User Data Privacy in Web Services Context Using Semantic Desktop – SemanticLIFE Case Study

Mansoor Ahmed, Amin Anjomshoaa, A Min Tjoa
Institute of Software Technology and Interactive Systems
Vienna University of Technology
FavoritenStrasse 9-11, Vienna, Austria
{mansoor, anjomshoaa, amin} @ifs.tuwien.ac.at
WWW homepage http://www.ifs.tuwien.ac.at

Abstract

The growing number of Web Services technologies and their use have revolutionized the web. Web Services will play an important role in the next web generation (i.e. Semantic Web) together with Semantic Web technologies. As a matter of fact, Web Services and Semantic Web are two building blocks to provide machine process-able services. One of the biggest challenges in both Web Services and Semantic Web is privacy issues. Privacy means which part of information should be hidden and which should be visible. In the web services context, no matter if it is a simple or a complex service, the requester and provider of the service has to disclose information for handshaking, so privacy issues will always exist . In the utilization of web services, there exist exchange and storage of information, so the protection of personal information is very important. To acquire any service, one has to disclose personal information (e.g. home address, DoB, mobile number, credit card information etc.) so as to fulfill the requirement and utilize the service properly. But the problem arises when the submitted information is shared with a third party. In collaborative environment, where different people are interacting with each other and also the information is being shared among different people, the sharing of information without exposing unrelated information becomes a difficult task. The problem statement and its solution explained in this paper are focused in the domain of health information systems, but they can be used in a collaborative enterprise environment. In this paper we have introduced ontology-based user data privacy in the web services domain using the semantic desktop's (SemanticLIFE) SOPA framework.

1. Introduction

The fast growth of the World Wide Web and the emerging pervasiveness of digital technologies within our information society have significantly revolutionized business transactions, trade and communications between people and organizations. Besides the augmentation effect, business-related information is characterized by the fact that it also originates from heterogeneous sources and get more and more complex in structure, semantic and communication standard. Therefore, mastering heterogeneity becomes a more and more challenging issue for research in the area of Business Process Management. This challenge involves all the facets of process integration, composition, orchestration, and automation amongst heterogeneous systems.

Fortunately, Web services, built on top of existing Web protocols and open XML standards, have recently emerged as a systematic and extensible framework for application-to-application interaction. Web services allow automatic and dynamic interoperability between systems to accomplish business tasks. However, the implementation and the effective use of Web Services are not yet fully explored. The process of assembling “pieces of functionality” into complex business processes is often thinkable just for big enterprises and for ordinary computer users there is no easy way to interact with the Web Service ecosystem. Nowadays the personal computers are 1000 times more powerful, but just a small percent of their resources is effectively used. We think the time has come to use the wasted power of PCs to enhance the people-to-people and people-to-machine communications.

SOPA is a lightweight implementation of a service-oriented framework; it stands for “Service Oriented Pipeline Architecture” and is aimed at extending the usage domain of Semantic Web Services to personal computers with a simple and powerful approach. Using the SOPA framework, it is possible to build a useful gadget from existing services and share the composed gadget with others. The shared gadget can be again reused and customized by others as a building block to make new gadgets.

Moreover the SOPA framework is the basic communication means in the SemanticLIFE framework. So on the one hand it provides the service composition and execution issues and on the other hand it deals with user ontologies. In this paper we will exploit SOPA and SemanticLIFE framework to address the Web Service security and privacy issues.

This paper is organized as follows. In section 2 we reviewed the related work. Section 3 explains the architecture of SemanticLIFE system. Section 4 consists of problem statement and usecase scenario, section 5 describes the proposed solution and section 6 comprises the necessary ontologies to accomplish our goal. In section 7 we have mentioned some policy examples and section 8 consists of our conclusion and future directions.

2. Related Works

Due to increase in web services-based business applications and processes, data privacy in web services is becoming more and more important. Privacy as defined by Westin is “the claim of individuals, groups, and institutions to determine for themselves, when, how, and to what extent information about them is communicated to others” [1]. A number of interest groups are working in the domain of semantic web services. For example, Digital Enterprise Research Institute (DERI) tries to address intelligent Web Services upon Semantic Web technologies. Since there will be millions of services available on the web, the real challenge is discovering them and the way in which they automatically communicate with each other. The selection of the suitable web service to carry business interaction among enterprises can be automatically discovered on the basis policies [2]. Another working group at DERI is ESSI WSMO which aims at developing a language called Web Services Modeling Language (WSML) that formalizes the web services modeling ontology (WSMO) [3].

