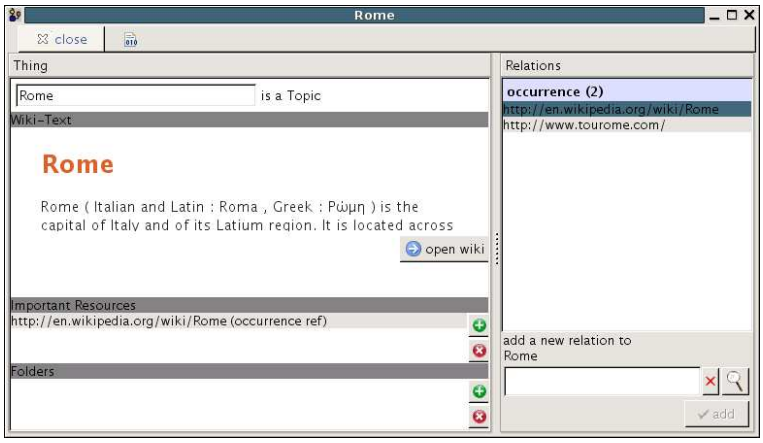


**Fig. 4.** The *Topic* instance *Rome*

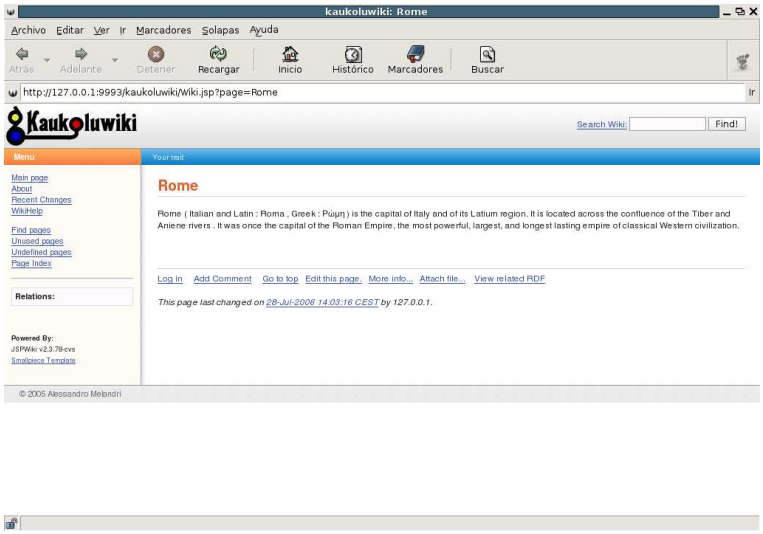


**Fig. 5.** Kaukolu Wiki page for *Rome*

- Queries usually reflect the interests of the user, making them attractive as a source of information for personal information model capture.
- Queries can be used to easily tag a resource with conceptually rich information. As a difference to classical tagging, where the users are completely free to tag a resource with whatever tags they want, when querying users are interested in obtaining relevant results as soon as possible. So we expect queries to be cleaner than tags, in the sense that personally useful but conceptually empty tags as *toread*, common in systems as del.icio.us, will not be frequent in queries. In any case, as tagging has

become a popular way of annotating resources, SQA4Desktop allows also the classical model of tagging based on user typed keywords and not only on queries.

**User Collaboration** As was the case of the system described in [7], SQA4Desktop is basically a manual approach to annotation and PIMO population, so user collaboration is required. In any case the process of finding the right concepts to be used in the annotation process is supported by search engines, which usually provide good results in query disambiguation tasks. Additionally, as we have seen, the system allows different levels of resource annotation, requiring each different degrees of user collaboration. Low-collaborative users can easily tag resources using the keywords in their queries. Medium-collaborative users can associate concepts (represented by Wikipedia resources) to the keywords in the query and annotate the resource with these concepts. Advanced and high-collaborative users can even look for new concepts to be used in annotating the resource, in what can be seen as a kind of semantic tagging, in the sense that the tagging is done with concepts instead of text tags.

