TU WIEN

Technische Universität Wien

MAGISTERARBEIT

# Exploitation of Ontologies in Security and Privacy Domain

Ausgeführt am Institut für Softwaretechnik und Interaktive Systeme
Der Technischen Universität Wien

**Unter der Anleitung von O. Univ. Prof. A Min Tjoa**

**Betreuender Assisstent: Dipl.-Ing. Amin Anjomshoaa**

durch

**Mansoor Ahmed**
Matrikelnummer: 0327357
BrigittenauerLande 224/6543
A-1200, Vienna, Austria

May, 2008
Datum

Mansoor Ahmed
Unterschrift (Student)

# Eidesstattliche Erklärung

Ich erkläre an Eides statt, daß ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, Mansoor Ahmed

***“Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution”***

***Albert Einstein***

# Zusammenfassung

Die Entwicklung vom Semantic Web hat die Informationsspeicherung, Übertragung und Zugriff revolutioniert. Die Designprinzip von Sementic Web zielt auf Zusammenarbeit und Wissensaustausch ab. Die Vision von Semantic Web sind Information die für Maschinen verständlich sind, auch zu verwendet um sie mittels Computer zu kombinieren und auszutauschen. Ontologie beschreiben die Hauptkonzepte, gemeinsam mit Domainnamen und Verknüpfungen sind die Hauptbestandteile von Semantic Web. Ontologien erlauben die Interoperabilität und ein gemeinsames Verständnis von Domänen zwischen zwei Softwareanwendungen verschiedener Organisationen. Sie erlauben außerdem den Austausch von Daten auf syntaktische und semantische Ebenen. Im Semantic Web können zwei Beteiligte automatisch miteinander interagieren ohne voneinander zu wissen, daher sind traditionelle Zugriffskontrollmodelle nicht geeignet um in einer derartigen Umgebung eingesetzt zu werden. Deshalb werden Prozesse mit semantischer Unterstützung benötigt, die automatische Zugriffe auf sensitive Informationen behandeln.

Eine große Herausforderung in vielen Organisationen ist es Risikofaktoren für Computer und Netzwerke festzulegen. Legacysysteme in Organisationen sind durch unterschiedliche Risiken bedroht, z.B. Computervirus, bugs und Systemfehler durch Hardware- oder Softwarefehler. Diese können den Verlust von Daten zur Folge haben. Das Ziel ist es, Risiken abzuschätzen und so die resultierenden Probleme zu minimieren. Die Lücke zwischen Geschäftseinheiten (wie Projekten oder Rollen) und organisatorischen Infrastrukturen sollen mit Hilfe von Semantic Web überrücken werden. Die Geheimhaltung von Informationen in gemeinschaftlichem betrieblichem Umfeld indem verschieden Personen miteinander interagieren und Informationen austauschen ist wichtig. Eine schwierige Aufgabe in einem derartigen Umfeld ist der Informationenaustausch ohne fremde Information offenzulegen.

In dieser Arbeit zeigen wir, wie Semantic Web zur Lösung der oben genannten Probleme beitragen kann. Wir sind der festen Überzeugung, dass mittels Policy-Sprachen die Zugriffsrechte auf sensible Informationen entsprechend formuliert werden können, wenn sie mit domainspezifischer Semantik kombiniert werden. Zum Beispiel kann der Benutzer Art und Abstraktionsniveau der Informationen definieren sowie Randbedingungen aufstellen, wann diese veröffentlich werden dürfen. Der Benutzerkontext (d.h. Ziele, Projekte, Rollen und Profil) kann gemeinsam mit dem domainspezifischen Wissenskontext dazu beitragen, notwendige Richtlinien zu definieren, um sensitive Informationen als auch Weitergabe von Prozessen zu schützen.

# Abstract

The birth of Semantic Web has revolutionized the information storage, transmission and the way it is accessed. The design principle of the Semantic Web aims at providing collaborative working environment and knowledge exchange. The Semantic Web vision is the information which is understandable by machine and by using such technologies, information can be combined, exchanged and used easily by machines. Ontologies which describe the main concepts and terms of a domain and relationships among them will play a prominent role in the Semantic Web vision. Ontologies will also contribute to solve the problem of interoperability and shared understanding of common domains between software applications of different organizations. Also they allow the exchange of data both at syntactic and semantic level. In Semantic Web any two parties can interact with each other automatically without knowing each other, so traditional access control models are no more appropriate to be used in such environment where parties are known to each other in advance. Therefore, semantically enriched processes are needed to deal with automatic access to sensitive information.

The ultimate challenge in many organizations is to assess their risk factors for their computers and networks. Legacy systems in organizations are facing different kind of risks like viruses, bugs and system failure causing damages to hardware and software resulting in data loss. The goal is to calculate risks, so that problems resulting from them could be minimized and to fill the gap between business entities (like a project, a role) and organization infrastructure using Semantic Web technologies. Also the privacy of information in collaborative enterprise environment is important 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.

In this thesis we propose that Semantic Web technologies can be used to overcome the problems stated above. We have firm conviction that access control policies for

resources and sensitive information at different levels can be specified using policy languages and by applying appropriate filters when combined with the semantics of the knowledge domain. For example, the user can specify the information and the level of abstraction that can be shared and the circumstances under which this can take place. User context (i.e. user, goals, roles, projects) along with the semantics of the knowledge domain can help defining rules to protect the sensitive information (personal data privacy) and process sharing.

# Acknowledgement

I am thankful to GOD who gave me strength, courage and wisdom to achieve my goal.

This work is the successive result of a project done at the Institute for Software Technology and Interactive Systems (Vienna University of Technology) under the supervision of Professor Dr. A Min Tjoa. Without his inspiration, advices, guidelines and continuous support this thesis would not have been possible.

Special thanks to Amin Andjomshoaa who has always encouraged me and for having fruitful discussions. I am thankful to Mona Ioana Vasile for helping me in proof reading of my thesis. I am also thankful to my colleagues in the SemanticLIFE project team with whom I had very useful discussions regarding ontologies and its usage in lifetime store which lead to successfully finish my thesis. I am thankful to Shuaib Karim, Khalid Latif, Shah Khusro, Hoang Huu Hanh, and Khabib Mustofa.

Lastly, and most importantly, I wish to thank my parents, AurangZeb and Maryam Saddiqa. They raised me, supported me, taught me, and loved me. To them and to my brothers Aqeel, Tauseef and my sister Irum I dedicate this thesis.

I am especially thankful to Higher Education Commission (HEC) Pakistan for their generous funding for my studies in Austria. Austrian Exchange Service (OeAD) and the ASEAN-EU Network Initiative are also praiseworthy.

Vienna, May 2008.
Mansoor Ahmed

# Table of Contents

# LIST OF FIGURES

# List of Listing

# Chapter 1

# Introduction to Web, Semantic Web and Security and Privacy Concerns

## 1.1 Today's Web

Current Web technology is highly human dependent. This means that for accomplishing tasks, such as looking for an apartment, getting price information for different goods, checking flight and railway information, finding some hotel etc, humans have to interact with computers, looking for some specific pages and retrieving the required information. In order to search for information such as "list me the good visiting places near my house" or "which is the closest grocery store to my working place", one needs to use keywords and to search through search engines (e.g. Google[1] or Yahoo[2]) which are linked to the websites being searched. Such tasks cannot be accomplished by computers on behalf of humans. Moreover, searching for certain information, browsing through links and finally retrieving the information and keeping track of such information are time consuming tasks. But if the same task were done by a software agent, it would be less time-consuming and more efficient.

The traditional Web (i.e. World Wide Web) is a system of interlinked, hypertext documents accessed via the internet. With a Web browser, a user views Web pages that may contain texts, images, videos, and other multimedia and navigates between them using hyperlinks [4]. Figure 1.1 shows the data structure in traditional Web.

[1] www.google.com
[2] www.yahoo.com

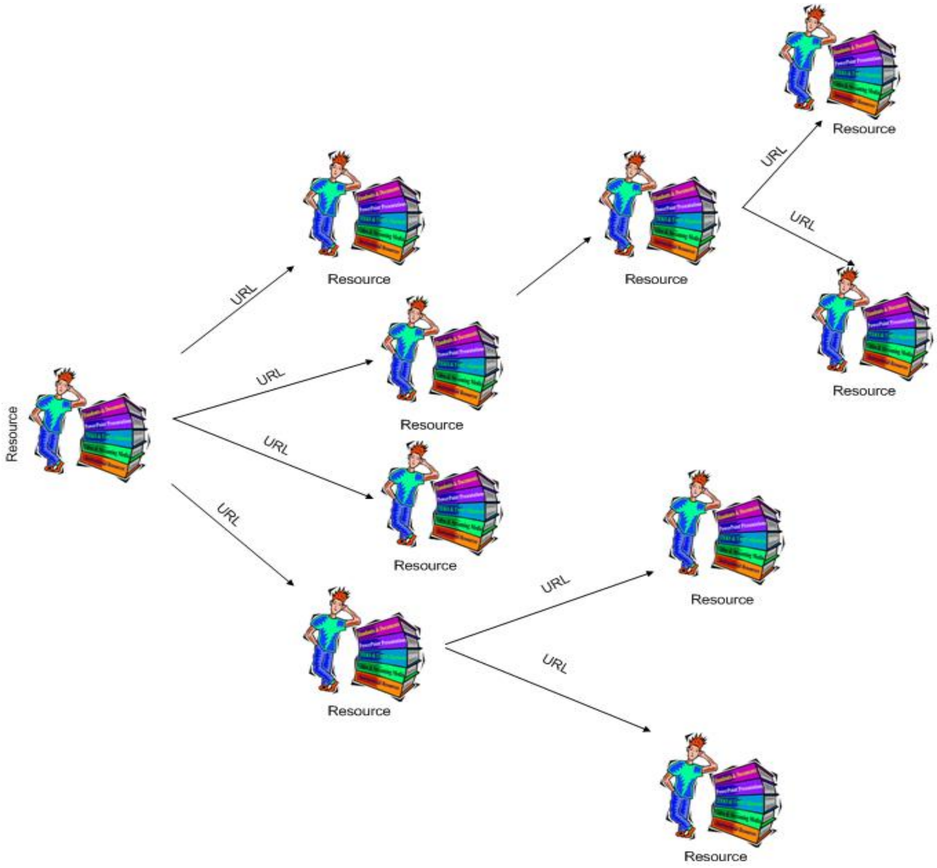


**Fig 1.1: Data Structure in Traditional Web by W3C**

## 1.2 A Motivation Example Scenario for Semantic Web

To have a broader view of what Semantic Web is and how it works and which characteristics make it different from the traditional Web, a general motivation example scenario which is used very often in literature is illustrated as follows.

Consider a person is trying to find a document with the author's name "Bob". When this query is submitted to traditional search engines, the search engines will give results which contain all the documents where the name "Bob" is mentioned along with the documents where people has referenced it. The key problem is finding the articles written by a particular author rather than those which include the author's name. In the Semantic Web, which is a new generation of WWW, it is possible to annotate a document, which means one can annotate documents by whom it was written, what it contains and when it was written. By adding such information, one will be able to find the exact required information. The additional information which is mentioned above is known as the *"metadata"*, which means data about data. Figure 1.2 depicts how the Semantic Web technology will enrich the entities.

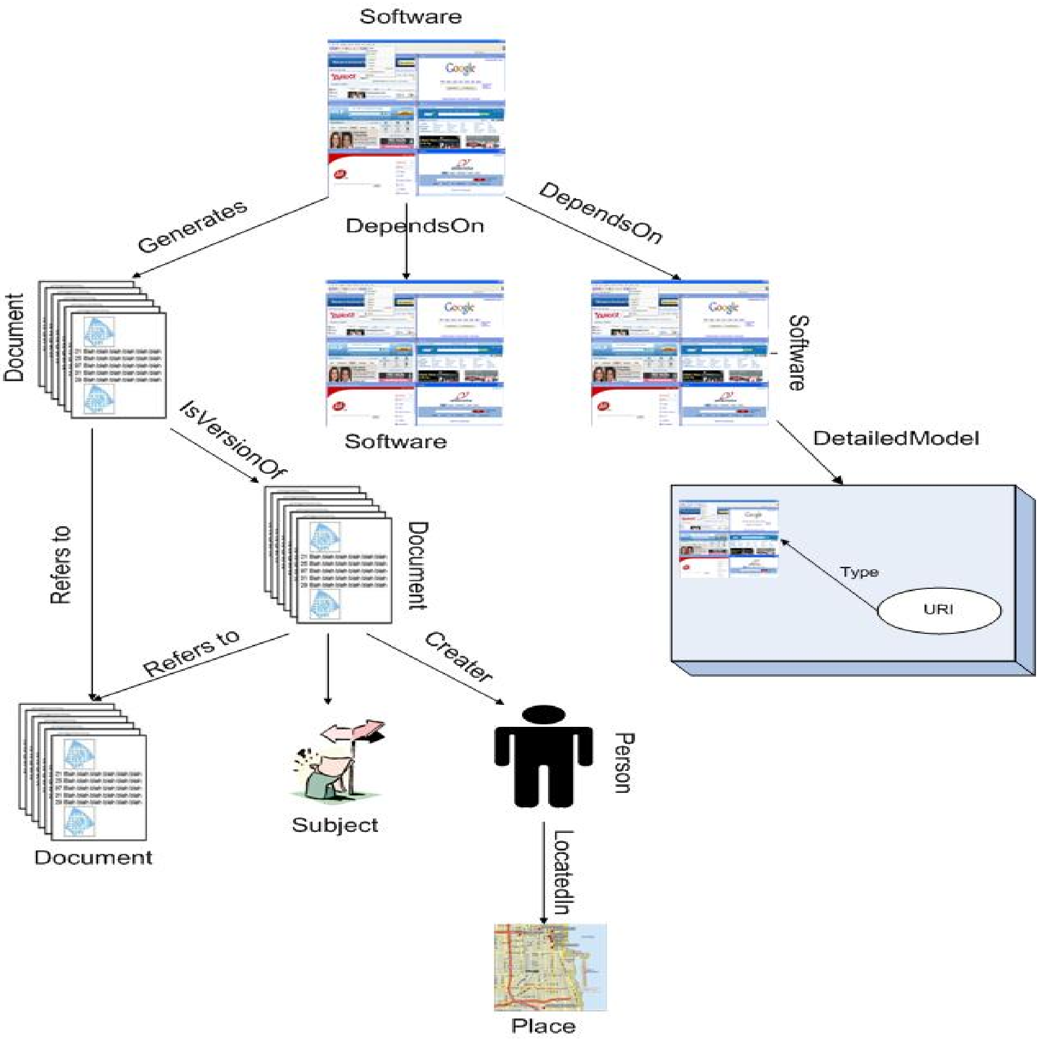


**Fig 1.2: Semantic Web Technology by W3C**

## 1.3 Classic Semantic Web Vision

The Semantic Web is inspired by a vision of the current Web which was influenced by the earlier work dating back to Vannevar Bush's idea of the 'memex' machine in the 1940s (based on a universal library, complete with a searchable catalogue) [1]. The founder of the World Wide Web Sir Tim Berners-Lee, originally envisioned the idea of a WWW in which documents were given a good description and links [2]. As a result, the human-oriented Web, which we know as World Wide Web, was introduced.

In May 2001 Sir Tim Berners-Lee introduced a bigger vision of the Web, i.e. the Semantic Web in [3]. In this article, he provided a convincing vision of the Web, which states that instead of humans laboriously trawling through information on the Web and negotiating with each other directly to carry out routine tasks, such as scheduling appointments, finding documents and locating services, the Web itself can do the hard work for them. Thus the notion of semantics being part of the Web arises, capturing the reason things are there.

People's mediation will no longer be needed the moment when the Web is enriched with a mechanism which defines semantics about the Web resources and links. This makes it possible for machines (software agents- programs working on behalf of people to locate the right things and make decisions) to understand the resources and links and work on behalf of the people. In the words of the article:

> *"The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users."*

The Semantic Web intends to make information understandable and processable by machines using common standard RDF and to make it available to humans. With the Semantic Web it is also possible to share and combine information on the Web.

### 1.3.1 Semantic Web Future Directions:

The aim of the Semantic Web is to allow a much more advanced knowledge management system. This technology promises to the user that [45]:

- Knowledge will be organized in conceptual spaces according to its meaning.
- Automated tools will support maintenance by checking for inconsistencies and extracting new knowledge.
- Keyword-based search will be replaced by query answering, requested knowledge will be retrieved, extracted and presented in a human-friendly

way.

- Query answering over several documents will be supported.
- Defining who may view certain parts of the information (even parts of the document) will be possible.

Tim Berners-Lee originally expressed the vision of the Semantic Web as follows:

> *"I have a dream for the Web in which computers become capable of analyzing all the data on the Web the content, links, and transactions between people and computers. A Semantic Web, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The intelligent agent's people have touted for ages will finally materialize* [46]".

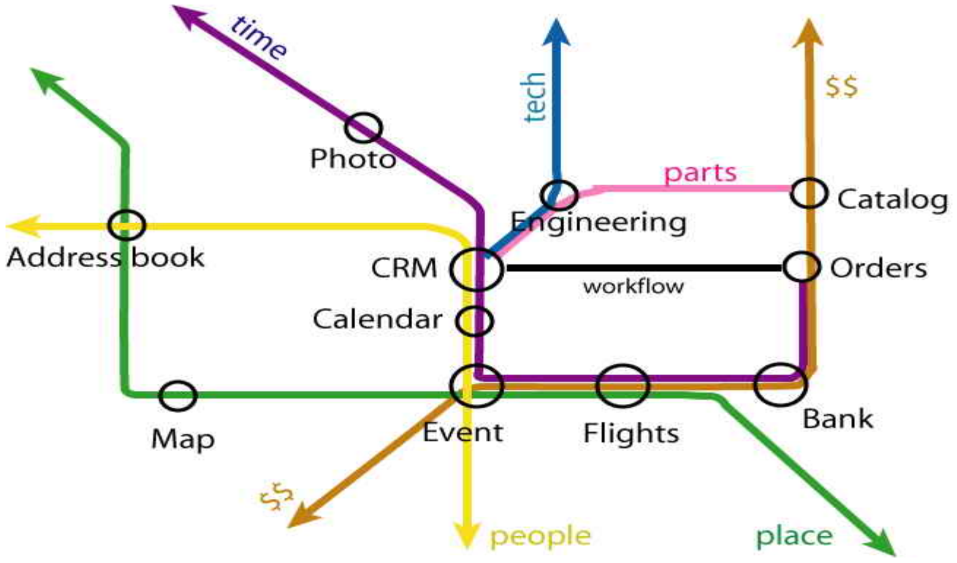


**Fig 1.3: Applications Connected by Concepts by Tim-Burners Lee**[3]

Figure 1.3 depicts the complete picture of the Semantic Web applications vision. The ultimate output of the Semantic Web is to improve the experience of the Web application end-user.