In [4] the authors have introduced semantic-based user privacy in web services based on the preferences defined by the user using rules. Declaring privacy preferences on the basis of service ontology prevents the user from repetitive specifications, since the privacy preferences at the upper classes are inherited by lower classes. Furthermore, the presented framework allows Web services to declare alternate data requests if a mandatory input is not given by the user. In [5] the authors have introduced the privacy authorization framework to tackle all the privacy requirements defined in the “Web Services Architecture (WSA) Requirements” document. A citizen’s personal information privacy is very important in digital government environment where different government departments interact with each other. The proposed solution is based on combining digital privacy credentials, data filters and mobile privacy, preserving agents to enforce privacy. Access to stored information in different government agencies is handled through the use of filters.

Kagal et al. state that policies should be part of semantic web services [6]. A policy specifies who has access to which service and under which conditions and how the requester's information will be handled at the requester's side. They also suggest that ontologies should be used to annotate OWL-S input and output parameters with respect to their security characteristics, including encryption and digital signature. Moreover, they propose to incorporate privacy and authentication policies into OWL-S descriptions and requester profiles. They extended the OWL-S VM with features for encrypting and signing messages exchanged between service requester and provider. In [7] Baresi et al. proposed a solution to monitor the functionality of web services, i.e. data communication, security and privacy, based on policies. Different types of policies i.e. service policy, requester policy, provider policy and server policy can be also defined along the life cycle of the web service.

3. SemanticLIFE Architecture

The SemanticLIFE framework is developed on a highly modular architecture which provides the basic components for the proposed web service interaction mechanism that will be discussed in later sections. SemanticLIFE stores, manages and retrieves the lifetime's information entities of individuals.

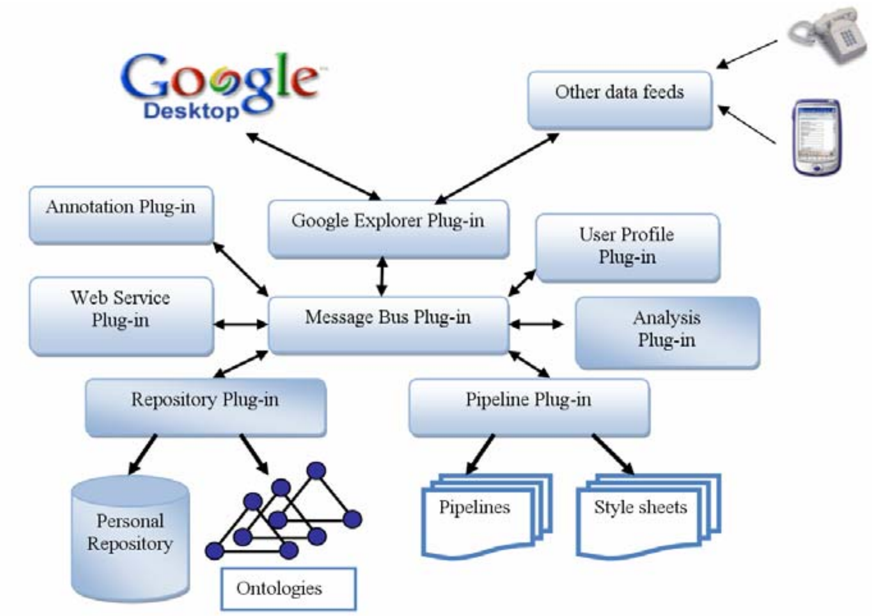

Fig 1: SemanticLIFE Framework Architecture

It enables the acquisition and storage of data while giving annotations to emails, browsed web pages, phone calls, images, contacts, life events and other resources. It also provides an intuitive and effective search mechanism based on the stored semantics (for more details see [19]).