**Why Wikipedia?** Continuing with the approach that we introduced in [7] we are using Wikipedia as a lightweight ontology which provides the concepts used in the annotation process. There are also several reasons to support this decision:

- It is open to public creation and modification of entries, making easier the process of knowledge maintenance.
- Usually the edition process of a Wikipedia page involves several different authors. So the contents of Wikipedia pages are the result of the agreement of different persons with different opinions and views of the same concept. This fits perfectly with the idea of ontology as a common and shared description of a certain domain of knowledge.
- It covers a broad range of concepts of different domains.
- Several external analyses have shown that its contents seems to have a reasonable quality [14].
- Web publishers can annotate their Web pages at authoring time by adding links to Wikipedia pages, clarifying the meaning of a certain expression.
- There are several different Wikipedia versions in different languages, and there also exist mappings between entries in different languages for the same concept. In this sense we can say that the Wikipedia is a multilingual ontology.
- The concepts in the Wikipedia ontology have their own Web pages. This makes possible the use of classical Web search engine disambiguation capabilities in the process of finding the right concepts for a query. The Wikipedia pages representing concepts can be annotated as any other Web resource, and we can use such annotations to detect possible new relations between concepts.

– Recent works like [15] also suggest the viability of using Wikipedia as an ontology.

On the negative side, one drawback of Wikipedia is that it is not a very formal ontology. It provides basically the semantics of a network of linked concepts with the addition of categories. Another difficulty comes from the fact that there is not a unique identifier for each concept: representations of the same concept in different languages have different URLs, and sometimes there are several representations for the same concept even in the same language, which Wikipedia handles through redirections. In order to deal with this problem, and avoid including the same concept in the user PIMO twice, we use the Wikipedia language links as a kind of equivalence relation, and computing the transitive closure of this relation, we detect in the PIMO if there already is some URL equivalent to the one which is being inserted as annotation. In any case, works dealing with making Wikipedia a more formal source can be found in the literature [16], and our work may benefit from that.

## 3 Related Work

In the state of the art we can find several approaches to the Semantic Desktop concept. These different approaches provide different ways of modelling the world of the desktop user, the concept(s) of interest for him/her. They also provide different ways of populating such model and using it to annotate resources.

**MindRaider** [17] is a Semantic Web outliner which uses mind maps [18] as the main way for user interests representation. In this tool the interests of the user are modelled by *concepts*. The user can define his/her own concepts manually by typing its label and description. This is done with the MindRaider GUI, which also allows to link concepts and to visualize the user mind map. Resource annotation is also allowed, and the user can, for instance, drag and drop a resource URL from other application (like Web browser, e-mail client, document visualizer, etc.) and link this URL with a concept or concepts. So, the population of the user mind map is mostly manual in MindRaider.

**IRIS** [19] is a semantic desktop system which provides a single environment to manage and organize the office information and structure it according to what the authors call a *personal map*. Analyzing the screenshots[3] and the information provided by the authors at [20] we conclude that the user and desktop model in IRIS consist of a set of *data item types* and *data items*. The data item types are predefined by the system: *People, Teams, Project, Tasks, Articles* and *Categories* and could be interpreted as classes. As far as we know, the only way of defining new data item types is by the use of applications that can be plugged into the IRIS framework and that provide these new data item type definitions. The data items could be seen as instances of the data item types. Data items could be created manually or automatically through the usage of applications. Different data items can be linked using what the authors call a *connection*.

[3] Some screenshots are available at: http://www.openiris.org/screenshots

The IRIS framework includes several applications which allow the definition of new data items. For instance, there is an e-mail client which allows to gather information from received e-mails (like new *People* data items from the "to" and "cc" fields of a received e-mail). Another example of application related to the topic of desktop model population is the clustering application which can process e-mails and organize them into clusters. These clusters can be manually named by the user and automatically inserted as *Projects* into the system.