[3] http://www.w3.org/2003/Talks/05-gartner-tbl/slide21-0.html

*"People keep asking what Web 3.0 is. I think maybe when you've got an overlay of scalable vector graphics - everything rippling and folding and looking misty - on Web 2.0 and access to a Semantic Web integrated across a huge space of data, you'll have access to an unbelievable data resource* [49]*".*

In brief, the Semantic Web will provide data located anywhere on the Web and it will be processable and understandable by both machines and humans. This is more a vision than a technology. The technologies which will play an important role in making the Semantic Web dream come true are mentioned as follows:

- ***Explicit metadata:*** Metadata is data about data, i.e. data context. It describes an individual data, content items or a collection of data. Metadata is used to manage and facilitate the understanding of data. It allows Web pages to show their contents (what they are about). For example, on the Web page of a university, metadata can identify contacts, phone numbers, members, positions, courses, publications, etc.
- ***Ontologies:*** Ontologies describe the main concepts of a domain and the relationships among them. For example, vehicle ontology may contain concepts like car, wheel, truck, license number, engine, model year etc, and relationships such as subclass information, e.g. all cars are vehicles.
- ***Logical reasoning:*** With logical reasoning it becomes possible to setup the consistency and correctness in data sets. Also the inferred results can be drawn from these data sets. In other words "a given a precondition, a conclusion, and a rule where the precondition implies the conclusion". In the Semantic Web it is achieved by combining metadata of rules with ontologies.
- ***Agents:*** Agents are pieces of software capable of existing independently and of working preemptively on behalf of humans. The basic act of a personal agent is to seek information from the Web resources, communicate with other agents, compare information about user requirements and preferences, select certain choices, and give answers to the users.

## 1.4 Semantic Web Cake Layered Approach

The Semantic Web has been developing a layered architecture. Sir Tim Burners-Lee has specified different layers of the Semantic Web which are shown in Figure 1.4. Brief descriptions of technologies in layered approach are mentioned below:

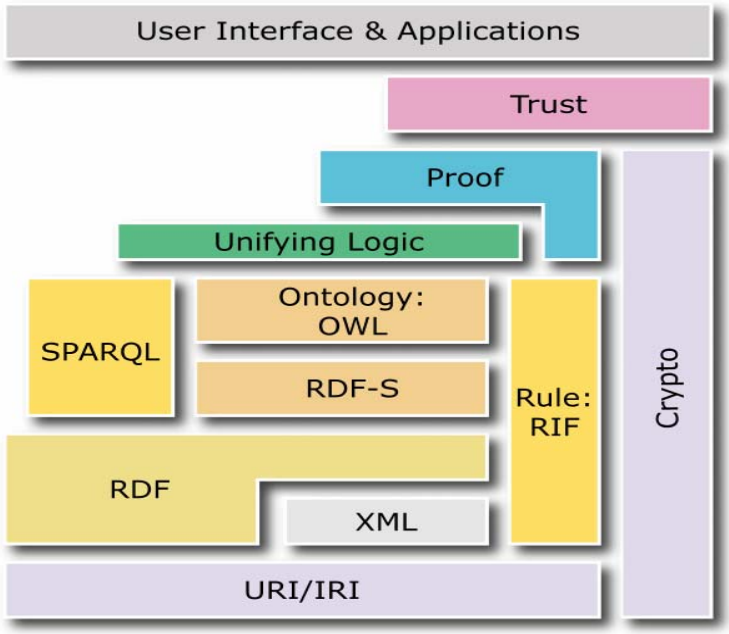


**Fig 1.4 Semantic Web Cake Layer by W3C**

- **Unicode and URI:** Unicode provides a unique number for every character. This standard is adopted by big companies, i.e. IBM, HP, Microsoft, Sun etc., whereas URI is a compact string of characters used to identify or name a resource. URI's are used in the World Wide Web to identify resources using a specific protocol.
- **XML:** XML is a general purpose markup and extensible language which allows the users to define their elements. XML facilitates the sharing of structured data across different platforms. The current Web technology is one such blessing using this technology.

- **Resource Description Framework:** The very first layer in the Semantic Web cake layer is RDF. RDF is W3C specification which is designed for modeling metadata. The RDF metadata model is composed of statements about resources in triple form, i.e. Subject-Predicate-Object. The subject refers to the resource; the predicate refers to aspects or properties of the resource and expresses the relationship between subject and object.
- **RDF Schema:** RDF Schema is an extensible knowledge representation language for describing classes of resources and properties between them. RDF Schema provides reasoning framework for inferring types of resources and provides basic vocabularies for specifying RDF application languages to use. It also provides basic elements for the description of ontologies.
- **Ontologies:** Ontologies defines representational characteristics for modeling knowledge domain. Ontology languages provide complex constraints on the types of resources and their properties.
- **Logic and Proof:** On top of the ontologies layer is a logic and proof layer which is used for automatic reasoning. It makes new inferences which will help the software agents to make decisions if a particular resource satisfies its requirement or not.
- **Trust:** The final layer of the Semantic Web cake layer addresses the issue of trust in the Semantic Web. Unfortunately this issue has been considered an afterthought, trustworthiness of the information on the Web is very important in order to provide people an assurance of its quality.

## 1.5 Layer by Layer Approach to Secure Semantic Web

As shown in the Semantic Web cake layer, the top layers are trust and proof. But security cuts across all the layers [5] .In the past few years there has been a lot of development in the Semantic Web [6]. It is essential that the Semantic Web should be secure, which means that its components, i.e. XML, RDF and ontologies, should be secure. The following steps are meant to make the Semantic Web secure [5]:

**Layer 1:** The bottom layer in the Semantic Web cake is TCP/IP, which deals with the communication on the Web. TCP/IP and HTTP are the data communication protocols on the Web. To have secure communication on the Web, it is necessary to have secure sockets (SSL) and secure HTTP.

**Layer 2:** The following layer possesses XML and XML schema technology. In the Semantic Web, the document structure will be based on XML technology and it will be possible to have access control on certain parts of the document, which means that certain parts of the document are accessible to certain people while they are restricted to others, e.g.  browsing, modification, reading and deletion etc. [7, 8].

**Layer 3:** The next important technology which needs good protection is RDF. Securing RDF means securing the semantics. Consider an example where person ‘x ‘publishes his/her information on the Web; he/she also wants that not all the contents of the page should be available to everybody (e.g. email address, mobile number, home address). In this case, the person needs to annotate the triples or group of triples to mention the security level for reading his/her Web page. Maybe his/her colleagues have full access to that page, while other people can only see a portion of it. The architecture for RDF security is described in detail in [9].

**Layer 4:** Once these layers are protected, the next layer is the ontology layer. Ontologies play an important role in fulfilling the dream of Sir Tim Berners-Lee the Semantic Web. Ontologies are used for knowledge sharing. OWL is the emerging language for representing ontologies. One way to secure ontologies is to annotate a part of the ontology with security level to make it secure, while other part being shared. For example, if a group of people is working on a project and the project manager gives annotation to a part of the project to be not visible to person ‘x’ in the same project, then the real problem which arises is how to deal with such information integration.

**Layer 5:** While discussing issues to secure Semantic Web, one should not forget about the inference problem. Inference means posing queries to deduce new information. It is difficult to take measures against a person who queries the information box and comes across the deduced information which he/she is unauthorized to know. There is lot of work going on inference issues on the Semantic Web [10, 11, 12].

## 1.6 Data Privacy and Security

Privacy and security are every person's basic rights. In our daily life we come across many situations where we have to make decisions regarding privacy and security, e.g. filling an online form on the Web, creating an email account, using any Web Service etc. 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" [13]. For example, an organization "X" is allowed to collect information while I am using their services, but they can keep my information only for one month.

Security refers to "The prevention of or protection against access to information by unauthorized recipients and intentional but unauthorized destruction or alteration of that information" [14]. Security is associated with three core areas, which can be conveniently summarized by the acronym "CIA" [14].

- ***Confidentiality:*** Ensuring that the information is not accessed by unauthorized persons
- ***Integrity:*** Ensuring that the information is not altered by unauthorized persons in a way that is not detectable by authorized users
- ***Authentication:*** Ensuring that the users are the persons they claim to be.

## 1.7 Comparison of Security and Privacy Issues in WWW vs. Semantic Web

While dealing with daily life security and privacy scenarios, we sometimes consciously do not give importance to certain elements such as name, email-ID, contact number, national identification number, home address etc. While divulge personal information, we happily pass the information to other parties because we either trust the third party, or we believe that the piece of information we unwrap is not very important. But what we don't realize is that these tiny pieces of information when gathered can become useful information and can be a threat to one's identity.

There are two major factors which causes privacy problems on the Web [99]:

1. The inherently open, nondeterministic nature of the Web
2. The complex, leakage-prone information flow of many Web-based transactions that involve the transfer of sensitive personal information.

The current Web technology captures privacy information in the form of cookies, collecting information through online registration, software downloads, Trojan horses, IP addresses, Web beacons and screen scraping which monitor the user's activities on the Web [15]. To overcome the privacy problems in the Web, W3C took an initiative by introducing the Platform for Privacy Preferences (P3P) project [16].

The rapid growth in information systems has resulted in computerizing applications in various domains. Individuals can store nearly every kind of digital information in their systems. In organizations, where data plays an important role, it has become easier to share information. In organizations with a project development environment, it is useful if there is a possibility of sharing information and access to data among different project members. While discussing security issues in the Semantic Web, one should not forget the importance of privacy. Privacy means making some part of your document public while keeping the rest as private. Privacy issues have gained a lot of attention recently, especially in the area of personal information management.

Unfortunately, the consideration of security and privacy in Semantic Web has been considered and afterthought, but we have strong conviction that this should be considered right from the evolution process so that necessary measure could be taken timely. It is important to provide assurance to people the trustworthiness of the information, its quality, security and privacy of sensitive data on the Web. In the literature this problem has been addressed at numerous places [17, 18, 19].

Privacy plays an important role in situations where agents will be interacting with each other for the retrieval of information. Privacy can be maintained effectively by making use of the Semantic Web technologies. In contrast to the existing Web, the Semantic Web allows to describe resources in the form of triples where group of triples can be annotated as public or private. For example, suppose that *Alex* has published his resume as an RDF document and annotated that certain fields, such as phone number and postal address, should not be disclosed. When a *recruiting agent* comes across that resume, only the parts which are declared public by *Alex* will be shown [20]. Nowadays the best practiced way on the Web for describing privacy policies is P3P which uses XML to describe policies in a machine-readable format [21].

As discussed above the role of agents in Semantic Web, Moreover the functionality of agents is to seek information from Web resources, communicate with other agents, compare information according to the user's requirements and preferences, select certain choices, and reply to the user. Consider an example where a person is planning to arrange a journey to Vienna for a conference. The personal agent will interact with different agents, e.g. ticket reservation, hotel reservation etc, to get the best deal and will return the information to the user. For the confirmation of the reservation, the other agent needs a certain amount of information, i.e. user credit card information, home address, email address, home phone number, passport number etc. In this scenario, the entities do not know each other in advance, so the traditional access control models cannot work in such environment. We need to have access control models which are enriched with semantics. Policy-Based access control models provide sophisticated means to support sensitive information disclosure where a person can semantically annotate the information with privacy/ security policies to regulate the information disclosure.

In the following two paragraphs we will investigate the security and privacy issues in Semantic Web and also if the traditional methods for security and privacy are sufficient to protect personal information in Semantic Web. If they are not, then we will investigate the merits and demerits of existing approaches, using [90].

The Semantic Web is considered as personalized Web where negotiation among users could be performed directly to carry out routine tasks, such as scheduling appointments, finding documents and locating services. The Web itself can do the hard work for humans. Since there will exist exchange of information between different people, it will give birth to a collaborative environment. As a result, information privacy and access control are very important. Traditional access control models cannot fulfill these requirements anymore. For example, a well known traditional access control model is Identity-Based access control which assumes that parties are known in advance and the machine needs to know the identity of the requester to allow or deny access to any resource. This kind of access control cannot work in a Semantic Web environment where people interact with each other automatically. Hence, there is a need for semantically enriched access control models for the automatic authorization to sensitive information.

Distributed access control models which were introduced in the past do not fulfill the requirements for Semantic Web yet because policies are bound to public keys mechanisms and they are not expressive enough to deal with Semantic Web scenarios [23]. Prior access control models like DAC (Discretionary Access Control) and MAC (Mandatory Access Control) models don’t support access control to sensitive information in Semantic Web environment due to lack of semantics. Moreover, a well known access control model, i.e. RBAC (Role-Based Access Control) model in which different roles are created according to job description and permissions are assigned to specific roles does not fulfill the Semantic Web requirements since it is hard to manage assigning roles to users which are not known in advance. The W3C’s initiative for privacy on Web (P3P) is not expressive enough to cope with the Semantic Web requirements to deal with the sensitive data and limiting access to resources since it is not a language just schema and it only describes the purpose of the gathered data and it doesn’t allow the enforcement mechanism. There are a number of policy languages which has been developed till today [24, 25, 26, 27, 28].

## 1.8 Ontology, Ontology Languages and Processing Tools

### 1.8.1 Exploitation of Ontologies in Semantic Web

The Semantic Web is an extension of the current Web in which information is given well defined meaning and machines will be able to understand and interpret the results. The Semantic Web is perceived as a personalized Web where people will be able to share their calendars, workspace, locate services and find documents. The role of ontologies in the Semantic Web comes into action for encoding meaning into Web pages. As described earlier in section 1.7, agents will play an important role, working on behalf of humans to fulfill jobs. So these agents, while surfing the Web can understand the contents of the Web page and therefore they will provide humans with more useful concerted services [29].

The traditional methods of information exchange and business transactions on the Web are changing from single isolated device to distributed information networks. Therefore, there is a need for support on the Web for the knowledge and data exchange. Ontologies provide a shared and common understanding of a domain knowledge that can be communicated between people and application systems [30].

### 1.8.2 Ontology Terms and Definitions

The term ontology has been used in different places in the literature. Below are some definitions found in the literature.

### Ontology in terms of Philosophy

In philosophy, ontology studies the nature of being and existence. The term ‘ontology’ is derived from the Greek words *“onto”*, which means *being*, and *“logia”*, which means *written or spoken discourse*. It is the study of being or existence and forms the basic subject matter of metaphysics. It seeks to describe or posit the basic categories and relationships of being or existence to define entities and types of entities within its framework [47].

The term ontology has been in use for quite a long time. Different people defined this term in different ways. In the words of the Webster dictionary, ontology is defined as:

- A branch of metaphysics relating to the nature and relations of being
- A particular theory about the nature of being or the kinds of existence.

“Although the terms "ontology" and "metaphysics" are far from being univocal and determinate in philosophical jargon, an important distinction seems often enough to be marked by them. What we may call ontology is the attempt to say what entities exist. Metaphysics, by contrast, is the attempt to say, of those entities, what they are. In effect, one’s ontology is one’s list of entities, while one’s metaphysics is an explanatory theory about the nature of those entities” [100].

Smith reviewed the metaphysical aspects of ontology from Aristotle’s time and drew the conclusion that they “provide a definitive and exhaustive classification of entities in all spheres of being” [31]. Keeping in view the work done in the field of ontology, Quine’s work gave a dimension to this research towards the formal theories in the conceptual world [32]. Gruber further extended Quine’s work into a new interpretation of ontology as “a specification of a conceptualization” [33].

“Ontology should be seen only as an interdisciplinary involving both philosophy and science. It is a discipline which points out the problems of the foundations of the sciences as well as the borderline questions, and which further attempts to solve these problems and questions. Ontology is not a discipline which exists separately and independently from all the other scientific disciplines and also from other branches of philosophy. Rather, ontology derives the general structure of the world; it obtains the structure of the world as it really is from knowledge embodied in other disciplines” [22].

> *“Recently, the term of "(formal) ontology" has been taken up by researchers in Artificial Intelligence, who use it to designate the building blocks out of which models of the world are made. An agent (e.g. an autonomous robot) using a particular model will only be able to perceive that part of the world that his ontology is able to represent. In a sense, only the things in his ontology can exist for that agent. In that way, an ontology becomes the basic level of a knowledge representation scheme.”*

In terms of Computer Science, according to Horrocks and Sattler

> *“An ontology is an engineering artifact constituted by a specific vocabulary used to describe a certain reality. A set of explicit assumptions regarding the intended meaning of the vocabulary. Thus an ontology describes a formal specification of a certain domain: shared understanding of a domain and an interest; formal and machine manipulation of a domain of interest.”*

In general, ontology describes a domain of discourse formally. Typically, ontology consists of a finite list of terms and relationships between terms. The terms denote the important concepts (classes of objects) of the domain.

For example, in a university setting, staff members, students, courses, lectures theaters, and disciplines are some important concepts. In the context of the Web, ontologies provide shared understanding of a domain. Such a shared understanding is necessary to overcome differences in terminology. Figure 1.5 shows the ontology hierarchy of university people [45].

From these ontology definitions, we can conclude some important aspects of ontologies, such as:

1. They are used to describe a specific domain.
2. The meaning of the term is used consistently.
3. The terms in an ontology are organized in a proper hierarchical structure, i.e. IS-A or HAS-A relationship.

4. The domain of ontology contains specification of terms and relationships among them.

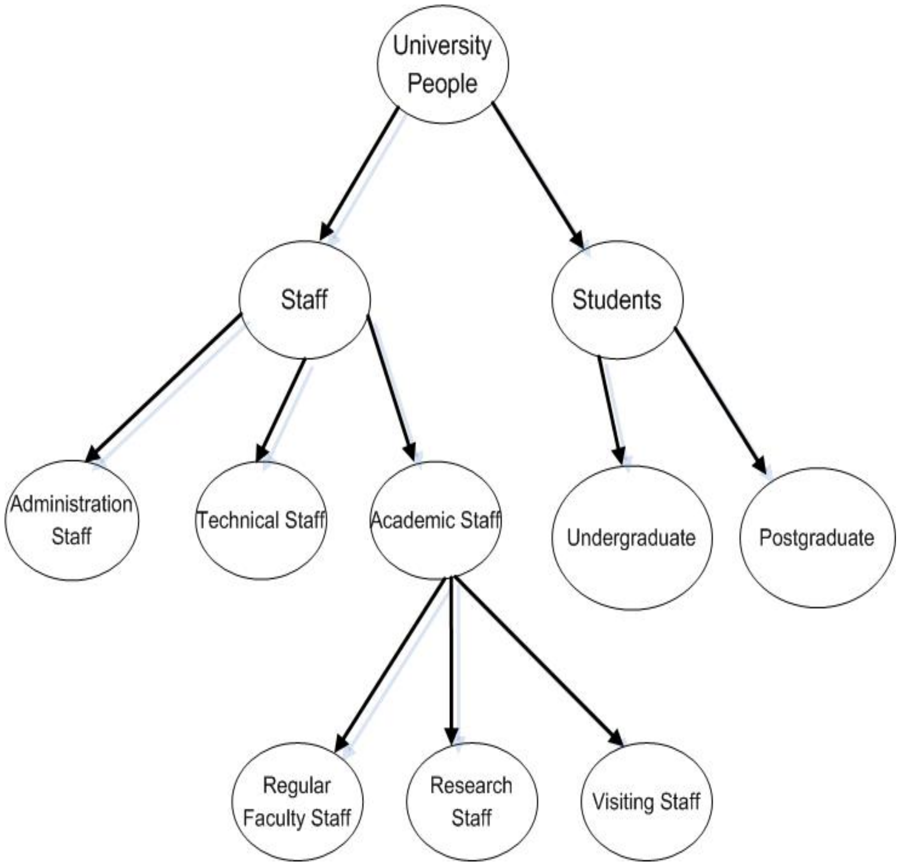


**Fig 1.5: University People Hierarchy**

## 1.9 The Semiotic Triangle

Ogden & Richard (1923) presented the famous semiotic triangle (Figure 1.6) which states that the referent of an expression may vary according to different language users [34]. The two solid edges in a semiotic triangle represent casual relations of “symbolization” and “reference” to the casual relation. The relation between symbol and referent is shown by the line which stands for the imputed relation. According to Pierce "A sign, or *representamen*, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign.