An overview of the system architecture is depicted in Figure 1. The whole SemanticLIFE system has been designed as a set of interactive plug-ins that fit into the main application and this guarantees the flexibility and extensibility of the SemanticLIFE platform. Communication within the system is based on a service-oriented design with the advantage of its loosely coupled characteristics. The Service Oriented Pipeline Architecture has been introduced in order to compose complex solutions and scenarios from atomic services from SemanticLife plug-ins. SOPA provides a paradigm to describe the system-wide service compositions and also external web services as pipelines. SOPA provides some mechanisms for the orchestration of services and the transformation of results. The pipeline plug-in plays a central role in the orchestration of basic system services and in the creation of new business services. It enables the end user to describe his/her scenario using the pipelines and existing SOPA services. The newly created services (pipelines) may be shared with other users that may need the new service. Web Service plug-in manages the global system services including ordinary plug-in services, pipelines and external web services. The Web Service plug-in will also manage the semantics of pipelines like all other services, i.e. the pipeline functionality. More importantly its input/output parameters are annotated using the domain ontology.

Data with user annotation is fed into the system using a number of dedicated plug-ins from a variety of data sources like Google Desktop captured data, communication logs and other application's metadata. The data objects are passed on by the message handler to the analysis plug-in. This plug-in contains a number of specific analysis plug-ins providing semantic mark-up by applying a bunch of feature extraction methods and indexing techniques in a cascaded manner. The semi-structured and semantically enriched information objects are forwarded to the repository plug-in for an ontologically structured storage, called the meta-store. A set of query processing and information visualization tools provides the means for information exploration and report generation. The analysis module and metadata extraction capabilities make associations among the lifetime items and lifetime events based on user annotation, user profile and the system ontologies.

4. Problem Statement: A Use Case Scenario

In the use of web services there are two major entities involved in the exchange of information, i.e., the requester who requests some service for the successful completion of his/her task and the other is the provider which provides the services for some particular service, e.g. accounts information service, calendar service, on-going tasks service etc.

Consider a scenario of project development in an organization where there are different departments like the sales department, the HR department, the management department and the quality assurance department. Different people like managers, developers, programmers, software engineers work in their respective departments. Consider a user - who can be an employee of an organization or an outsider (client) - wants to access information within or outside the organization. In both cases, to have an access to a resource, permission will be granted on the basis of policies and rules defined by an organization. For example, a programmer need not know when the manager goes on private holidays, similarly a developer need not know about the activities of a person working in a different department, but a manager should have access to information such as who is involved in which project, what are the deadlines for the projects and who is leaving for holidays. Additionally, if person-X and person-Y are working on the same project, then person-Z should not have access to the code of that project (the project being confidential), although they are working in the same organization.

In the later case when the user is an outsider, he/she has to disclose personal information for identification. But before submitting personal information, the user must know the privacy practice of the organization for the requested resource and has to agree on that. A step-by-step interaction process between the user and the organization is depicted as follows:

1. In the first step, the user locates the desired web service through the UDDI registry.
2. After choosing one, the service provider asks the user to submit personal information.

3. Before disclosing the personal information, the user asks for the privacy policies of the service provider, which mean how his/her personal data will be handled etc.
4. The Service provider extracts its privacy policies information and sends this information to the user.
5. The User evaluates the privacy information of the service provider and, if it satisfies the user, then data will be sent.
6. When handshaking is done, then the user information at service provider side will be stored according to the user's preferences.

5. Proposed Solution

In this section we will explore the proposed solution for user data privacy in web services context using semantic desktop (SemanticLIFE) architecture, which plays an important role in the proposed solution. The SemanticLIFE architecture provides a collaborative environment for serving semantic services. The core concept of SemanticLIFE is wraps the services in semantic containers to make the services and their results machine process-able. The SemanticLIFE services can be an internal service like a desktop query or even an external web service that is managed uniformly in the SemanticLIFE environment. So whatever we discussed before about Web Services will be also valid for SemanticLIFE services. We will use the pipelining features of SemanticLIFE to apply policies and filters to Web Service call results and the scenarios will be realized via creating the relevant pipelines and services.