Finally, we have to say that a more recent paper on IRIS [21] describes the idea of a two level desktop model based on a user constructed topic map [22] and an ontology hidden to the user. The topic map would provide access to the information in a personalized manner, whereas the ontology would provide more interoperability for information interchange among users. So it seems that changes in the data item/data item type model are being carried out, allowing the user to provide his/her own topics.

**Haystack** [23, 24] provides an integrated environment for personal information management. As is reflected in [25] Haystack comes with a predefined model covering many *object types* such as e-mails, contacts, MP3 files, photographs, papers and text documents. The user is allowed to modify this model by defining properties of object types to capture relationships between information that the user considers relevant. It is not specified in [25] if the user can also add new object types, but as object types should have associated *views*, which specify how to render the objects of that type, and *operations*, which indicates what can be done with the information in such objects, it is probably the task of new applications working in the Haystack environment to provide such new object types. In order to populate the system with concrete instances of the object types, Haystack includes *extractors* which can process information resources such as directory hierarchies, documents of different types, e-mails, Bibtex files, MP3 files, LDAP data, photographs, RSS feeds, messages or even Web resources to gather automatically such information. The system allows also to tag manually resources with keywords.

**DBin** [26, 27] is a tool which provides facilities for knowledge capture and P2P-based sharing within Semantic Web Communities. In DBin the authors introduce the concept of *Brainlets*, which are defined as *domain specific applications*. Each Brainlet provides domain-customized user interface, ontologies, rules, annotation types and queries. Brainlets are represented by XML files and can be downloaded and installed to extend the DBin functionalities. In that sense, and as a difference with other tools, DBin is domain configurable, not limited to a concrete domain like, e.g., the user desktop domain. But, as the model of the domain is provided by the Brainlet and can be populated but not extended by the user, we can say that in DBin the model is developer provided, instead of being a fully personal one.

Regarding to annotation, DBin allows the manual definition of annotations of different (domain-dependent) types. Integration with common desktop tools as Web browser or e-mail client is not currently available in DBin, but, as is indicated in [27] a local file system processor can be used to automatically extract

metadata from local files according to the file type.

**Chandler** [28] is a Personal Information Manager (PIM) integrating calendar, e-mail, contact management, task management, notes, and instant messaging. Taking a look at the Chandler GUI we can see that the model of Chandler provides a set of predefined personal information management object types or *item types* like *Message, Task, Event, Note*. The user can create at any time a new object or item as an instance of one item type by typing in some information. The user can also create his/her own *collections of items* which can be seen as a mechanism for item classification: for example the user can classify under the collection *SemDeskPaper* all the e-mails, personal notes, appointments in the calendar or tasks that are related to the process of writing a paper for the *SemDesk* workshop. The system allows to import information from other sources like e-mail servers using that to generate objects or items, populating the model in that way. The user can declare that an object is member of a certain collection simply via drag and drop, but the current system version does not allow to users to define their own object types nor to organize the collections into hierarchies.

Analyzing all these works, we see that most of the proposals include in their models definitions for describing things that are expected to be handled by common desktop users, like persons, locations, documents or projects. We can see also that all of them include means for populating the models, and most of them in an automatic manner, by gathering information from tools integrated in the environment, like e-mail clients or Web browser. Apart from MindRaider, where the entries in the model are represented by user generated concepts, the rest of systems distinguish between item types or classes and instances of these item types or classes. In all the systems, a relation between entries in the model can be defined by the user, but in most of them the user can only define instances and relations, but not new classes or types of items. In any case most of the systems are designed to be extensible, and these extensions may comprise the definition of new classes or types of items, so the extension of the model with new classes is left to the developer and not to the user. In principle this decision favors the interoperability: The users cannot extend the ontology but only populate it, so the classes are the same for all the users that use the same desktop application. On the negative side, in our opinion, the price paid for that is limiting the extensibility and, due to this, the personalization capabilities of the desktop application. In relation with the annotation task, we can say that all of the systems allow the annotation of resources, for instance by linking such resources to concrete concepts, and most of them allow also tagging of resources with text keywords.