That sign which it creates I call the *interpretant* of the first sign. The sign stands for something, its *object (*or referent). It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the *ground* of the representamen." (Peirce, 1931-1958, 2, 228) [35].

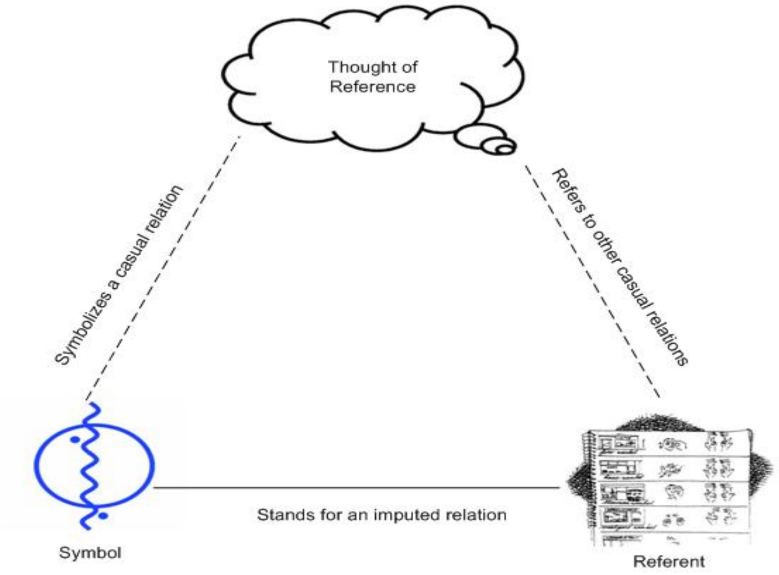

Thought of Reference
Symbolizes a casual relation
Refers to other casual relations
Stands for an imputed relation
Symbol
Referent

**Fig 1.6: The Ogden and Richards (1923) Semiotic Triangle**

Sowa (2000) dilates the idea of Ogden & Richards' (1923) semiotic triangle of meaning as "The semiotic triangle (Figure 1.6) has a long history. Aristotle distinguished objects, the words that refer to them, and the corresponding experiences in the *psychê*. Frege and Peirce adopted that three-way distinction from Aristotle and used it as the semantic foundation for their systems of logic. Frege's terms for the three vertices of the triangle were *Zeichen* (sign) for the symbol, *Sinn* (sense) for the concept, and *Bedeutung* (reference) for the object [36]".

## 1.10 Ontology Languages

There are different ontology languages which exist now. Bellow is a brief description of each language.

### RDF[4]

Resource Descriptive Framework was invented to overcome the deficiency of the XML language which describes the information structure and which fails to define the semantics in a machine understandable format.

RDF is based on XML format, i.e. its syntax is defined in XML. It is used for adding meta-information to the Web documents. RDF data model consists of three object types:

- ***Resources:*** Resources are denoted by URIs. A resource can be anything, e.g. a part of a Web page, a complete Web page or a collection of Web pages.
- ***Properties:*** A property is a specific characteristic, attribute or relation to describe a resource.
- ***Statements:*** A specific resource together with a named property plus the value of that property for that resource is an RDF statement.

In RDF terminology, these three individual parts are known as subject, predicate and object. Where the subject refers to resource, predicate refers to traits or aspects of the resource. In brief, RDF defines triples (object-property-value*)* that represent the semantics of Web resources and introduces a standard syntax for them.

### OIL, DAML+OIL and OWL

In recent years, lots of efforts have been made for developing an expressive ontology language for the Semantic Web. Below we shortly present the most important developments in the area of ontology languages. These developments include OIL, DAML+OIL and most importantly OWL.

### OIL[5]

Ontology Inference layer or Ontology Interchange Language is known as an ontology infrastructure for the Semantic Web. OIL was developed as part of the European IST project On-To-Knowledge. The syntax definition uses RDF(s) and

[4] http://www.w3.org/RDF/
[5] www.ontoknowledge.org

XML(s) in order to maintain backward compatibility.

The basic features of OIL are as follows [44]:

1. It provides most of the modeling primitives commonly used in frame-based and Description Logic (DL) oriented Ontologies;
2. It has simple, clean and well defined first-order semantics;
3. Automated reasoning support, (e.g., class consistency and subsumption checking) can be provided.

The main contribution of OIL is that it provides the means for describing structured vocabulary with well defined semantics.

### DAML+OIL[6]

DARPA Agent Markup Language + Ontology Interchange Language are a combination of two languages. It is a semantic markup language for Web resources created as a joint effort of the American and European ontology communities for the Semantic Web by merging DAML-ONT and OIL.

DAML+OIL exploit existing Web standards (XML and RDF) by adding ontological characteristics of object-oriented and frame-based systems and formal rigor of expressive description logic. It implements an object-oriented approach, with the structure of the domain being described in terms of classes and properties, and the set of axioms that assert characteristics of these classes and properties [98].

### OWL[7]

Ontology Web Language is an initiative of the W3C Web Ontology Working Group. It consists of a description of classes along with their related properties and instances. OWL ingredients are OIL and DAML+OIL, therefore OWL characteristics are very much common.

OWL consists of three main components[8]:

[6] http://www.daml.org/
[7] http://www.w3.org/TR/owl-features/

- ***Individuals:*** Represent objects in the domain that we are interested in. It must be explicitly stated that individuals are the same as each other, or differ from each other, otherwise they might be the same as each other, or they might be different from each other.
- ***Properties:*** Properties are binary relations on individuals, i.e. properties link two individuals together.
- ***Classes:*** Classes are interpreted as sets that contain individuals. They are described using formal descriptions that precisely state the requirements for class membership. OWL consists of three flavors, i.e. OWL-Lite, OWL DL and OWL Full.

## 1.11 Ontology Language Processing Tools

- **OWLJessKB**[9] is a description logic reasoner for W3C's Ontology Web Language (OWL). The semantics of the language is implemented using Jess, the Java Expert System shell. OWLJessKB is a successor to DAMLJessKB and its features are based on the Jess Rete inference engine [37].

- **Jena**[10] is developed at HP Labs at Bristol [38]. Jena is an open-source Semantic Web framework for Java which provides inference support for OWL, RDF and RDFS (except for blank node types) allowing users to create customized rule engines. Rule-based inference environment is also supported in this framework. The Jena framework includes the following characteristics:
    - A RDF API
    - Reading and writing RDF in RDF/XML, N3 and N-Triples
    - An OWL API
    - In-memory and persistent storage
    - SPARQL query language

[8] http://www.w3.org/TR/owl-guide/.

[9] http://edge.cs.drexel.edu/assemblies/software/owljesskb/

[10] http://jena.sourceforge.net/

- **FaCT++**[11] is developed at the University of Manchester and is the new generation of the well-known FaCT OWL-DL reasoner [39] [40]. It is implemented using C++ in order to create a more efficient software tool and to maximize portability. The current version provides full support to OWL-Lite, whereas in future, its enhanced version will provide complete support for OWL-DL reasoning.

- **Racer**[12] reasoner is based on description logic reasoning [41]. It handles large Aboxes in combination with large and expressive Tboxes. It supports inference over RDFS/DAML/OWL ontologies through rules explicitly specified by the user. It provides OWL support with rules, constraint reasoning and expressive query answering. nRQL is the query language.

- **Pellet**[13] developed at the University of Maryland is an open-source Java based OWL DL reasoner that can deal with both TBox reasoning and non-empty ABox reasoning [42]. Pellet reasoner can be used with both Jena and OWL API libraries and also provides DIG interface. It provides OWL reasoning for SWOOP ontology editor and manages in-depth ontology analysis and repair [43]. Other features include datatype reasoning, user-defined simple datatypes, multi-ontology reasoning using E-connections and ontology debugging.

## 1.12 Ontology Development Tools

Below is a list of some of the most common editors used for building ontologies:

- **WebOnto**[14] is a Java applet connected with a customized Web server which allows users to browse and edit knowledge models over the Web.

[11] http://owl.man.ac.uk/factplusplus/
[12] http://www.sts.tu-harburg.de/~r.f.moeller/racer/
[13] http://www.mindswap.org/2003/pellet/
[14] http://kmi.open.ac.uk/projects/webonto/

- **GKB-Editor**[15]. Generic Knowledge Base Editor (GKB) is a tool which provides graphical interface for browsing and editing knowledge bases across multiple Frame Representation Systems (FRSs) in a uniform manner. Also, the intuitive user interface presents objects and data items as nodes in a graph, with the relationship between them forming the edges.

- **OilEd**[16] was developed at the University of Manchester. With the help of this ontology editor, users can build ontologies using DAML+OIL.

- **OBO-Edit**[17] is an open source ontology editing tool for editing and constructing OBO format ontologies. It is a platform-independent and graph-based tool which facilitates the biologists with a user-friendly interface for viewing and constructing ontologies.

- **Protégé**[18] is a free and open source ontology editor and knowledge-base framework based on Java and it provides plug-n-play environment. It is developed by Stanford Center for Biomedical Informatics Research at Stanford University School of Medicine. Ontologies can be modeled in two main ways, i.e. protégé-frames and protégé-OWL editors. Protégé editor supports exporting ontologies in RDF(S), OWL and XML schema.

## 1.13 Ontology Storage and Retrieval

- **Sesame**[19]

Sesame is a Java framework for querying and inferencing the RDF-based repository. It was developed in the Netherlands as one of the key deliverable in the European IST project (On-To-Knowledge). The system consists of a repository, a query engine and data is managed (added or deleted) through an administration module. It can be used either as a Web server or as a Java library.

[15] http://www.ai.sri.com/~gkb/
[16] http://oiled.man.ac.uk/
[17] http://www.geneontology.org/GO.tools.shtml
[18] http://protege.stanford.edu/
[19] http://www.openrdf.org/

It supports different query languages, i.e. SeRQL, SPARQL etc. Sesame can store large quantities of RDF and RDF Schema information [84].

- **Jena**[20]

Jena is developed by Hewlett-Packard and it contains a collection of RDF tools written in Java, such as an RDF parser (supporting an N-Triple filter), a query system based on RDQL and a Java model/graph API [85].

A widely-used scheme for storing RDF statements in a relational database is the *triple store*. In this approach, each RDF statement is stored as a single row in a three column 'statement' table. Typically, a fourth column is added to indicate if the object is a literal or a URI. A common variation of this scheme, which uses much less storage space, is the *normalized triple store* approach. This scheme uses a statement table plus a literals table and a resources table [86].

- **KAoN**[21]

KAoN is developed at Karlsruhe University in the context of the Semantic Web infrastructure. KAoN-API was developed using Java and it is used to access ontologies. The main-memory based implementation of the KAoN-API maps directly onto the RDF-API (an implementation of graph-model for processing RDF). Since there exist different languages for the ontologies development, KAoN-API tries to be a representation language neutral [87] [88].

- **RDFStore**[22]

RDFStore is a pure Perl implementation of a model centric API over RDF constructs.  It inherits most of its class definitions from the Draft Java API from the Stanford University Database Group by Sergey Melnik and from the RADIX proposal by Ron Daniel [89].

The storage system allows transparent storage and retrieval of RDF nodes, arcs and labels from a variety of storage systems, i.e. either from an in-memory structure, from the local disk or from a very fast and scalable remote storage. It

[20] http:// jena.sourceforge.net/
[21] http://kaon.semanticweb.org/
[22] http://rdfstore.sourceforge.net/

supports several different persistent storage models such as SDBM and BerkeleyDB. RDFStore implements the SquishQL language to query RDF repository and all query-filtering operations on the values are processed using Perl regular expressions.

- **Kowari**[23]

Kowari is an Open Source, massively scalable, transaction-safe, purpose-built database for the storage, retrieval and analysis of metadata. Kowari is written in Java. Kowari supports Resource Description Framework (RDF) and Web Ontology Language (OWL) metadata.

Kowari features include multiple databases (models) per server, simple SQL-like query language, full text search functionality, datatype support etc. From the point of view of permanence, kowari is optimized for metadata storage and retrieval, multi-processor support, independently tuned for both 64-bit and 32-bit architectures, low memory requirements and steamed query results.

## 1.14 Logic and Inference in Semantic Web: Rules

The field of Knowledge representation is as old as before the emergence of World Wide Web, in the area of artificial intelligence and ancient Greece history to Aristotle, who is known as the father of logic. Predicate logic (known as first-order-logic) is still the foundation of knowledge representation. The reasons for its importance are as follows [46]:

- It provides a high-level language in which knowledge can be expressed in a transparent way and it has high expressive power.
- It has well-understood formal semantics, which assigns an unambiguous meaning to the logical statement.
- There is a precise notion of logical consequence, which determines whether a statement follows semantically from a set of other statements. In fact, the

[23] http://www.kowari.org/

primary original motivation of logic was the study of the objective laws of logical consequences.

- There exist proof systems that can automatically derive statements syntactically from a set of promises.
- There exist proof systems for which semantic logical consequence coincides with syntactic derivation within the proof system. Proof systems should be sound (all derived statements follow semantically from the premises) and complete (all logical consequences of the premises can be derived in the proof system).
- Predicate logic is unique in the sense that sound and complete proof systems do exist. More expressive logics (higher order logics) do not have such proof systems.
- Because of the existence of the proof system, it is possible to trace the proof that leads to a logical consequence. In this sense, the logic can provide explanations for answers.

RDF and OWL (Lite and DL) can be viewed as specializations for predicate logic. The correspondence was illustrated by the axiomatic semantics in the form of logical axioms.

Another subset of predicate logic with efficient proof systems comprises the so-called rule system (also known as Horn logic or definite logic programs).

A rule has the form

$$A1,\ldots\ldots An \longrightarrow B$$

Where *Ai* and *B* are atomic formulas. In fact, there are two intuitive ways of reading such rules:

1. If *A1,…..An* are known to be true, then *B* is also true. Rules with this interpretation are referred to as *deductive rules*.
2. If the condition *A1,…..An* are true, then carry out the action *B*. Rules with this interpretation are referred to as *reactive rules*.

## 1.15 Organization of Thesis

The organization of thesis is described as follows:

- Chapter 1 states the difference between current Web and Semantic Web technology, comparison of securing data, a brief introduction to Semantic Web technologies and tools used to build these technologies.
- In Chapter 2 describes related work pertaining to Ontology-based security and privacy.
- Chapter 3 gives an overview to problem statement.
- Chapter 4 consists of some use case scenarios.
- Chapter 5 describes the research project SemanticLIFE which was conducted at Institute of Software Technology and Interactive Systems, Vienna University of Technology.
- Chapter 6 mentions our work related to Ontology-based risk assessment in collaborative environment using the SemanticLIFE system.
- In Chapter 7 introduces data privacy in Web Services context using Pipeline architecture of the SemanticLIFE system.
- Chapter 8 gives the conclusion and future work.

# Chapter 2

# Related Works

## 2.1 Brief Introduction to Personal Information Management Systems and their Security Approach

**2.1.1 MyLifeBits**[24] offers traditional access control lists (ACLs) for files and role-based access control (RBAC) for the database.

**2.1.2 Haystack**[25] *(MIT)* plan to implement the trust layer just beneath the UI and RDF store. Then it will be possible to manage the list of trusted users by the users themselves. Consequently only those RDF statements which are duly signed by these trusted users will provide information to UI, and not all.

**2.1.3 e-Person**[26] uses role based access control (RBAC) for hosting services,. Message communication at transport layer supports message signing and verification without using a full public key infrastructure (PKI), authority, authentication and end-to-end security. But it also creates problems when there are slight changes at transport layer message format. Statement level access control is provided for RDF store. Roles associated with allowable information patters, are assigned to trusted

[24] http://research.microsoft.com/barc/mediapresence/MyLifeBits.aspx

[25] http://groups.csail.mit.edu/haystack/

[26] http://www.hpl.hp.com/personal/Steve_Cayzer/eperson.htm

users.

**2.1.4 Edutella[27]** uses the underlying transport layer security (TLS) which is common to JXTA framework. Secure communication between the peers is done by public key cryptography.

## 2.2 Ontology-Based Security and Privacy Related Work

An author in [56] presents an approach to corporate assets in a company when taking into account the entire infrastructure. The approach proposes a quick calculation of effective countermeasures using the security ontologies in Figure 2.1.

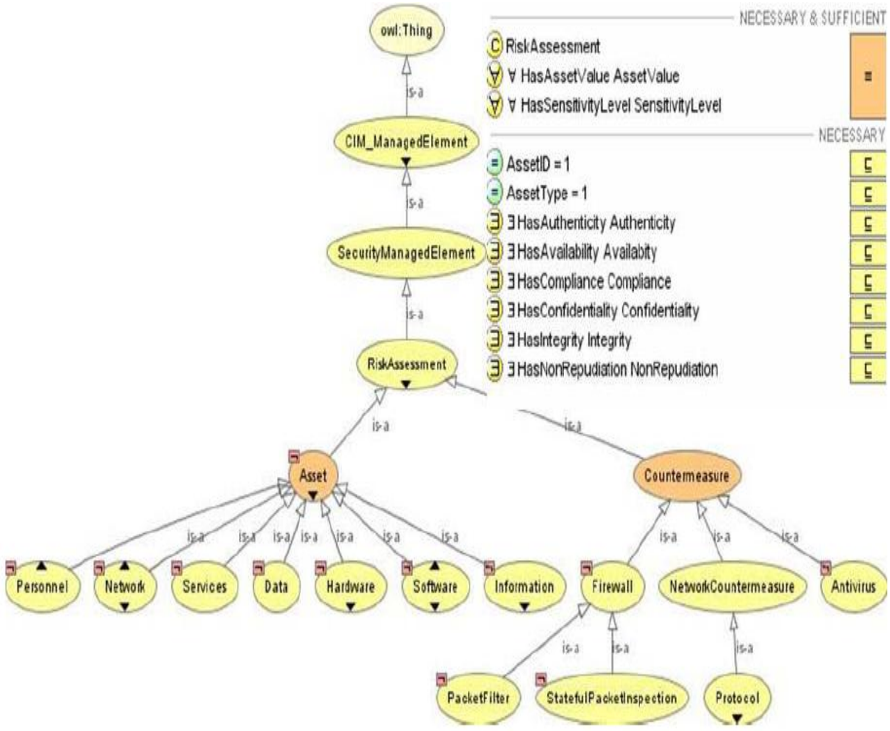


**Fig 2.1: Risk Assessment Security Ontology**

The company infrastructures such as computer, network, server, person, etc are taken into account in the measurement evaluation.

[27] http://www.edutella.org/edutella.shtml

However, the model uses some heuristic estimation parameters thus it could cause imprecise risk evaluation results. They also suggests a more a precise risk assessment model which also takes into account the semantic relation between the business objects into account.