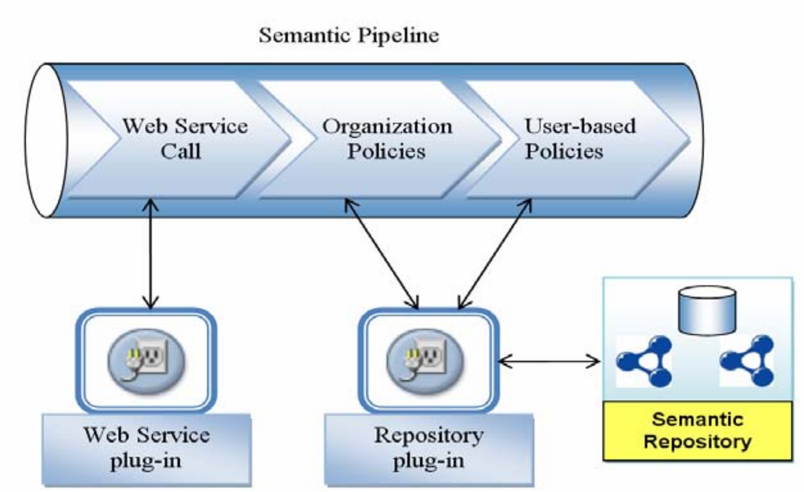

Fig 2: Web Service call using a Semantic pipeline

A SemanticLIFE user will send a request to access some resource, e.g. how many projects are going on in the organization and who is working on which project, or to access the code of a project. Depending upon the policies defined for that resource, only that amount of information will be exposed to the user. To accomplish this task the user has to identify him/ her while disclosing information like name, job title or department etc. The functionality of different components in the SemanticLIFE architecture is explained as below.

Organization and user-based policies are subject to various rules and constraints. Such rules can benefit the Semantic Web Inference technologies like RuleML [8] and SWRL [9]. In the SemanticLIFE case we have used Jena 2 Inference [10] support to implement the policies as rules.

In the proposed solution the rules are applied on the fly according to the calling user's specifications and also to the semantic of the web service call results.

A complete picture of the proposed solution is depicted in Figure 2. The scenario starts when a Web Service request has been received from the end user. Since the web service invocation information is already stored in an internal repository in the Web Service plug-in, the pipeline knows how to call the web service and get the raw results. In the next step the Web Service ontology will be considered and based on the retrieved items and organization policies, the rules will be applied to the raw data set. As a result at the end of this phase, we will end up with the filtered results that comply with organizational policies.

In the next step the pipeline should consider the specific requirement of the calling user (the user who has requested to receive the data) and apply the user-specific policies. The user-based policies are combined with pipeline results to produce the final pipeline output. As an example, depending on the user's role in the organization, part of the results should be closed to the calling user.

6. Exploitation of ontologies

As explained earlier, our approach to secure user data privacy is based on ontologies. To achieve this goal, we have used four ontologies, i.e. user, privacy, organization and service ontology. In this section we will explore the introduced ontologies and explain their roles in the proposed scenario.

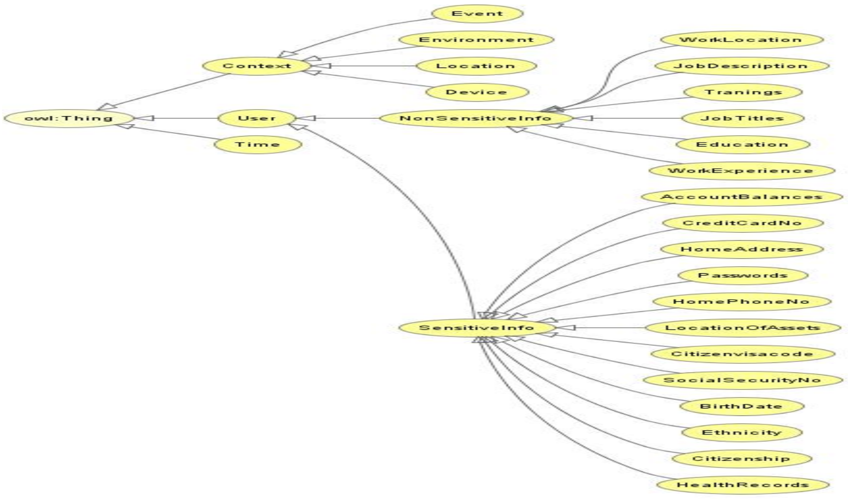

Fig 3: User Ontology

The user ontology as shown in Figure 3 defines the privacy preferences as sensitive and non-sensitive for his/ her personal data in response to the web service. Sensitive information means that the user doesn't want to disclose the personal information while non-sensitive stands for ready-to-be-disclosed personal information. In addition to that, the user context is also very impotent for information disclosure. For that purpose we have introduced the "context" class which describes the location e.g. home or office, which kind of device is used for accessing services, e.g. desktop computer, laptop or PDA etc. The context sensitivity is important in disclosing personal information. Another important feature is the "time", which means the user's location at a particular moment in time, e.g. before or after death. In other words, there might be some information that can be disclosed after a person's death. In short, the user data can be evaluated differently along time and context dimensions.