In this paper we continue the work introduced in [7], where we proposed the usage of queries and Wikipedia in integrating semantic annotation with information retrieval. As far as we know the main contributions of this work are:

1. We have integrated the approach in [7] with the Gnowsis Semantic Desktop, to take advantage of the query-based annotation process in populating the PIMO of a certain user.
2. We have proposed a mechanism to exploit the information in multilingual

links for dealing with one the problems of using Wikipedia as source ontology: the lack of a unique identifier for a certain concept.

3. In [7] we only describe the annotation of resources with the user queries and their associated concepts. In our current prototype we have also added the possibility of tagging a resource with user defined keywords. It is also possible to find concepts for these tags or keywords, allowing for a kind of semantic tagging.
4. We suggest that mapping personal entries in the PIMO with concepts taken from Wikipedia provides a way of using Wikipedia as a kind of common grounding conceptualization shared among the different desktops of the different users. The existence of this common conceptualization allows the implementation of a GaV (Global as View) information integration solution, where we have the personal PIMOs of different users (the sources to be integrated) and a common model or global schema (the Wikipedia) acting as a kind of "bridge" among these different personal models.

## 4 Conclusions and Future Work

In this paper we have described the SQA4Desktop system, which aims at exploiting user queries in the process of desktop and user model population and resource annotation. Our system exploits Wikipedia as source ontology, and provides also facilities for tagging, both with user queries or with user specified text. A prototype version of this system for the Gnowsis Semantic Desktop framework has been implemented and is available for download at [8].

As our application is still a prototype, some limitations and future lines of further development exist like for instance:

- The current facilities provided by the Gnowsis PIMO browser do not allow the conversion of an instance of a certain class into a new class. As we define by default the new entries in the PIMO as instances of the *Topic* class, it is not possible at the moment to use such instances in defining new classes. In principle this is more a difficulty derived from the current user interface than for the SQA4Desktop system itself, so future versions of the Gnowsis interface could provide this functionality.
- It can happen that the user defines a concept by other means, not using SQA4Desktop, and that after some time, (s)he uses the concept in one of his/her queries. At the moment our tool is not able to detect the existence of that previous version of the concept, because the duplicate checking is based on the information captured with the help of SQA4Desktop, that is, in the existence of mappings between the concept and some Wikipedia page already processed in the past. Of course we cannot expect our system to be the only one providing new entries to the user PIMO, so a mechanism for duplicate or candidate Wikipedia mapping detection should be provided. A possible alternative in that sense is to show to the user in the annotation window not only the Wikipedia results provided by the search engine, but integrate

these with possible results already contained in the user's PIMO. In that sense the user could use the concepts at the PIMO to annotate the resources, minimizing the possibility of duplicate insertion. In any case, if a duplicate is inserted, the current Gnowsis GUI would allow to the user to manually define a relation between two things stating that they are equivalent.
- At the moment our application allows the definition and visualization of annotations, but does not provide any advanced capability exploiting the semantics of these annotations, like semantic search or *more like this* functionalities. As the Gnowsis desktop infrastructure provides a mechanism for knowledge sharing, these functionalities could exploit the knowledge provided by the different system users as long as they agree on a common ontological model, like the one provided by Wikipedia in our case.

## Acknowledgements

This work has been partially funded by the *Ministerio de Educación y Ciencia de España*, as part of the Infoflex Project, TIC2003-07208, by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW C01, Project EPOS: Evolving Personal to Organizational Memories) and by the European Union IST fund (Grant FP6-027705, project Nepomuk).