[48] Proposes target-centric ontology for intrusion detection (Figure 2.2). The ontology specifies a model for computer attacks. Based upon empirical evidence the model of computer attacks are categorized by:

1. The system component targeted.
2. The means and consequences of attack, and
3. The location of the attacker.

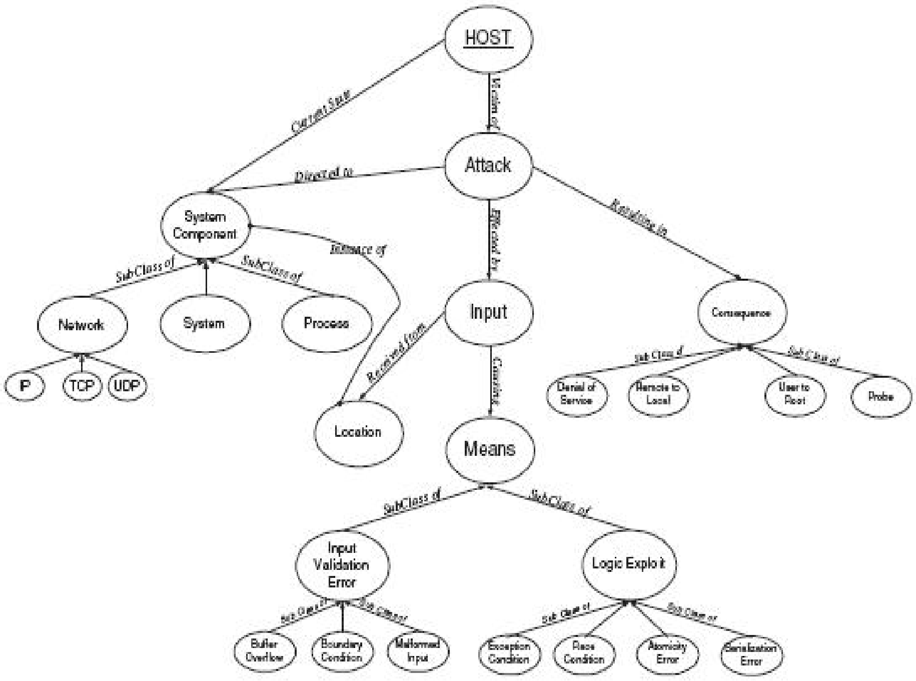


**Fig 2.2: IDS Ontology**

The model is presented as target-centric ontology, where the structural properties of the classification scheme is in terms of features that are observable and measurable by the target of the attack or some software system acting on targets behalf. This ontology is used to facilitate the reasoning process of detecting and mitigating computer intrusion.

Authors in [53] address security requirements in developing secure applications. They propose the use of ontologies for capturing and depicting the security expert's knowledge. Security ontology can facilitate the communication between security experts, users and developers. The security ontology depicted in Figure 2.3 shows security related issues. To furnish these task two application scenarios were taken into consideration i.e. e-tax and e-voting application.

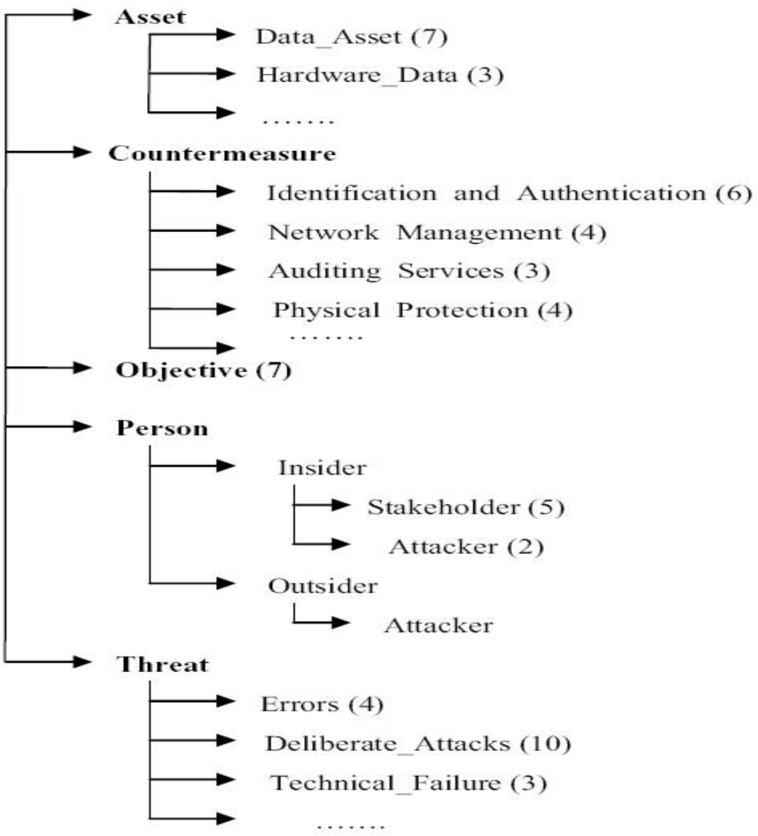


**Fig 2.3: Security Ontology Hierarchy**

In [59] authors propose security ontology based on the taxonomy of computer security and dependability by Landwehr. The proposed security ontology shown in Figure 2.4 provides solid base for an applicable and holistic IT-security approach for small and medium enterprises. According to their belief heavy weight ontology can be used to systematically structure knowledge on threats, safeguards and assets.

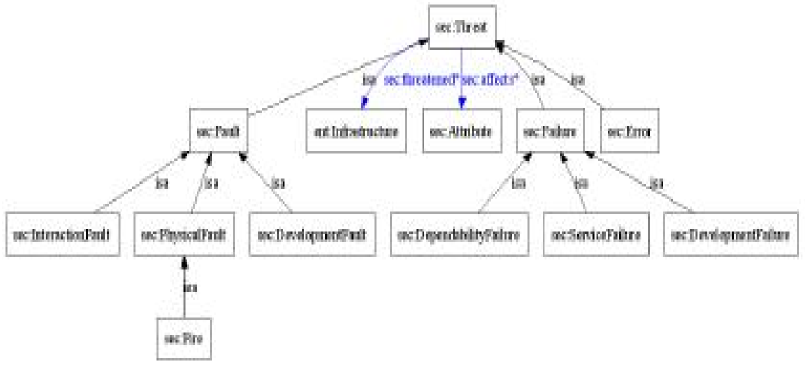


**Fig 2.4: Threat Ontology**

Security ontology for annotating resources is introduced in [50]. Types of security information that could be described include mechanisms, protocols, objectives, algorithms, and credentials in various levels of detail and specificity. This ontology is capable of representing more types of security statements and can be applied to any electronic resource. The ontologies were applied in Service Oriented Architecture to annotate security aspects of Web Service descriptions and queries. A refined matching algorithm was developed to perform requirement-capability matchmaking that takes into account not only the ontology concepts, but also the properties of the concepts.

The NRL security ontology comprises seven different ontologies. Each ontology covers different domain knowledge. Figure 2.5 and 2.6 shows part of these ontologies. Description is mentioned as follows:

1. Main security ontology describes the key security concepts
2. Credential ontology defines the authentication credentials
3. Security ontology describes various security algorithms
4. To specify different assurance standards security assurance ontology is used
5. To mark security annotation of semantic Web Services, service security ontology is created
6. Agent security ontology aims for querying of security information

7. For input and output parameters of Web services Information object ontology is used.

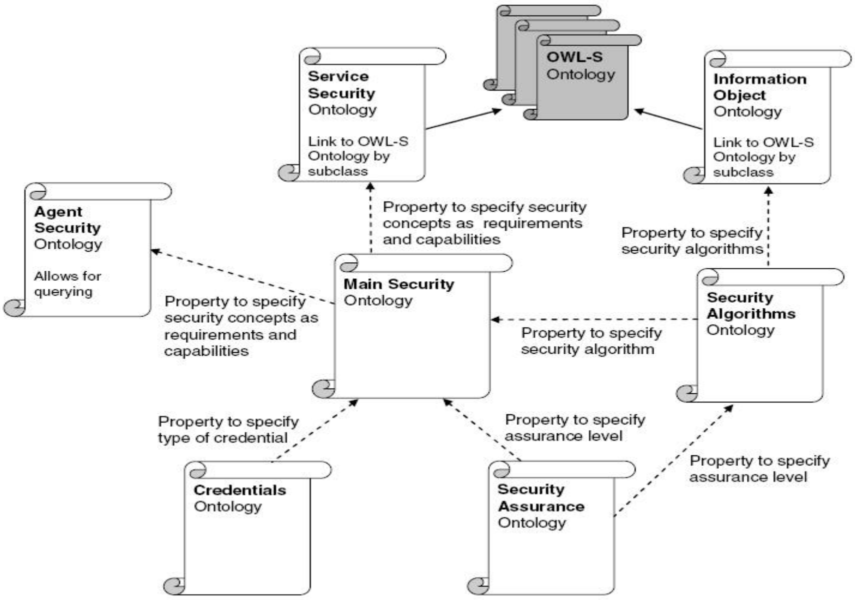

OWL-S Ontology
Service Security Ontology
Link to OWL-S Ontology by subclass
Information Object Ontology
Link to OWL-S Ontology by subclass
Agent Security Ontology
Allows for querying
Property to specify security concepts as requirements and capabilities
Property to specify security algorithms
Main Security Ontology
Property to specify security concepts as requirements and capabilities
Security Algorithms Ontology
Property to specify security algorithm
Property to specify type of credential
Property to specify assurance level
Property to specify assurance level
Credentials Ontology
Security Assurance Ontology

Fig 2.5: Security Related Ontologies and Their Relationships

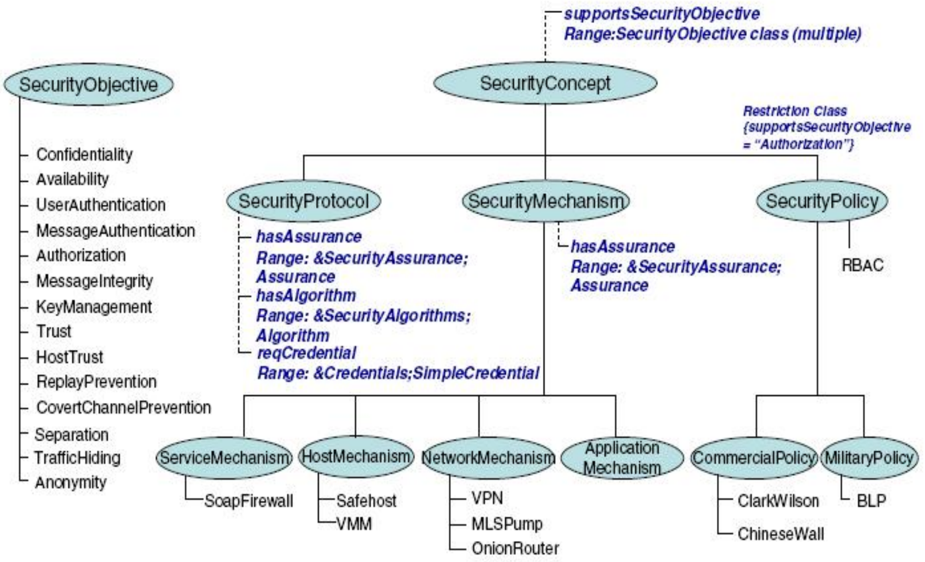


Fig 2.6: Part of Main Security Ontology

In [51] authors have introduced ontology for computer security domain (Figure 4.7). To automatically correlate security data from different sources, and to ease information management about security incident, they propose ontology for unique vocabulary of concepts and relations related to security incidents.

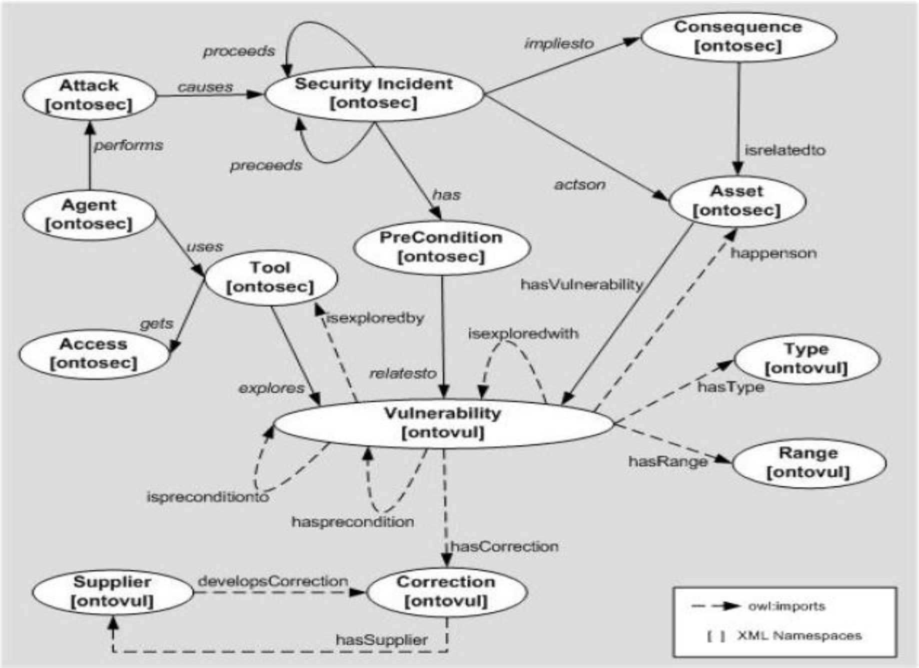


**Fig 2.7: Main Concepts and Relations of the OntoSec**

The authors in [52] proposes WS security threats and state that they have to be analyzed and classified systematically in order to allow the development of better distributed defensive mechanisms for WS using F/IDS.

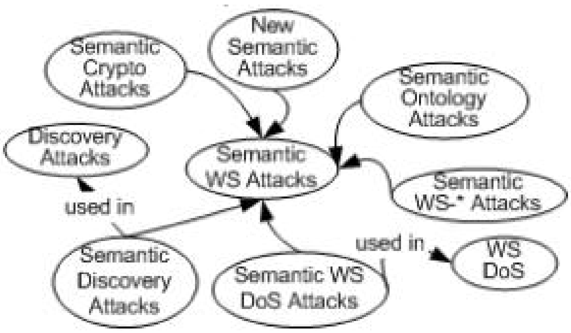


**Fig 2.8: Semantic Web Services Attacks**

They choose ontologies and OWL/OWL-S over taxonomies because ontologies allow different parties to evolve and share a common understanding of information which can be reasoned and analyzed automatically. They developed the security attack ontology (part of ontology is shown in Figure 2.8) for Web Services.

# Chapter 3

## Problem Statement

The rise in interconnectivity in the last few years has made computer systems and networks more vulnerable to threats as they are accessed by an ever increasing number of users. Nowadays organizations are lacking proper security measures and means to calculate risk assessment for their assets. Legacy systems in organizations are facing different kind of risks like viruses, bugs and system failure causing damages to hardware and software resulting in data loss. The ultimate challenge in many organizations is to assess their risk factors for their computers and networks. There is no way to completely overcome the threat that an organization might have. The goal is to calculate risks, so that problems resulting from them could be minimized and to fill the gap between business entities (like a project, a role) and organization infrastructure.

The growing number of Web Services technologies and their use has 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 processable services. One of the biggest challenges in both Web Services and Semantic Web concerns 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 essential. To acquire any service, one has to disclose personal information (e.g. home address, date of birth, mobile number, credit card information etc) so as to fulfill the requirements 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.

# Chapter 4

# Research Project: SemanticLIFE

## 4.1 Introduction

The enormous amount of existing knowledge and its immense continual growth have already surpassed the capabilities of conventional information storage and exploration techniques that one uses in managing one's lifetime information. Now the technology is at such a point that this enormous amount of information can be stored somehow, somewhere, but is not being used effectively and efficiently due to lacking semantics, absence of innovative information storage, retrieval and visualization techniques. Back in 1945 Vannevar Bush dreamed of an innovative personal information management system, Memex, as an enlarged intimate supplement to ones memory "in which an individual can store all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility".

The 'SemanticLIFE' project is an attempt to come a step closer to Vannevar Bush's vision of the Memex. Recently we can observe a mushrooming of new projects aiming at some of the goals of Bush's innovative ideas. This is mainly caused by the racy technological development which opens new large realization potentials. An indicator for the narrowing of the discrepancy between the visions of Bush's Memex and its realization is the announcement of *"Memories for life"* -- Managing information over a human lifetime as one of the seven Grand Challenges for Computing Research by the UK Computing Research Committee. Similarly, it constitutes one of the ISTAG Grand Challenges for Information Society Technology in the emerging 7th framework IST program of the European Union as a Blackbox for Humans capturing a Life Log. It aims at providing augmented episodic memory, security and aid for the population at large.

## 4.2 An Overview to 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.

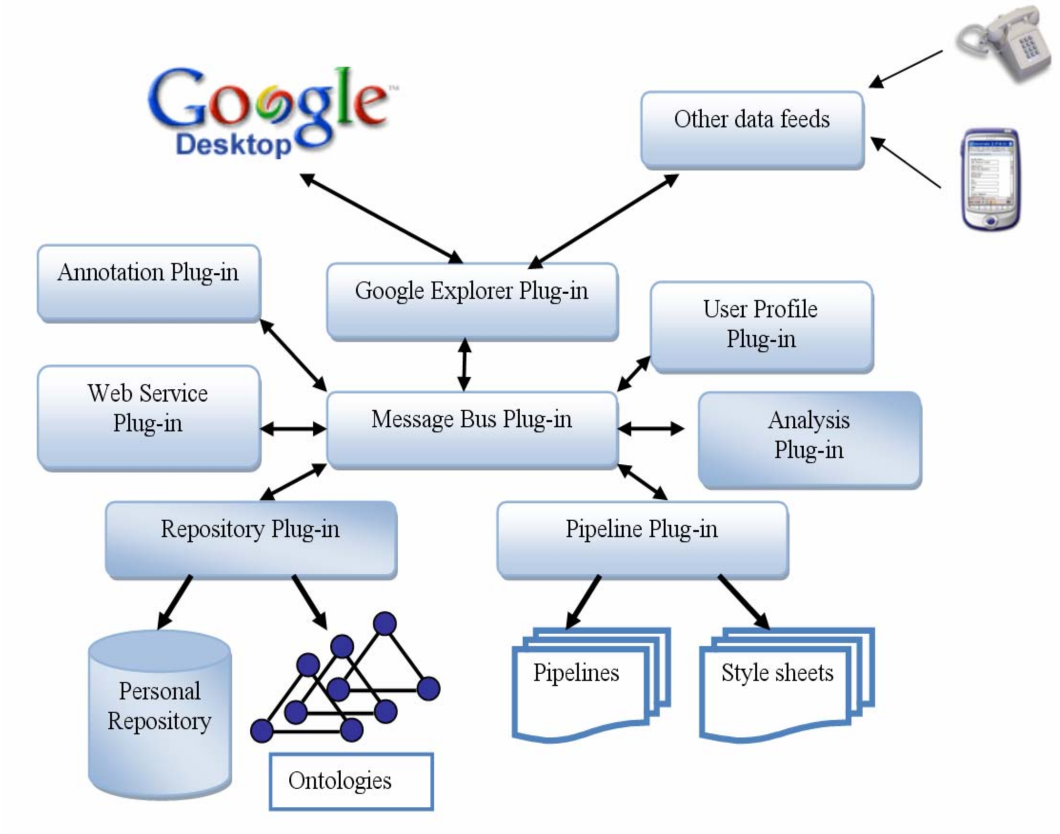


**Fig 4.1: SemanticLIFE Framework Architecture**

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 [93]).

An overview of the system architecture is depicted in Figure 4.1. 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.