Figure 4 illustrates the hospital ontology (it is basically an organization ontology that is adjusted to the hospital scenario). In a hospital there exist the technical staff, the staff, the administration, the pharmacy, etc. Each department performs specific tasks according to their job description. As explained earlier, the sharing of related information in a collaborative environment is a difficult task. For this purpose we have introduced the hospital ontology which describes the basic structure.

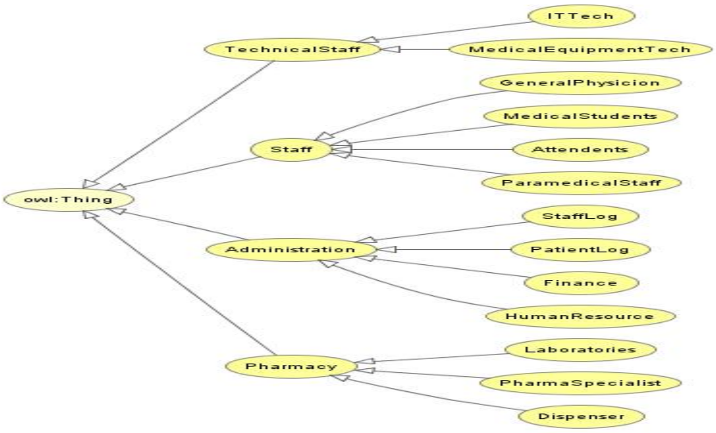

Fig 4: Hospital Ontology

The core technology in Semantic Web is web services. Additionally, agents will play an important role in the exchange of information, so the human control over the information will be smaller. In such case, the protection of personal information becomes challenging. Exchange of necessary information between service requester and provider will be accomplished with the help of agents [11]. To define the privacy policies and rules, we have used the privacy ontology from DAML-Services as follows.

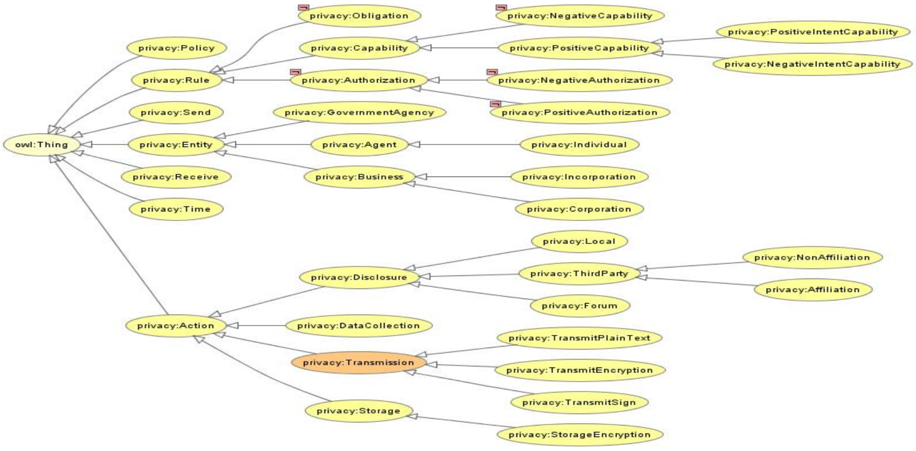

Fig 5: Privacy Ontology

The privacy ontology shown in Figure 5 by DAML-S expresses privacy policies and protocol for matching the privacy policies. The privacy ontology explains concepts like action, entity, rule, policy, time etc. The Entity class consists of three subclasses, i.e. government agency, agent and business. Each rule has an action and action is applied to some resource. If no "onResource" property is specified, the rule will be applied to all types of resources. The resource refers to the information that must be protected [11].

OWL-S is a Semantic Web Services description language which enriches web services description with semantics. Top level service ontology is depicted in Figure 6. OWL-S is divided into three main classes, namely Service model *"how it works"*, Service Grounding *"how to access it"* and Service profile *"what it does"* [12] [13].