## References

1. Sauermann, L.; Schwarz, S.; Introducing the Gnowsis Semantic Desktop. In 3rd International Semantic Web Conference 2004 Poster Track, Hiroshima, Japan, November 7-11, 2004.
2. Decker, S.; Frank, M.; The Social Semantic Desktop. DERI Technical report available at: http://www.deri.ie/fileadmin/documents/DERI-TR-2004-05-02.pdf
3. T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". Scientific American, May 2001.
4. Gnowsis. Available at: http://www.gnowsis.org/
5. Sauermann, L.; Dengel, A.; van Elst, L.; Lauer, A.; Maus, H.; Schwarz, S.; Personalization in the EPOS project. In Proceedings of the Semantic Web Personalization Workshop at the 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006.
6. PIMO a PIM Ontology for the Semantic Desktop. Available at: http://www.dfki.uni-kl.de/~sauermann/2006/01-pimo-report/pimOntologyLanguageReport.html
7. Fernández-García, N.; Sánchez-Fernández, L.; Blázquez-del-Toro, J.; Luque, V.; Exploiting User Queries and Web Communities in Semantic Annotation. In 5th International Workshop in Knowledge Markup and Semantic Annotation, located at the 4th International Semantic Web Conference, ISWC 2005, November 2005.
8. SQAPS Web Site. Available at: http://www.it.uc3m.es/berto/SQAPS/.
9. Wikipedia: The Free Encyclopedia. Available at: http://www.wikipedia.org/.
10. Java Server Pages - Wikipedia, the free encyclopedia. Available at: http://en.wikipedia.org/wiki/JSP

11. Yahoo Search Web Services. Available at: http://developer.yahoo.com/search/
12. del.icio.us. Available at: http://del.icio.us/
13. Kaukolu Wiki. Available at: http://kaukoluwiki.opendfki.de/
14. Wikipedia evaluations.
Available at: http://en.wikipedia.org/wiki/Wikipedia#Evaluations
15. Hepp, M.; Bachlechner, D.; Siorpaes, K.; Harvesting Wiki Consensus. Proceedings of the First Workshop on Semantic Wikis: From Wiki to Semantic, SemWiki2006, colocated with the 3rd Annual European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 2006.
16. Krötzsch, M.; Vrandecic, D.; Völkel, M.; Wikipedia and the Semantic Web - The Missing Links. Proceedings of Wikimania 2005: The First International Wikimedia Conference, Germany, August 2005.
17. MindRaider - Semantic Web Outliner.
Available at: http://mindraider.sourceforge.net/
18. Mind Map - Wikipedia, the free encyclopedia.
Available at: http://en.wikipedia.org/wiki/Mind_mapping
19. IRIS Semantic Desktop. Available at: http://www.openiris.org/
20. OpenIRIS documentation.
Available at: http://www.openiris.org/documentation
21. Park, J.; Cheyer, A.; Just For Me: Topic Maps and Ontologies. In First International Workshop on Topic Maps Research and Applications, TMRA 2005, Leipzig, Germany, October 6-7, 2005, Revised Selected Papers. Lecture Notes in Computer Science 3873 Springer 2006.
22. Topic Map - Wikipedia, the free encyclopedia.
Available at: http://en.wikipedia.org/wiki/Topic_Map
23. Haystack project. Available at: http://haystack.lcs.mit.edu/
24. Karger, D. R.; Bakshi, K.; Huynh, D.; Quan, D.; Sinha, V.; Haystack: A General Purpose Information Management Tool for End Users of Semistructured Data in Second Conference on Innovative Data Systems Research, CIDR 2005, Asilomar, California, January 4-7, 2005.
25. Karger, D. R.; Quan, D.; Prerequisites for a Personalizable User Interface. In Intelligent User Interfaces, IUI 2004, Workshop on Behavior-based User Interface Customization.
26. DBin Project. Available at: http://dbin.org/
27. Tummarello, G.; Morbidoni, C.; Puliti, P.; Piazza, F.; The DBin Semantic Web platform: an overview. WWW2005 Workshop on The Semantic Computing Initiative (SeC 2005).
28. Chandler, a next-generation Personal Information Manager (PIM) integrating calendar, e-mail, contact management, task management, notes, and instant messaging. Available at: http://chandler.osafoundation.org/

This article was processed using the LaTeX macro package with LLNCS style