## 4.3 System Architecture

### 4.3.1 Data Input

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. 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.

### 4.3.2 Communication Framework

SemanticLIFE framework is built upon of three main plug-in frameworks which communicate via messaging and collaboration components. The fundamental plug-ins which supports the system communication is Message Bus, Pipeline and Web Services plug-ins. Short description of each plug-in is mentioned below.

#### 4.3.2.1 Service Bus

Service Bus plug-in manages all information exchanges between the SemanticLIFE processes. This plug-in supply also a level of abstraction between systems services by providing a transparent, uniform access interface to all services.

Service Bus offers the extension point for service developers to publish their standard Java classes as Web Services. The standard extension point mechanism of Eclipse facilitate visual configuration of extensions with the extension provider. During the application start-up, the Services Bus loads all the connected services and automatically deploys them using embedded Jetty and Apache AXIS. The deployment scripts are created on the fly from the service description. Thus developers can, at the same time, benefit from Rich Client environment of Eclipse and Java Web services in the similar and coherent mechanism. Lastly, the Services Bus uses the standard WSDD [96] and WSDL [97] conventions for the service configuration.

#### 4.3.2.2 Web Services

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.

With Web Service plug-in, Web Services can be plugged at anytime to the SemanticLIFE system by locating the corresponding configurations (WSDL file’s url). More importantly, the plug-in supports capturing services semantic which are used in both “*locating appropriate services*” and “*ranking the competitor services*” tasks. The services semantic, defined in OWL-S standard describes the functions of the service in terms of the transformation effected by the accordant service. It also specifies required inputs, pre-condition to invoke a service, the generated outputs, and the expected effects that result from the execution of the service.

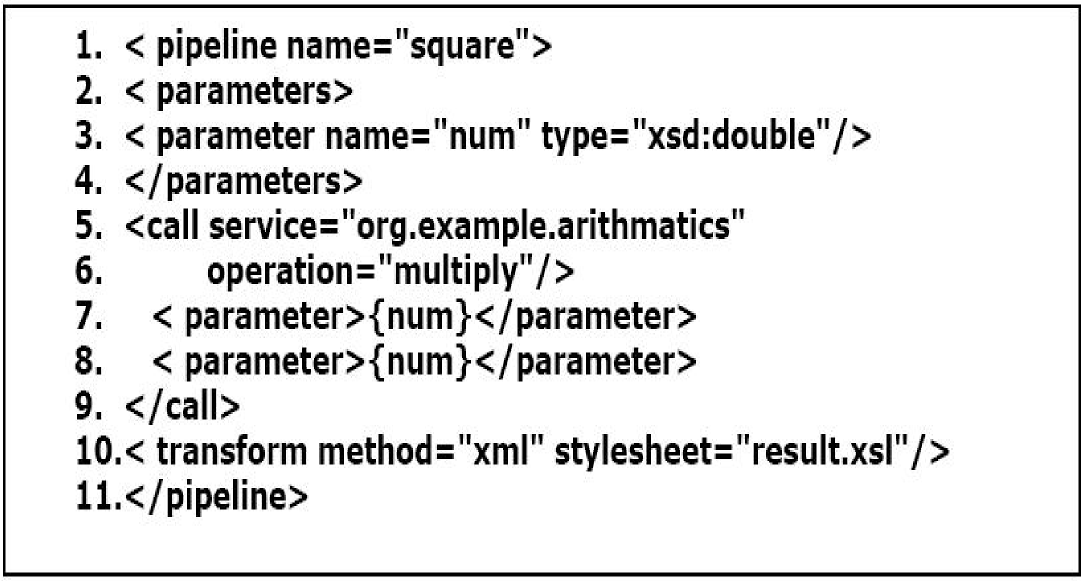
#### 4.3.2.3 Pipeline

Another fundamental plug-in is the pipeline plug-in that plays a central role in the orchestration of basic system services and in the creation of new business services. We introduce the notion of a pipeline as ***a uniquely named set of service-calls and intermediate transformations***. The pipelines are defined using an XML structure that specifies pipeline steps and relevant transformations.

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. ***A "Pipeline" in SOPA terminology is a uniquely named set of service-calls and intermediate transformations***. The pipeline plug-in enables the SOPA systems to realize scenarios based on the basic services and the pipelines. The newly created services (pipelines) may be shared with other users that may need the new service.

The pipeline idea has been inspired from Apache Cocoon [92] which is a Web development framework built around the concepts of separation of concerns and component-based Web development. Cocoon implements these concepts around the notion of *'component pipelines'*, each component in the pipeline specializing on a particular operation. This makes it possible to use a Lego(tm)-like approach in building Web solutions, hooking together components into pipelines without any required programming.

The pipelines and their corresponding structure are defined using an XML structure that specifies the pipeline components and relevant transformations. Listing 3.2 shows the basic structure of a typical pipeline:



```
1.  < pipeline name="square">
2.  < parameters>
3.  < parameter name="num" type="xsd:double"/>
4.  </parameters>
5.  <call service="org.example.arithmatics"
6.          operation="multiply"/>
7.     < parameter>{num}</parameter>
8.     < parameter>{num}</parameter>
9.  </call>
10.< transform method="xml" stylesheet="result.xsl"/>
11.</pipeline>
```

**Listing 4.2: A Simple Pipeline**

As shown above a pipeline is identified by its name (line 1). Each pipeline may receive some input parameters that might be used anywhere inside the pipeline's scope. Lines 2 to 4 show the parameter section and definition of a parameter called "num". The most interesting part of a pipeline which distinguishes our approach from other such solutions is the service-call part. At line 5 the "multiply" operation of the service *"org.example.arithmatics"* is requested. The operation call can consume parameters of the pipeline. It is important to mention that the services in a SOPA system are not limited to those provided by other plug-ins but also include pipelines and external Web Services (distinguished by complete end-point URI).

The results returned by the services may be transformed during the execution of a pipeline. This feature let the results be transformed and converted to required format. The transformation is performed by applying an XSLT transformation to the current pipeline results. The pipeline plug-in keeps the results internally and finally at serialization phase the results are rendered in required format. The supported serialization formats are TEXT, XML, HTML, and XSWT [95].

As explained in the previous section, the available services in the SOPA environment are routed via the Services Bus plug-in; i.e. all services will be requested from

Services Bus which is responsible for finding and then invoke the corresponding service to do the task. This feature provides a service transparency in the whole SOPA environment. As stated earlier the services in SOPA are not limited to plug-in exposed services but optionally may include pipelines and external Web Services. As a result the SOPA system brings the service orchestration scenarios to a new horizon. The business scenarios developed under eclipse programming framework can combine resources coming from internal or external components via a single service routing plug-in (Services Bus plug-in). Figure 4.3 depicts the service transparency.

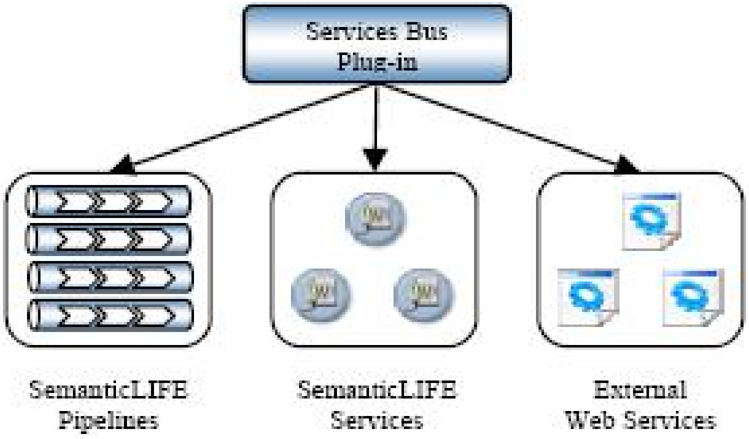


**Figure 4.3: Service Transparency in SOPA**

The created pipelines will be used by other system components and may provide a range of services covering the business logic, visualization features or a combination of these two. A pipeline, containing calls to some services and combines the results together to create new piece of information, can be documented and reused as a new business service. Visual rendering and styling of the results is also an edge of pipelines that combines the results of business processes with different visualization options. As a result a specific set of results can be rendered differently based on the context and user requirements.

### 4.3.3 Information Analysis

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.

### 4.3.4 Storage

The semi-structured and semantically enriched information objects are forwarded to the repository plug-in for an ontologically structured storage, called the meta-store.

### 4.3.5 Annotation

The SemanticLIFE system supports both free text annotations where user can give comments e.g. to a Web page or to any report or research paper regarding the quality and relative ness etc, while semantic annotations can be done against any group of triples in the store e.g. triples of picture, an email etc [94].

### 4.3.6 Visualization

Visualization plug-in provides user interface for our system which helps the end user to fulfil his/her tasks. This plug-in also possesses some nice visualizations.

# Chapter 5

# Motivation Examples: Use Case Scenarios

## 5.1 Scenario 1: Risk Assessment in Collaborative Environment

Consider a project development environment in an organization, where different people like programmers, developers, quality assurance and architects work on different projects distributed on different nodes. Organizations with such setup usually work in a distributed environment, which means the project is distributed on different nodes like the database server running at one node while the application server, versioning system like (CVS, SVN) and Web server at some other nodes.

On the other hand the organization employees play different roles and each of them accesses different resources with a specific security level. As a result the user access in the entire organization is the union of user-resource permissions for all user accessible resources. The administrators should always be aware of this spread of accesses and avoid overriding an access rule when adding or modifying a rule/access. There have been some attempt to undertake this complexity by introducing resource directories and uniform resource management protocols, but such approaches are not usually followed in heterogeneous system environments.

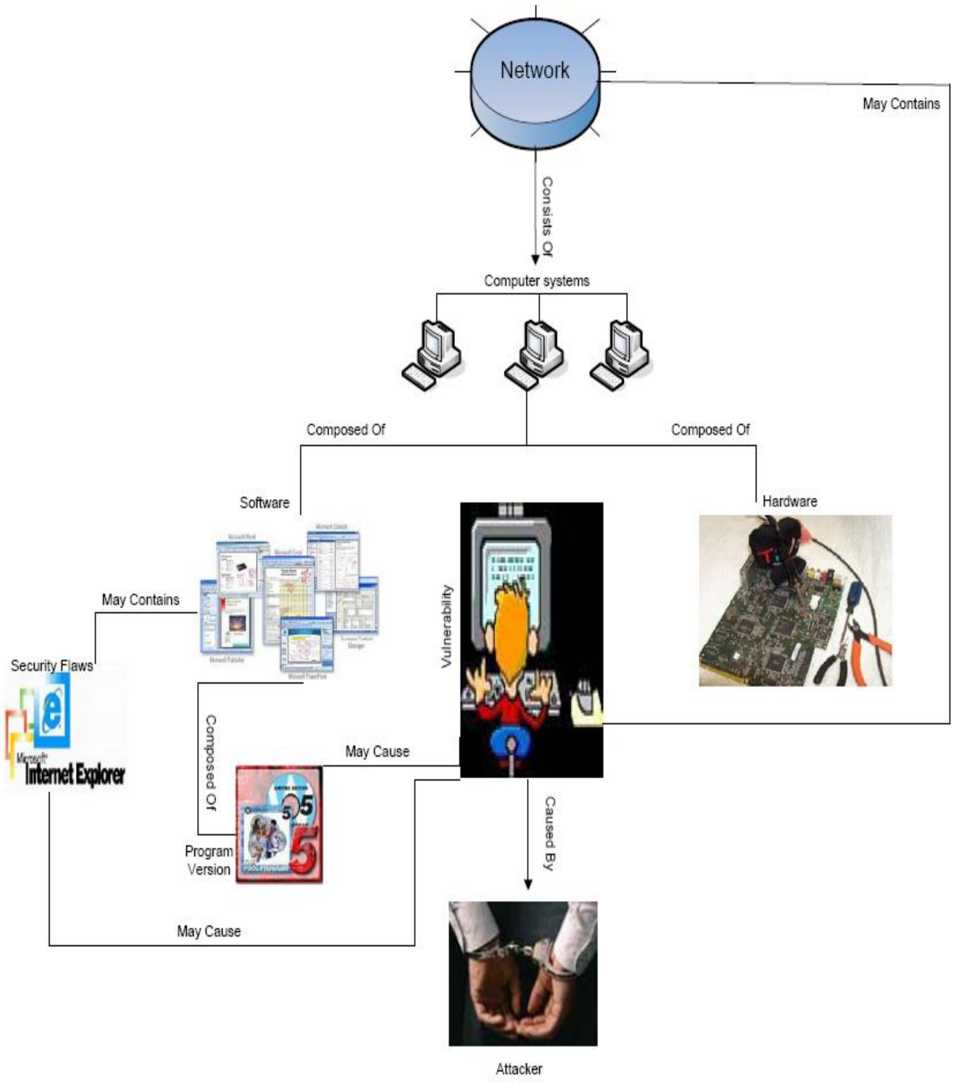


**Fig 5.1: A Typical Network Environment**

While working in networked environment it is quite possible that a node (individual computer or server) is attacked because of viruses, hackers, fire, vibrations, weak network policies or loopholes in software programs and operating system. For a node to be vulnerable, it has a precondition followed by an impact or aftereffects. There is a numbers of preconditions for a node to be exposed to vulnerability. A typical network environment is depicted in Figure 5.1.

Below are some use-cases for the type of attacks on a node in a networked environment followed by the reasons and aftereffects.

- Node “a” comes under virus attack, preconditions for this kind of attack are missing of a proper antivirus client, old virus definition or some patch is not updated.
- In networked environment open ports give passage to hackers to attack the network, resulting loss of information (confidentiality, integrity and availability). The precondition for this kind of attack can be data communication ports etc.
- Installation of malicious software on nodes can cause vulnerability. Intruders can get access easily. The precondition for such attack can be the installation of P2P communication software for data transferring.
- In organizations where the project is distributed on different nodes, concurrent versioning system plays an important role; the purpose of such system is to share the files in workplace. Sometimes due to weak rights, unauthorized person get access to confidential data.
- In an organization, Website is hosted at different location. For example, database of the Website at database server while information pages on some application server. If the database is not properly secured then attacker can penetrate causing damages.

Figure 5.2 and 5.3 gives detailed overview of project distribution on nodes and user assigned to nodes respectively.

There are some other means by which a node is exposed to vulnerability, like weak cryptography, inadequate password management and easy access to facility.

The impacts or aftereffects of the vulnerabilities explained above are destroyed files, exposed data, lost productivity, lost machine control, wasted IT staff time to rebuild machine.

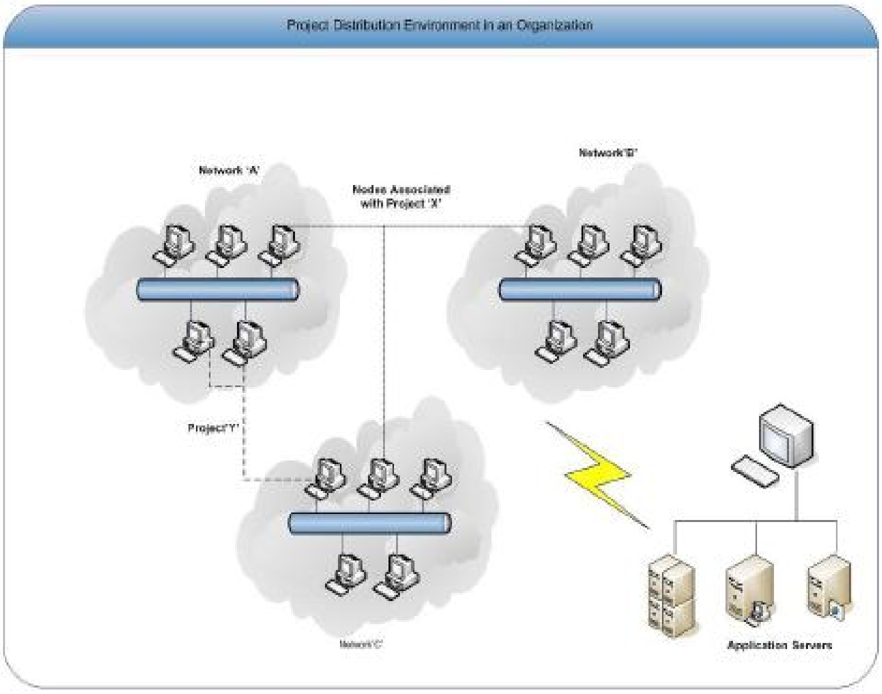

Project Distribution Environment in an Organization
Network 'A'
Network'B'
Nodes Associated with Project 'X'
Project'Y'
Network'C'
Application Servers

**Fig 5.2: Project Distribution on Nodes**

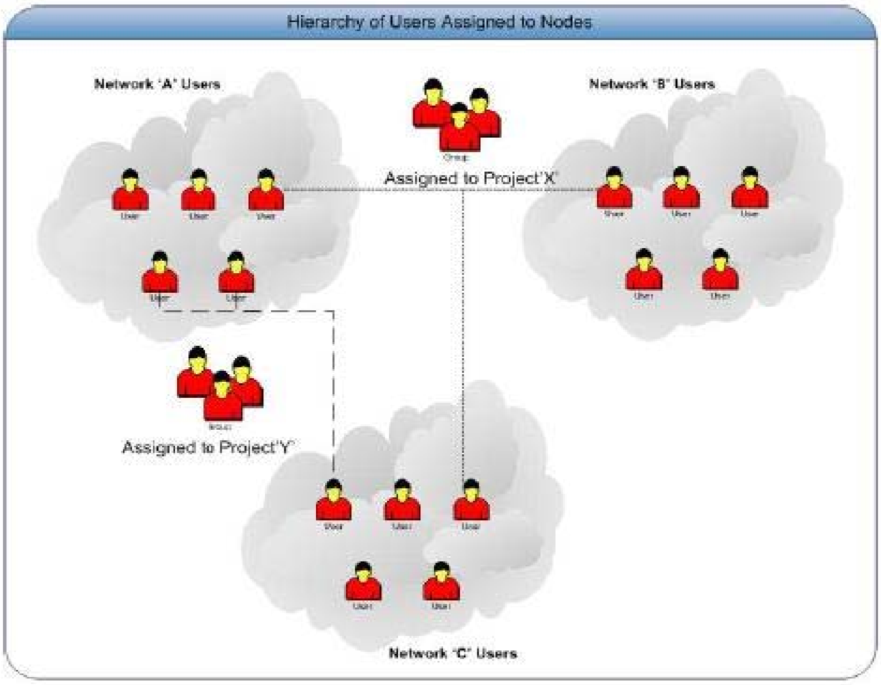


**Fig 5.3: Users Assigned to Nodes**

In a project development environment of an organization the interaction between different business entities like user, project, node and attack is depicted in Figure 5.4.