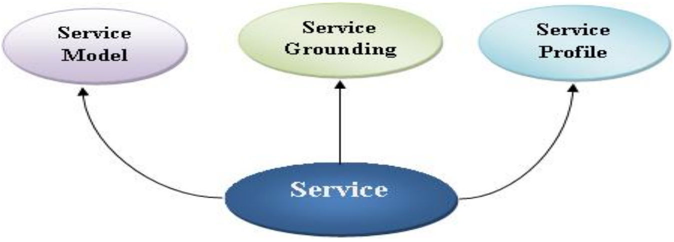

Fig 6: Web Service Ontology (OWL-S)

7. Policies

Policies are defined as "A set of rules that specify how a company or organization handles personal information collected from clients, which information from client needed to accomplish the task, for which purpose client information will be used, who else can use that information, e.g. government agencies, third parties etc and how long that information will be kept" [17].

In the health information system scenario, people have different types of access to information resources depending on their job description. In a collaborative environment where people work together, access to information resources should be allowed according to the defined policies and rules.

Policies specify who can use a service and under which conditions, how information should be provided to the service and how the provided information will be used [18]. In health information systems, the system collects detailed information about the user; this information contains the user's personal information, i.e. name, address, contact number, date of birth, home address etc and also the medical history. Most of the information collected by the health information systems is shared among different departments in the same domain for different purposes. For example, the information is required by insurance companies to keep the record of the user and also by the billing department of the health center for charging, while practitioners use patient information for future reference. Since the user information is scattered in different departments and different people are handling that information, the whole information is not required by each department, so the user's information must be handled separately among different departments. As an example, if a medical record of a patient shows that he/ she is carrier of STD (Sexually Transmitted Disease), this information is irrelevant for the billing department or insurance companies.

In our work, when the user wants to invoke a web service (health service), there is a need for policies. Privacy policies specify under what conditions information can be exchanged. For example, a privacy policy specifies that data transmission between requester and provider can take place only when they support data transfer in encrypted form. If none of those (i.e. requester and provider) fulfill this requirement, transmission cannot take place. Likewise, if the requester's policy says his/ her personal information should be deleted after a certain period of time and the provider has a different policy for handling data, then the transaction will fail.

In SemanticLIFE, the information in the semantic store will be handled through policies. Policies are stored as Jena rules. When the information is requested via SemanticLIFE, it will apply the rules to the Semantic Repository and initiate the inferred ontology which will be used to answer further queries. Furthermore, at runtime the new set of rules can be applied to the inferred set of triples. The fact that resources and corresponding rules and policies are distributed among many nodes will be especially important for dynamic environments like service-oriented architectures.

Also information about how data was handled previously will be stored in the semantic store to help the user with future decisions. Information in triple store will be handled through access control component and the user will be able to modify, delete or add policies through interface. The user defines the policy for some specific service, e.g. personal information sharing policies, data transmission action policies etc.

Few examples of the privacy rules for protecting personal information are shown below.

```
[hiddenTo1: (?requestor ws:requestsItem ?item)
        (?requestor rdf:type privacy:thirdParty)
        (?item rdf:type user:sensitiveData) → (?item sec:hiddenTo ?requestor)]
```

When this rule is applied to the union of the previously defined ontologies, we will end up with an inferred set of triples. Now if an outsider (third party) sends a request to access user's home address, the above mentioned rule hides will hide this information item from third party. At runtime, the user request will be processed as follows:

1. The original request will be rerouted to the relevant pipeline for the retrieval of information
2. The Pipeline will call the web service and hold the response
3. According to the web service ontology, the components of the web service result will be examined one by one against "user policies" and "requestor's context". the Pipeline will do this by repeated calls to the repository plug-in (see figure 3).
4. The part of information that should be closed to requestor will be filled out with blank.
5. The result will be sent back to requester.

The following rule is an example of context-based reasoning:

```
[hiddenTo2: (?requestor ws:requestsItem ?item)
        (?requestor context:hasLocation ?location)
        (?location rdf:type sec:inSecure)
        (?item rdf:type user:sensitiveData) → (?item sec:hiddenTo ?requestor)]
```

The rule closed the sensitive information to the requestors who are located in insecure places. Please note that from the first rule we already know that the requested information is or is not close to requestor and the second rule simply checks the requestor's location.

8. Conclusion and future work

The evolution of Semantic Web technology has opened a new window in IT and specially data engineering fields. However the higher layers of Semantic Web cake [16] which are proof and trust layers are not fully implemented yet. The proposed scenario showed the SemanticLIFE's approach to address security and privacy issues in service-oriented environments. Also, the presented SemanticLIFE platform as a personal information manager has the capacity to be used in other business processes dealing with personal information.