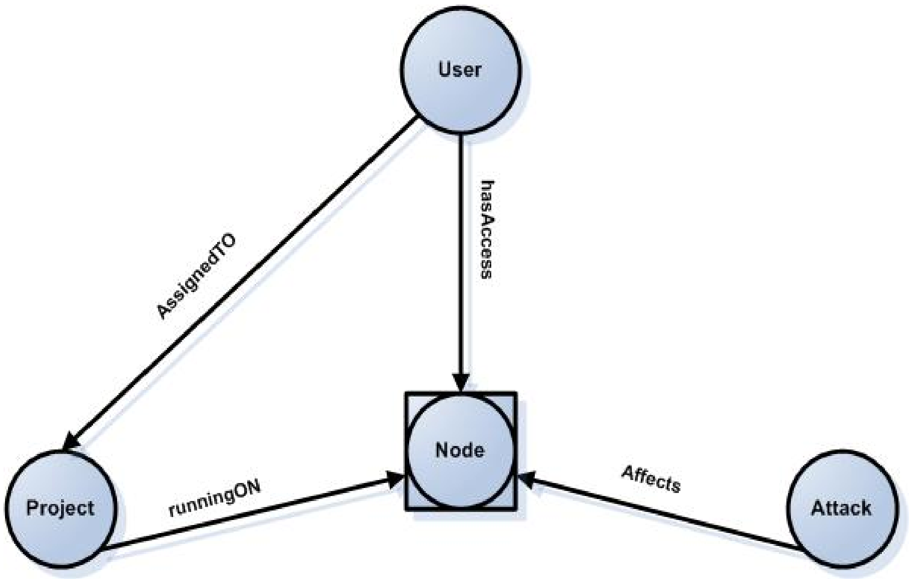


**Fig 5.4: Business Entities Relationship**

Users are assigned to projects. Projects are running at different nodes. To have access to projects, users are assigned to nodes with roles depending upon their job description. Nodes are attached to each other through network. In distributed project environment nodes are grouped into developer nodes, programmer nodes, manager nodes and server nodes etc. In network environment, nodes get exposed to attacks.

Various reasons for this node to get exposed to attack and vulnerabilities are explained above. In organizations, the ultimate solution is risk assessment of attacks, which provides a basis for the prevention of future attacks. In a nutshell the basic problem encountered in such environment is *"to manage some very dynamic creatures that are highly sensitive, distributed and interconnected"*.

## 5.2 Scenario 2: Personal Data Privacy in Web Services Context

In the use of Web Services there are two major entities involved in the exchange of information, i.e. the requester who requests to access some resource and the provider who provides the services 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 and in Figure 5.5:

1. In the first step, the user locates the desired Web service while searching through the UDDI registry.
2. After choosing an appropriate Web Service, the service provider asks the user to submit personal information in-order to have an access to a resource.

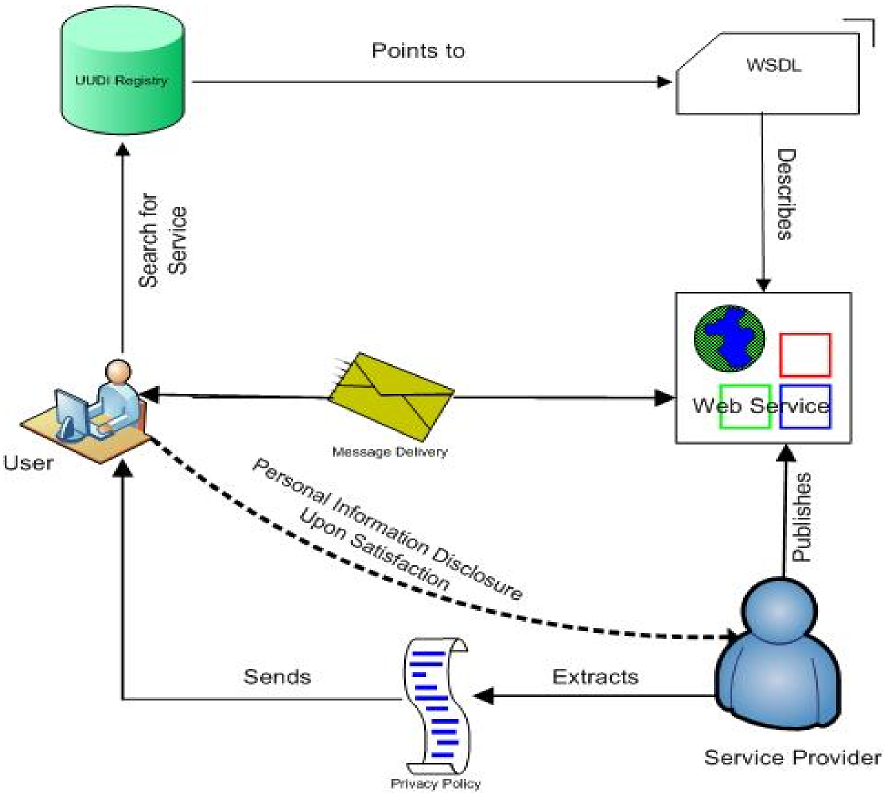


**Fig 5.5: Service Requester and Provider Interaction**

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. The user can agree or disagree with the privacy practice of the service provider.
5. The User evaluates the privacy practice information of the service provider and, if it satisfies him/ her, then information will be delivered.
6. When handshaking process is completed, then the information provided by the user is stored at service provider side according to the user's preferences.

# Chapter 6

# Ontology-Based Risk Assessment

## 6.1 Introduction

The widespread use of the Internet and Information advanced technologies in the various application domains results in the ubiquitous data distribution amongst different computing systems. People nowadays do not only work or access data on their own computing system, but they also need to work in other systems within the collaborative environment. The need for safe and secure systems that are capable of sharing the data safely, detecting the risks, preventing attacks, reducing the vulnerable effects etc becomes more and more essential, especially in collaborative environment.

Attacks could come in various ways depending on the environment in which the systems were deployed and the data they process. The user's non-authorized accesses to data, illegal packages execution, virus distribution, spam emails, server hacking, are some of examples of attacks that could happen every day. Each attack has different levels of danger and effects on the systems. Some attacks could be very dangerous, and affect the whole organization, while other limit only to the personal computer. Detecting and preventing attacks are time and budget consuming tasks in the administration of an IT-organization. The system administrators should track the changes in heterogeneously running systems with different installed software, operating systems, processes & applications.

Beyond all the complexities mentioned above, the system administrators should take care of the side effects of the combination of software and hardware configurations that might put the computers and the network at risk. On the other hand this data should be merged with other business aspects of an organization, like business processes, projects, tasks, roles, etc. Usually the system administrators are not that much aware of the business concepts and are more equipped with pure technical skills. As a result there is usually a gap between organizations' business entities and the software, hardware and the human resources. The Semantic Web technologies seem to be a good candidate to bridge this gap and assist the system administrators to manage and control the systems more smartly. On the other hand a semantic combination of entities-resources will enable the administrators and managers to minimize the side effects of their decisions that might put the organization at the risk.

Another major challenge is the information gathering about the physical entities (i.e. mail servers, Web servers, databases, personal computers) and the software, applications installed on them. Such information changes dynamically time by time and is also scattered in the organization. The relation between those entities (i.e. which computer/server belongs to the people from the same project, which process could cause domino effect if being stopped or killed due to its belonging to several entities, etc) should also be managed so that the suitable security policy could be issued within the organization. Closing the gap between physical entity information, the objects running, and semantic information of dependency relations between them to asset the attack risks are the purpose of our approach using SemanticLIFE in organizational risk assessment.

## 6.2 Background:

Risk assessment in Information Technology has been deeply investigated in [54]. This is a guideline to risk classification and risk evaluation in the general domain of the Information Technology System. The authors also state that reliably assessing information security risks can be more difficult than assessing other types of risks, because the data on the likelihood and the costs associated with information security risk factors are often more limited and because risk factors are constantly changing.

They propose some case studies on risk assessment methodology in various companies.

The author in [55] suggests a methodology which could extract, model and analyze the security requirements from multiple documents and then use the ontological process to infer valuable knowledge on system secure assurance. The approach, however, could only be possible in case there already exist well-established Certification and Accreditation (C&A) documents.

Ontologies have been applied in the security management domain to overcome the complexity of modern information systems [56], [57]. A number of security-related knowledge sources within organizations are kept in the security ontology which is the centric of the security framework. The specific Risk Assessment security ontology [58] was built to manage the security requirement and the threat countermeasure assessment. The proposed ontology describes in detail the variety of threats and the association between the threats and the countermeasures although the threat asset measurement is quite simple. [59] Presents an approach to corporate assets in a company when taking into account the entire infrastructure. The approach proposes a quick calculation of effective countermeasures using the security ontologies. The company infrastructures such as computer, network, server, person, etc are taken into account in the measurement evaluation. However, the model uses some heuristic estimation parameters thus it could cause imprecise risk evaluation results. Our paper will go a step further by suggesting more a precise risk assessment model which also takes into account the semantic relation between the business objects into account.

Ontologies are also widely used in other specific security sub-domains such as network security [60], data privacy [61], access control [62], [63], pervasive computing [64]. However, the existing researches do not consider the affection of threats in the entire enterprise which is one of our paper’s aims.

## 6.3 Terms and Definitions

Below are some important definitions for understanding concepts [68].

### 6.3.1 Security Flaw

A ***security flaw*** is a defect in a software application or component that, when combined with the necessary conditions, can lead to a software vulnerability.

### 6.3.2 Vulnerability

***Vulnerability*** is a set of conditions that allows violation of an explicit or implicit security policy.

### 6.3.3 Exploit

An ***exploit*** is a piece of software or a technique that takes advantage of a security vulnerability to violate an explicit or implicit security policy.

Before going into details first we have to understand that what is risk and what are other factors involved with it.

### 6.3.4 Risk

A report that shows assets, vulnerabilities, likelihood of damage, estimates of the costs of recovery, summaries of possible defensive measures and their costs and estimated probable savings from better protection.

### 6.3.5 Risk Analysis

A *"risk analysis"* is the process of arriving at a risk assessment, which is also called a "threat and risk assessment."

### 6.3.6 Threat

A "threat" is a harmful act such as the deployment of a virus or illegal network penetration. A "risk" is the expectation that a threat may succeed and the potential damage that can occur.

### 6.3.7 Denial-of-Service Attack (DoS)

The state in which a malicious attempt is made by exploiting the weakness or design limitation in an information system to block user, process or system from accessing a network service is called Denial of Service Attack (DoS). Denial of service attack floods the network with additional requests/traffic and the network services get busy with accomplishing these requests and the regular request/traffic can not be accomplished or either slowed or completely interrupted. Examples of DoS attacks include flooding network connections, filling disk storage, disabling ports, or removing power.

## 6.4 What is Risk Assessment?

A risk assessment is the method of identifying risk associated to organization business entities. In general, the are two types of factors which can lead the organization at risk, (a) ***external factors*** which means that the factors outside of an organization can cause damage e.g. economy, competitive landscape and legislative changes, (b) the ***internal factors*** are those which are within organization and organization has direct control. Internal factors include availability of competent personnel, improper vision etc. There is no such method by which one can completely get rid of threats which an organization might have, but by calculating risks assessment big lose can be minimized [91].

### 6.4.1 Components of Risk Assessment

In the following two paragraphs we will investigate what are the different factors which cause risks and how to overcome these factors, using technical report [91].

During past few years numerous reports published narrates protecting electronic data against attacks and different kind of risks associated with that. Such kind of problems arises because of weak security mechanism. Sometimes poor security mechanism creates problems and sometimes lack of expertise in the specific domain to completely overcome the problem makes it difficult to overcome the problem. The major challenge in most of organizations is to identify the assets, and also what are possible attacks which can be harmful for assets. By overcoming these problems they can establish a proper security mechanism against attacks.

The process of risk assessment facilitates the organizations to setup policies and devises techniques to reduce the risk which they might have. Moreover risks and threats are dynamic in nature; therefore it is important for an organization that they repeat the cycle of risk assessment over period of time so that new policies and techniques could be adopted to cope with the changes. This continuing cycle of activity, including risk assessment, is illustrated in the Figure 6.1 depiction of the risk management cycle.

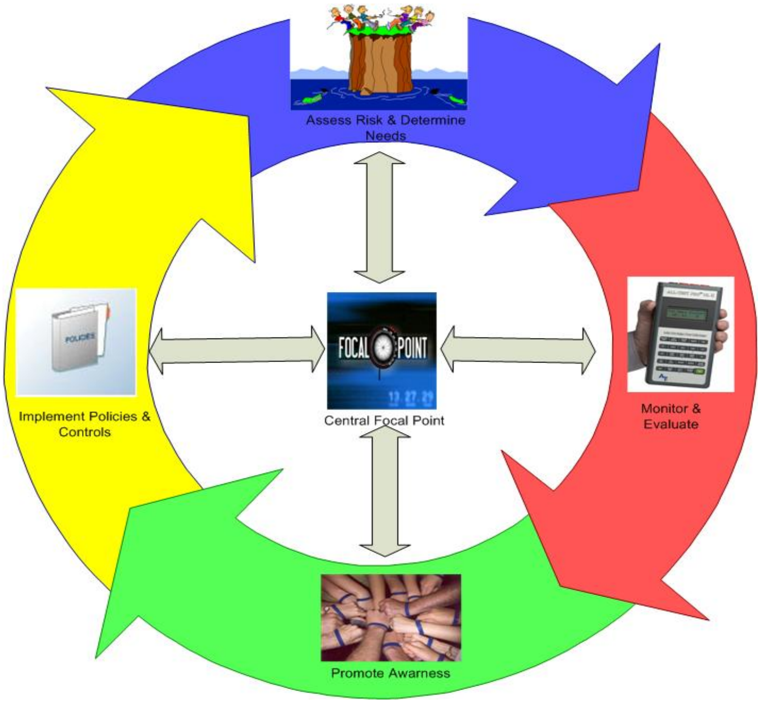


**Fig 6.1: Basic Elements of the Risk Assessment Process**[28]

The process of risk assessment in organization life cycle is an important factor since it gives them an idea for understanding which factors are harmful for the organization and what are the factors which can lead them to threats which causes risks to their infrastructure so as to make necessary arrangements to reduce the level of risk. The use of computer technology is increasing everyday and people (either

[28] www.**gao**.gov/special.pubs/ai00033.pdf

working at home or at workplace) almost store every kind of information in computer systems, so there is a need to protect electronic data since it can cause high losses. Regardless of the types of risk being considered, all risk assessments generally include the following elements [91].

1. Identifying threats that could harm and, thus, adversely affect critical operations and assets. Threats include such things as intruders, criminals, disgruntled employees, terrorists, and natural disasters.
2. Estimating the likelihood that such threats will materialize based on historical information and judgment of knowledgeable individuals.
3. Identifying and ranking the value, sensitivity, and criticality of the operations and assets that could be affected should a threat materialize in order to determine which operations and assets are the most important.
4. Estimating, for the most critical and sensitive assets and operations, the potential losses or damage that could occur if a threat materializes, including recovery costs.
5. Identifying cost-effective actions to mitigate or reduce the risk. These actions can include implementing new organizational policies and procedures as well as technical or physical controls.
6. Documenting the results and developing an action plan.

There are number of reasons which have been explained earlier. But the risk assessment process is composed of different steps. The important thing to be remembered is that guaranteeing 100% is never possible; however by taking necessary measures one can reduce the chances of big losses.

## 6.5 Exploitation of Ontologies

As explained earlier, our approach to calculate risk assessment is based on ontologies. To furnish this task we have divided the ontologies into three parts, i.e. (a) user environment ontology, (b) project ontology and (c) attack ontology.

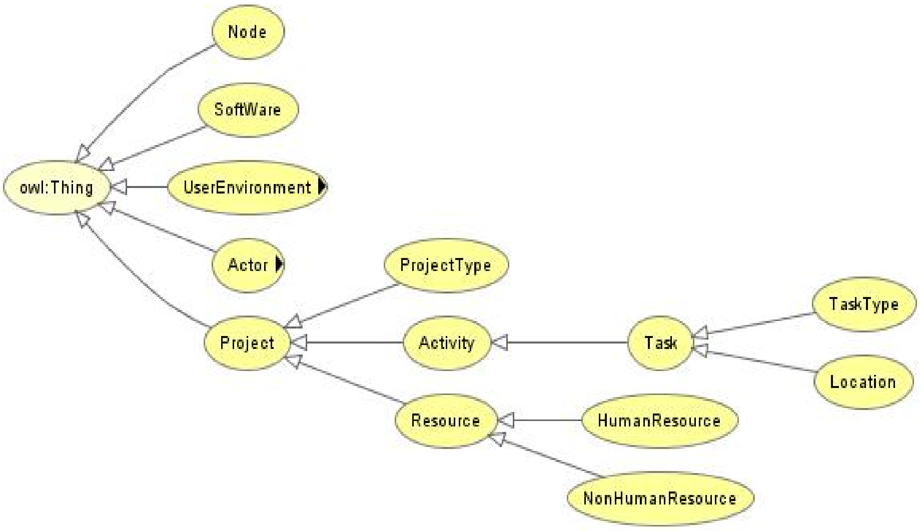


**Fig 6.2: Project Ontology**

The user environment ontology captures the concepts of an environment in which users work. By environment we mean the kind of operating system that is installed on the node, the kind of software that is in use and configuration of hardware at node, etc. A more detailed view of this ontology has been depicted in the Figure 6.3.

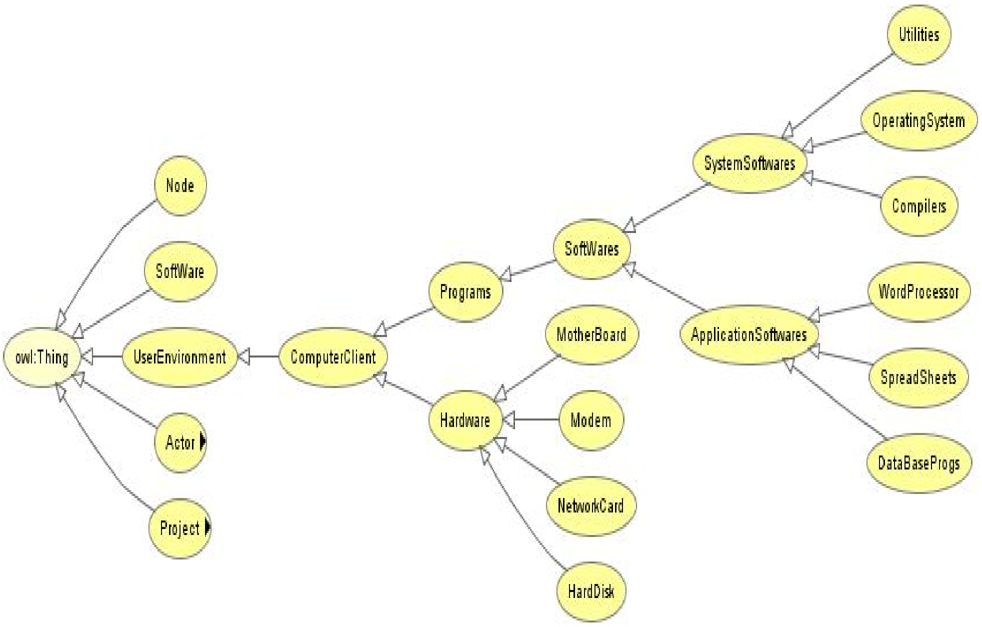


**Fig 6.3: User Environment Ontology**

Project ontology (Figure 6.2) describes the taxonomy of project-related entities such as tasks, project plans, assignments allocations, resources, & costs.