The proposed framework has already been applied to some scenarios like tourism and information retrieval [14] [15] and we are trying to apply it to other businesses and exploit the strength of Semantic Web Services as a business-enabler.

9. References

[1] L. F. Cranor. *Web Privacy with P3P*. New York, 1967

[2] Digital Enterprise Research Institute (DERI), http://www.deri.ie/ (Feb. 28, 2006).

[3] Roman.D et. al. Web service modeling ontology, working draft. http://www.wsmo.org/2004/d2/v0.3/. February 2004.

[4] Tumer.A, Dogac.A, Torouslu.I: A Semantic-Based User Privacy Protection Framework for Web Services,ITWP 2003.pp. 289-305.

[5] Rezgui.A, Ouzzani.M, Bouguettaya.A, Medjahed.B: "Preserving Privacy in Web Services". In ACM Proceedings of the 4th international workshop on Web information and data management, *WIDM'02,* November 8, 2002, McLean, Virginia, USA. Pages: 56 - 62 .

[6] Kagal. L, Paolucci. M, Srinivasan. N, Denker. G, K. Finin, T. Sycara : "Authorization and privacy for semantic Web services", IEEE Intelligent Systems 19 (4) (2004).

[7] Baresi. L, Guinea. S, Plebani. P: "WS-policy for service monitoring", In Proceedings of the 6th VLDB Workshop on Technologies for E-Services (TES'2005) held in conjunction with the 31st International Conference on Very Large Data Bases (VLDB'2005),Trondheim, Norway, 2005.

[8] Boley. H, Grosof. B, Tabet. S, and Wagner. G. RuleML: http://www.dfki.uni-kl.de/ruleml/indtd0.8.html, 2001.

[9] Horrocks. I, P.F. Patel-Schneider, Boley. H, Tabet. S, Grosof. B, and Dean. M, "SWRL: A semantic web rule language combining owl and ruleml", 2004,http://www.w3.org/submission/SWRL/.

[10] Jena 2 Inference, http://jena.sourceforge.net/inference/.

[11] DAML-S: http://www.daml.org/services/owl-s/security/privacy.owl.

[12] Martin. D, Burstein. M, Denker. G, Hobbs. J, Kagal. L, Lassila. O, McDermott. D, McIlraith. S, Paolucci. M, Parsia. B, Payne. T, Sabou. M, Sirin. E, Solanki. M, Srinivasan. Nand Sycara. K (2003). OWL-S 1.0 white paper. http://www.daml.org/services/owl-s/1.0/.

[13] DAML Services Coalition (Ankolekar.A , Burstein. M, Hobbs. J, Lassila.O, Martin. D, McIlraith. S, Narayanan. S, Paolucc. M, Payne. T, Sycara. K, Zeng. H),DAML-S: Semantic Markup forWeb Services, in Proceedings of the International Semantic Web Working Symposium (SWWS), July 2001.

[14] Semantic Enrichment of Search Result: the Coupling of Semantic Store and Google Services, K. Mustofa, A. Tjoa, A. Andjomshoaa; iiWAS2006.

[15] M. Nguyen, A. Tjoa, A. Andjomshoaa, F. Shayeganfar : "Utilizing Web Service Based Business Processes Automation by Semantic Personal Information Management Systems - The SemanticLife Case", PAKM2006.

[16] W3C Semantic Web Cake Layer: http://www.w3.org/2007/03/layerCake.png.

[17] Leino-Kilpi. H, Valimaki. M, Dassen. T, Gasull. M, Lemonidou. C, Scott. A, & Arndt. M (2001). Privacy: A review of the literature. International Journal of Nursing Studies, 38, 663-671.

[18] Carminnati.B, Ferrari.E, Hung.P.C.K : Exploring privacy issues in Web services discovery/agencies. Appears in IEEE Security and privacy magazine, Sept-Oct 2005,Volume 3, Issue 5, pp: 14- 21.

[19] Ahmed .M et. al, "SemanticLIFE' - A Framework for Managing Information of A Human Lifetime", Proceedings of the International Conference on Information Integration, Web-Applications and Services (IIWAS'04, 27-29th Sep. 2004 Jakarta-Indonesia).