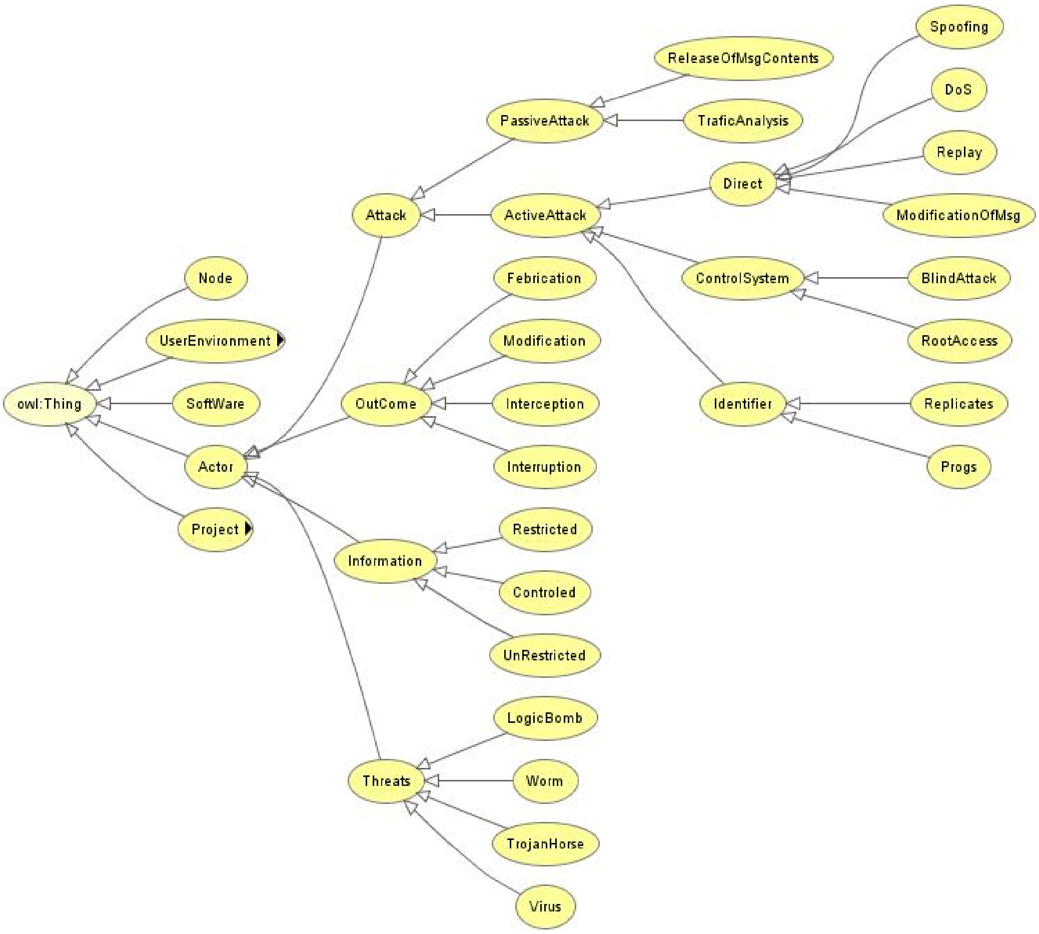


**Fig 6.4: Attack Ontology**

The attack ontology shown in Figure 6.4 is the main focus, describing the different kind of possible attacks, like active and passive attacks, which are the different preconditions for attacks, which are the outcome of an attack etc.

All these ontologies are mapped onto organizations' high level ontology, so that they can be used as the common means of information sharing. On the other hand it provides a solid base that can be used by organizations to translate the processes in a way that computers can interpret and apply them as business rules. As a matter of fact, business entities will be processed by machine after being enriched with organization ontology.

## 6.6 Proposed Solution

In this section we will explore the proposed solution to collaborative risk assessment in an organization. First of all the core components of SemanticLIFE framework that play an important role in the proposed solution will be introduced.

SemanticLIFE is built on several plug-ins components which communicate via the messaging and collaboration component. Message Bus, Web Service and Pipeline plug-ins are the fundamental plug-ins support of the communication framework.

### 6.6.1 Plug-in Framework

External Web Services that can be plugged to SemanticLIFE system at any time by locating the corresponding service description (WSDL file's URL). More important, the plug-in supports capturing the semantic of a service which can be later on used during the contract making mechanism of business processes. The semantic of a service is defined in OWL-S standard and describes the functions of the service in terms of the transformation affected by the accordant service. It also specifies required the inputs, a pre-condition to invoke a service, the generated outputs, and the expected effects that result from the execution of the service (detailed information about Plug-in framework can be found in section 4.3.2.3).

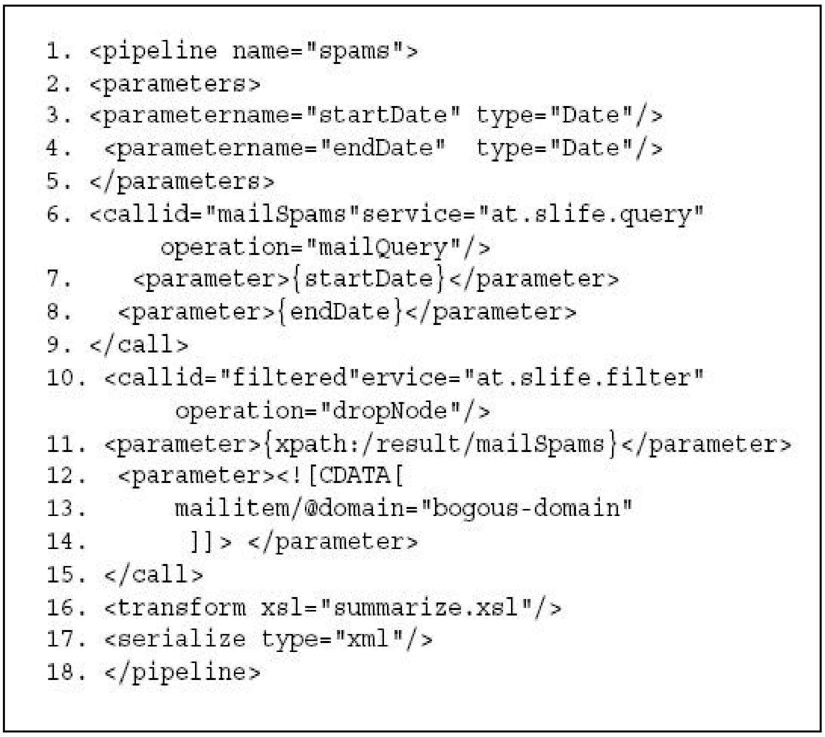
Internal SemanticLIFE services that are primary built-in services of SemanticLIFE are used by other framework components. Examples of such services are semantic query, the annotation and the storage services. Internal services can be also extended by advanced users to customize the environment for their special needs.

A SemanticLIFE pipeline is a mechanism to compose more complex services from primary services. The pipelines are also used for filtering the results based on user privacy and security policies.

### 6.6.2 Service Orchestration

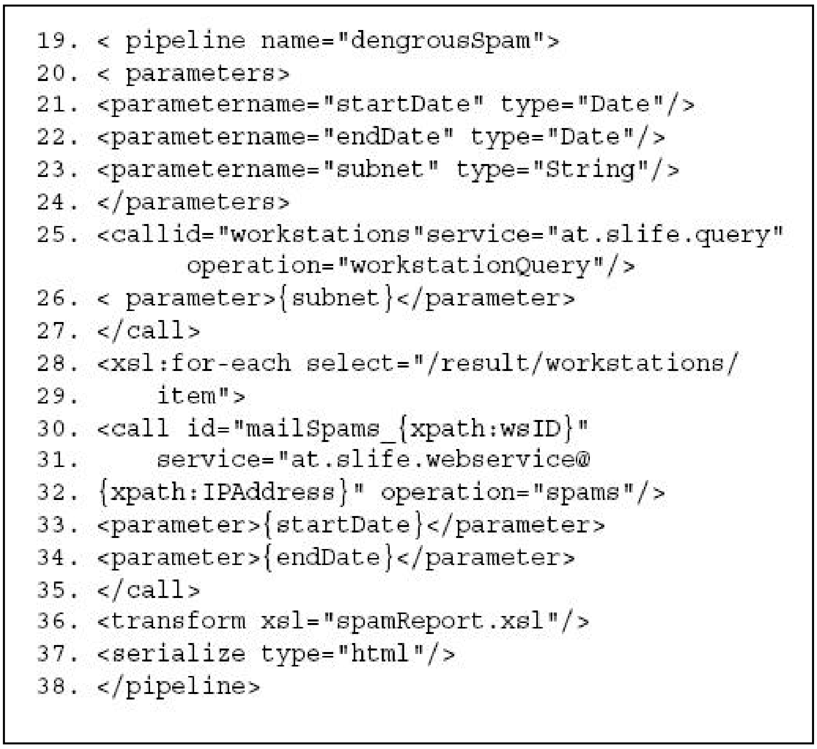
The pipeline plug-in plays a central role in the proposed solution and aims at orchestrate basic system services and creating new business services. We introduce the notion of a pipeline as an uniquely named set of service-calls and intermediate transformations. The pipelines are defined using an XML structure that specifies pipeline steps and relevant transformations. Listing 6.5 shows a simple pipeline that uses the internal SemanticLIFE services to get the Spam Emails from a workstation running SemanticLIFE (detailed information available at 4.3.2.3).

The authorization process starts when Person-P1 requests Person-P2 to show a list of authorized documents. Person-P2 has full access to accept or deny request, however persons can add or delete some documents related to the project for an authorized list depending on the confidentiality status. An important feature of the pipelines is that they can be shared with other users based on user/administrator defined security policies. For example the above defined pipeline can be installed in each workstation (inside the SemanticLIFE ecosystem) and report the spam from a specific domain to system administrator.



```
<pipeline name="spams">
<parameters>
<parametername="startDate" type="Date"/>
 <parametername="endDate"  type="Date"/>
</parameters>
<callid="mailSpams"service="at.slife.query"
     operation="mailQuery"/>
   <parameter>{startDate}</parameter>
  <parameter>{endDate}</parameter>
</call>
 <callid="filtered"ervice="at.slife.filter"
      operation="dropNode"/>
 <parameter>{xpath:/result/mailSpams}</parameter>
  <parameter><![CDATA[
      mailitem/@domain="bogous-domain"
       ]]> </parameter>
 </call>
 <transform xsl="summarize.xsl"/>
 <serialize type="xml"/>
 </pipeline>
```

**List 6.5: Simple Pipeline**



```
< pipeline name="dengrousSpam">
< parameters>
<parametername="startDate" type="Date"/>
<parametername="endDate" type="Date"/>
<parametername="subnet" type="String"/>
</parameters>
<callid="workstations"service="at.slife.query"
       operation="workstationQuery"/>
< parameter>{subnet}</parameter>
</call>
<xsl:for-each select="/result/workstations/
    item">
<call id="mailSpams_{xpath:wsID}"
    service="at.slife.webservice@
{xpath:IPAddress}" operation="spams"/>
<parameter>{startDate}</parameter>
<parameter>{endDate}</parameter>
</call>
<transform xsl="spamReport.xsl"/>
<serialize type="html"/>
</pipeline>
```

**List 6.6: Administrator's Pipeline to Access Workstation Services**

At the administrators' side there will be a similar situation & a pipeline will make a call to each workstation & will combine the results & will display the summary to administrator. The Listing in Figure 6.6 shows the administrator's pipeline for this scenario. As shown in the listing above, the pipeline concept offers many flexible features and complex scenarios can be described in terms of pipelines.

### 6.6.3 Semantic Filtering

The SemanticLIFE services open the workstation services to the outside world. So there is a need for taking care of the users' privacy and security issues. Figure 6.7 shows the security and privacy scenario in SemanticLIFE. The relevant information should be provided to authorized people only.

One way of defining such authorizations is to use ontologies and find out the relationship between items and users. For example project ontology can tell us that a person is a project member, so he/she should be able to access all items (mails, files, photos, etc) on local computer that are tagged to be shared with project colleagues.

The other usage of such ontological authorization rules is to filter the outgoing data. Some typical filters to services are.

- ***Content filter*** (filters all items containing a specified term, statement)
- ***Semantic filter*** (filters all project related documents)
- ***Annotation filter*** (filters all photos annotated by specific terms like Class Diagram, ERD)

Each filter performs a specific task; e.g. the semantic filter will filter the documents which are related to a project. The annotation filter will filter the information items; e.g. filter the email with annotation project (X) or pictures with annotation ERD. For developing filters we need to specify which kind of information objects need to be filtered and then the inference engine will verify which information objects can pass through and be available to the requested person.

Ontology and inference engine are the basis for semantic filtering. The ontology is represented in formal language like OWL which captures the key concepts and relationships in the domain of interest, for example, in the project some key concepts are emails, documents and tools.

### 6.6.4 Policy Implementation

In an organization, people have different types of access to the resources depending on their job description. In a collaborative environment where people work together access to resources should be allowed based on defined policy and privileges. In collaborative environment where people work in groups, to accomplish their tasks it is essential that they have privileges over certain resources.

In SemanticLIFE information in the semantic store will be handled through policies. Policies are stored in the triple store in the form of RDF, which will facilitate how data should be handled i.e. who has access to which information and for whom data is restricted. Also information about how data was handled previously will be stored

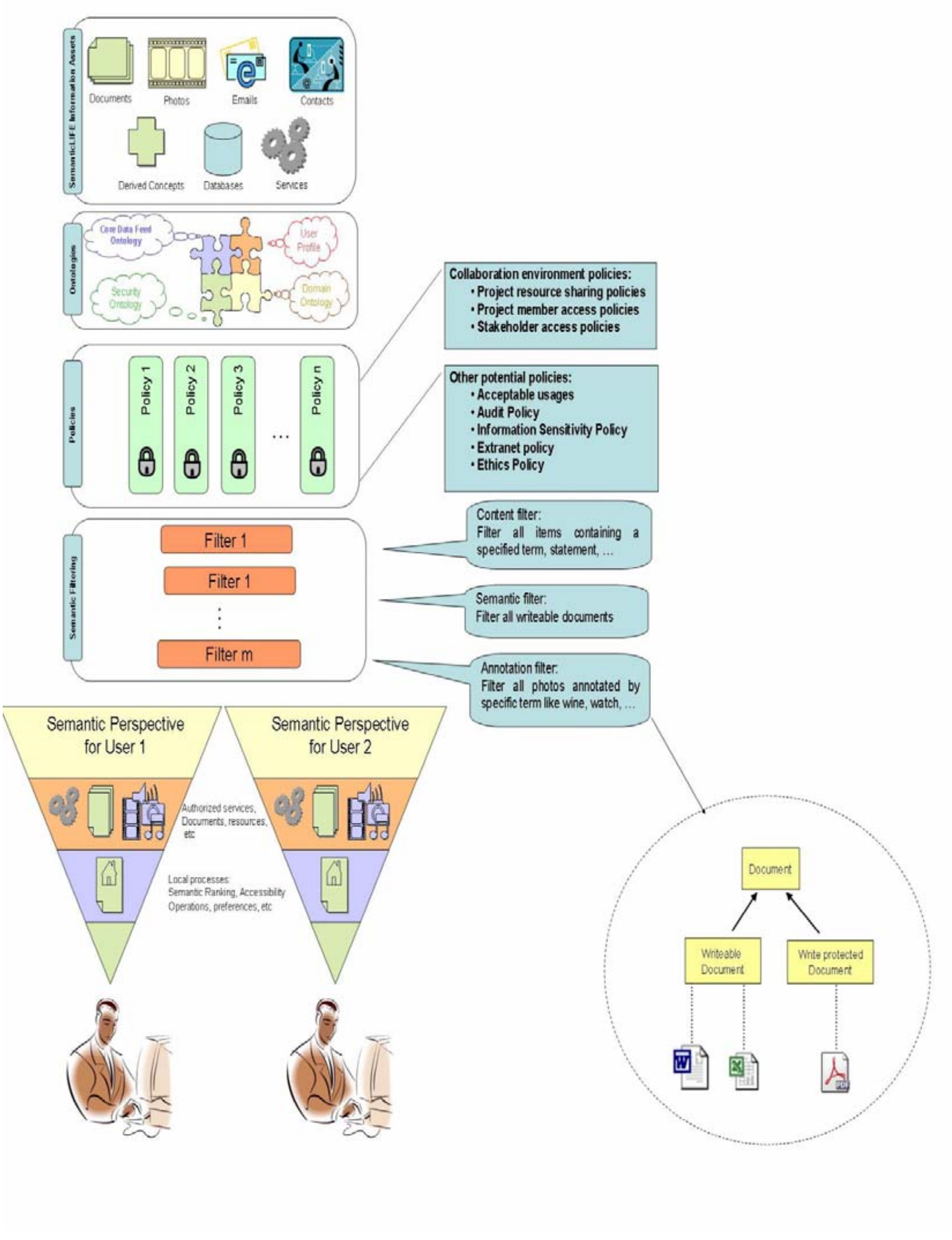

SemanticLIFE Information Assets
Documents
Photos
Emails
Contacts
Derived Concepts
Databases
Services
Ontologies
Core Data Feed Ontology
User Profile
Security Ontology
Domain Ontology
Policies
Policy 1
Policy 2
Policy 3
...
Policy n
Collaboration environment policies:
• Project resource sharing policies
• Project member access policies
• Stakeholder access policies
Other potential policies:
• Acceptable usages
• Audit Policy
• Information Sensitivity Policy
• Extranet policy
• Ethics Policy
Semantic Filtering
Filter 1
Filter 1
Filter m
Content filter:
Filter all items containing a specified term, statement, ...
Semantic filter:
Filter all writeable documents
Annotation filter:
Filter all photos annotated by specific term like wine, watch, ...
Semantic Perspective for User 1
Semantic Perspective for User 2
Authorized services, Documents, resources, etc
Local processes: Semantic Ranking, Accessibility Operations, preferences, etc
Document
Writeable Document
Write protected Document

**Fig 6.7: Security and Privacy scenario in SemanticLIFE**

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.

Numbers of policies can be implemented according to the collaborative environment. The user defines the policy for some specific operation, e.g. project resource sharing policies, project member access policies, stakeholder access policies etc. Take an example where the Person-P1 asks for documents related to the project. Person-P1 can be granted access if he/she is a member of that project and his/her status matches to the confidentiality of that document. There are several candidate policy implementation languages like SWRL [65], KAoS [66] and REI [67]. We are still in the exploring phase for the best suitable policy language which meets the requirements of our project.  Figure 8 shows different components of SemanticLIFE and how policies and filters are used to control the information flow between SemanticLIFE-enabled workstations.

## 6.7 Ontology-based Risk Assessment

In this section we will explore a risk assessment example based on SemanticLIFE toll for organizations' security which is a critical issue for planners and decision makers. As explained before SemanticLIFE gathers the user interaction events and correlates those using ontologies. In the following sections, it will be explained how SemanticLIFE components can be employed to deliver the required input for organizational risk assessment methods.

### 6.7.1 Data Capture and Event Correlation

SemanticLIFE provides an effective approach to capture user-computer interaction. Such information can be analyzed and correlated with other events to establish a security profile for users & their PC.

For instance, the following items can be identified from semantic repository:

- The applications that are installed on the system and their version.
- The processes that are running on the system.
- The Web pages that have been browsed by the user can be monitored and tracked.
- Emails and spam.

Combining this data with risk ontology, useful results can be generated. For example, from the risk ontology a specific attack may happen only when specific preconditions are met. Some typical preconditions are OS version, open ports, known spam, worms, etc. In other words the risk ontology will be weighted by the analysis results of user interaction records. As an example the reception of a known spam will give a higher weight to the corresponding ontology elements.

### 6.7.2 Collaboration at Organizational Level

Maintaining and monitoring the computer networks and nodes is a big challenge for system administrators and organizations. Supervising all computers and detecting the attacks in big organizations is nearly impossible.

SemanticLIFE architecture provides a mechanism to expose the risk factors to the system administrator who is responsible for securing the systems. This can be realized via some pipelines on user workstations that assess required information and share them in a “Service Oriented” like paradigm. Such a scenario was demonstrated in previous sections.

As proof of concept we show how the results of SemanticLIFE paradigm can be used to feed a typical quantitative risk assessment method that is used to assess the risk factor in organization. Quantitative methods generally use the available data to give a numerical description of the risk. Figure 6.8 shows one such quantitative method for calculating the Annual Loss Expectancy (ALE) that has been introduced in Common Framework.

According to definition the Annual Loss Expectancy is the estimation of the yearly potential loss of an organization if the risks are not handled. The ALE is calculated as follows:

$$ALE = \sum_{i=0}^{n} I(O_i)F_i$$

**Fig 6.8: ALE Formula**

Where ***O* = *{O1,O2,...On}*** is a set of harmful outcomes, ***I*(*Oi*)** is the impact of outcome ***Oi*** in US dollars, and ***Fi*** is the frequency of ***i*th** outcome. In the above equation the set of harmful outcomes and also the impact of each outcome can be estimated, however the frequency of the outcome can not be easily known and this makes the ALE coarse and not easy to calculate. However using the SemanticLIFE paradigm we can feed the ALE calculation process by the input from real events and as a result the ALE will be more realistic.

SemanticLIFE paradigm results can be also be used for assessing the project infrastructure risk. As explained before a project is distributed among different nodes in the organization and these nodes are configured and managed differently. In this case the project and environment ontologies will bridge the gap between project and infrastructure nodes. So the over all project infrastructure risk is the aggregation of single node risks that are involved in project development process. The following steps describe the steps needed to fulfill this task:

- The tasks and responsible persons are extracted from the project ontology
- Combining the results of previous step with the user environment ontology, the relevant network nodes can be selected
- From the risk ontology it is known which risks are conceivable for each node and the set of harmful outcomes can be assessed
- By setting up the appropriate pipelines, the organization wide data can be gathered & fed into risk assessment algorithms (such as ALE)

Similar approach can be followed to assess the risk for computer/user groups and answer questions like:

- Which department has the highest risk?
- Which project is at the highest risk?

## 6.8 Privacy Issues

Though the proposed method seems to be very effective to control and maintain the organization network and nodes, but it is important to take the privacy issues into consideration. Since in the risk assessment methods, the personal information is not required, it makes sense to depersonalize information before sharing them. At this point the semantics of user interaction records can be used again to locate the sensitive data and filter them by applying appropriate policies and filters.

# Chapter 7

# User Data Privacy in Web Services

## 7.1 Introduction

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” [69].

As a matter of fact, Web Services and Semantic Web are two building blocks to provide machine processable services. One of the biggest challenges in both Web Services and Semantic Web concerns privacy issues. Privacy means which part of information should be hidden and which should be visible. To acquire any service, one has to disclose personal information (e.g. home address, date of birth, mobile number, credit card information etc) so as to fulfill the requirements and utilize the service properly. Moreover, there should be control over who can access to information and for whom its restricted. Especially when there is requirement to share information with third parties it is important to limit the access to information. Also in collaborative environment, where information is being shared among different entities, there should be control over information being shared.

## 7.2 Background

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 extremely powerful, but just a small percentage 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.

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 [70]. 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) [71].

In [72] 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 [73] 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 a 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 [74]. 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. Baresi et al. [75] 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.

## 7.3 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 to wrap the services in semantic containers to make the services and their results machine processable. 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 everything valid for Web Services is 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.

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.

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 self while disclosing information like name, job title or department etc. The functionality of different components in the SemanticLIFE architecture is explained as below.

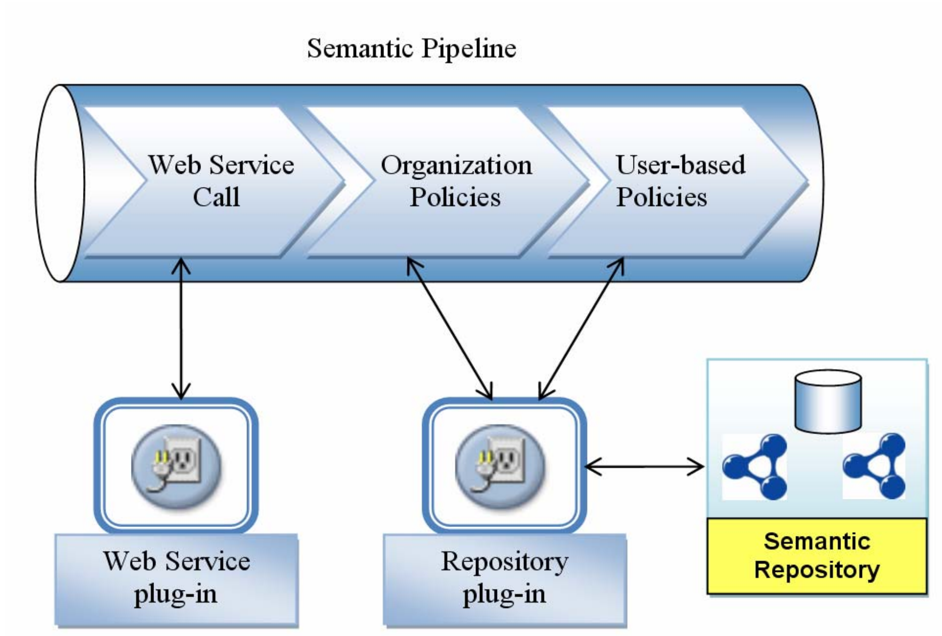


**Fig 7.1: Web Service Call Using a Semantic Pipeline**

Organization and user-based policies are subject to various rules and constraints. Such rules can be used for the Semantic Web Inference technologies like RuleML [76] and SWRL [77]. In the SemanticLIFE case we have used Jena 2 Inference [78] 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 7.1. 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 possesses 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.

## 7.4 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.

The user ontology shown in Figure 7.2 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.

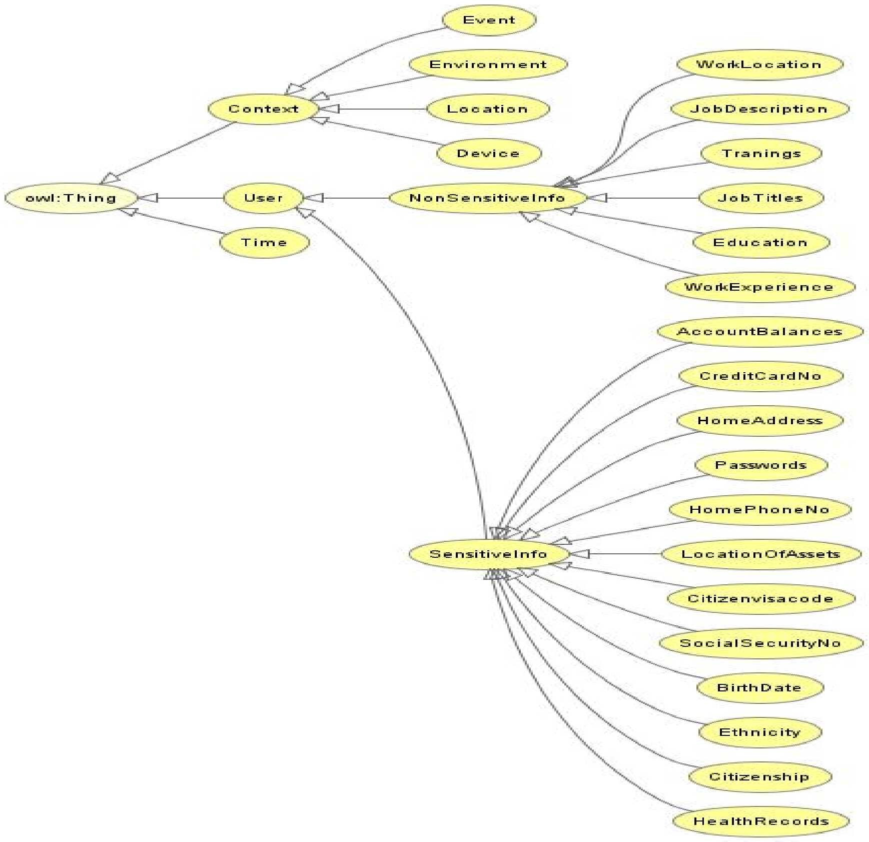


**Fig 7.2: User Ontology**

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 interval, 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 7.3 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.

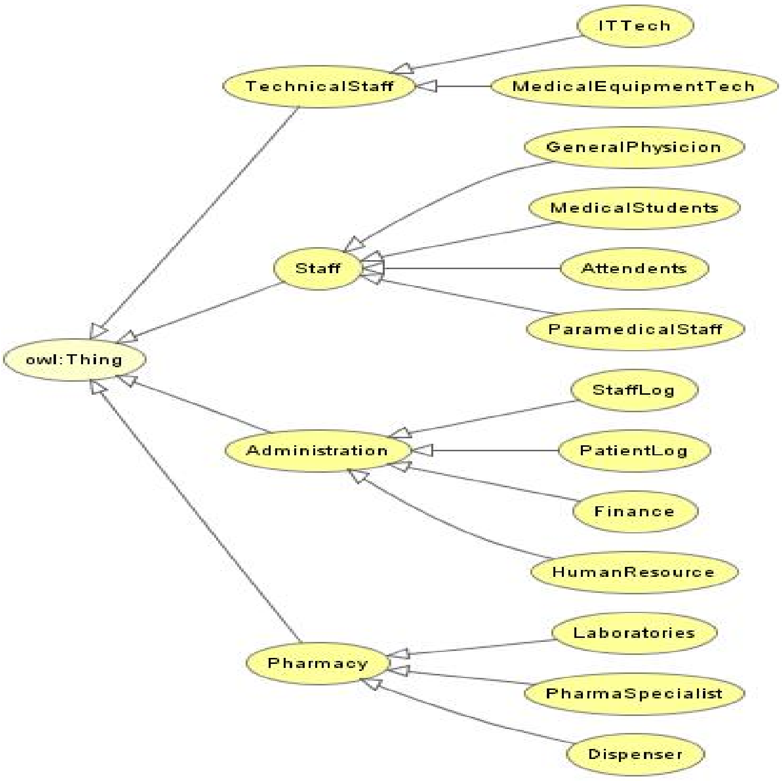


**Fig 7.3: Hospital Ontology**

Web Services and Semantic Web are two building blocks to provide machine processable 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 [79]. To define the privacy policies and rules, we have used the privacy ontology from DAML-Services as follows.

The privacy ontology shown in Figure 7.4 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.

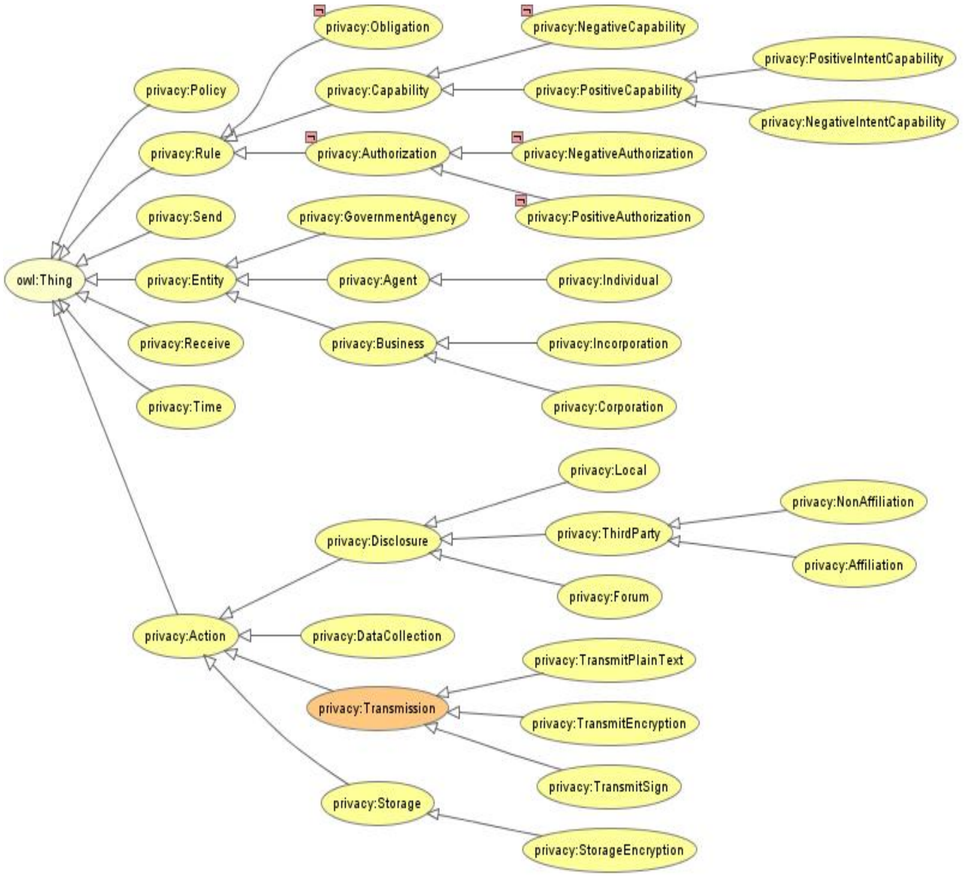


**Fig 7.4: Privacy Ontology**

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 [79].

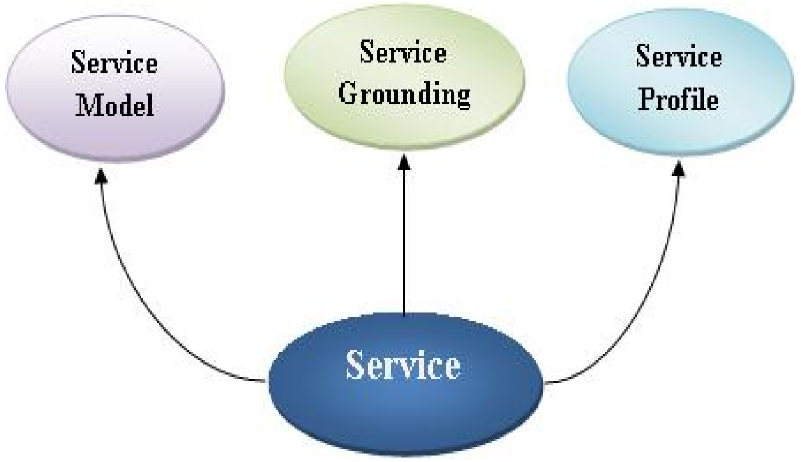


Fig 7.5: Web Service Ontology (OWL-S)

OWL-S is a Semantic Web Services description language which enriches Web Services description with semantics. The top level service ontology is depicted in Figure 7.5. 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"* [80] [81].

## 7.5 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" [82].

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 [83]. 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 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 7.1).
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.

# Chapter 8

## Conclusion & 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 which are proof and trust layers are not fully implemented yet.

In this thesis we have explored that how Semantic Web technologies can be used to deal with security and privacy aspects. Traditional access control models are no more useful (where parties know each other in advance) to be used in Semantic Web environment where people will be interacting with each other anonymously. Therefore, semantically enriched process is needed to deal with automatic access to sensitive information.

SemanticLIFE is a Personal Information Management System which gathers the user interaction events and correlates those by using ontologies. The proposed scenario suggests that technology can be used to make the daily life scenarios easier to organize. The presented SemanticLIFE platform has the capacity to be used in other business processes dealing with personal information (local data, resources, etc). The SemanticLIFE platform also proposes a paradigm to manage the security and privacy issues of information and process sharing. After a secure and robust share of such information, it is possible to assess organizational-level factors such as risk factors which is fundamental issue for planners and decision makers in the IT field. Moreover, the proposed scenario shows the SemanticLIFE's approach to address user data privacy and security issues in service-oriented environments and explored achieving the goal using SOPA (service-oriented pipeline architecture) framework. Some other challenging module like Semantic Web Services is still under development progress and we try to enhance the features and keep up with the latest advances. The SemanticLIFE domain ontology is also evolving and it aims at being empowered by the known risk, user profile and infrastructure ontologies. 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 and we are trying to apply it to other businesses and exploit the strength of Semantic Web Services as a business-enabler.

# BIBLIOGRAPHY

[1] V. Bush (1945),"As We May Think", The Atlantic Monthly, reproduced at http://www.ps.uni-sb.de/~duchier/pub/vbush/vbush-all.shtml .

[2] T. Berners-Lee (1989). Information Management: A Proposal. CERN. available at: http://www.w3.org/History/1989/proposal.html .

[3] T. Berners-Lee, J. Hendler, and O. Lassila (2001). "The Semantic Web Scientific American", available at: http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

[4] World Wide Web, available at: http://en.wikipedia.org/wiki/World_wide_web.

[5] B. Thuraisingham, "Security Issues for the Semantic Web," in Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC03). IEEE Computer Society, November 2003, pp. 633–638.

[6] B. Thuraisingham , "Xml Databases and the Semantic Web". CRC Press, FL, 2001.

[7] Bertino, E., et al, "Access Control for XML Documents", Data and Knowledge Engineering, Volume 43, #3, 2002.

[8] Bertino, E. et al, "Secure Third Party Publication of XML Documents", in IEEE Transactions on Knowledge and Data Engineering, 2004.

[9] Carminati, B., et al, "Security for RDF", Proceedings of the DEXA Conference Workshop on Web Semantics, Zaragoza, Spain, 2004.

[10] Farkas, C. And A. Stoica, "Correlated Data Inference", Proceedings data and Applications Security Conference, 2003.

[11] Thuraisingham, B., "Security Standards for the Semantic Web", Computer Standards and Interface Journal, 2005.

[12] Thuraisingham, B., "Administering the Semantic Web, Confidentiality, Privacy and Trust", Journal of Information Security and Privacy, 2006.

[13] L.F. Cranor, "Web Privacy with P3P", New York, 1967

[14]. Dictionary of Computing, Fourth Ed. (Oxford: Oxford University Press, 1996).

[15] Privacy.net site, http://www.privacy.net.

[16] Platform for Privacy Preferences (P3P) Project site, http://www.w3.org/P3P/.

[17] Center for Democracy and Technology (CDT) Briefing Book on Privacy Legislation – 2000.

[18] Hoffman L. J, "Building in Big Brother" Springer-Verlag, New York Berlin Heidelberg (1995).

[19] Landler, M., "Fine-Tuning for Privacy, Hong Kong Plans Digital ID", The New York Times. http://www.nytimes.com/2002/02/18/technology/18KONG.html. (Feb. 18,2002).

[20] A. Kim, L. J. Hoffman, and C. D. Martin, "Building Privacy into the Semantic Web: An ontology needed now," in Proceedings of International Workshop on the SemanticWeb,Hawaii,USA,May2002,semanticweb2002.aifb.uni-karlsruhe.de/proceedings/Position/kim2.pdf.

[21] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, "The P3Pspecification," W3C, Recommendation 1.0, 2001, http://www.w3.org/TR/2001/WD-P3P-20010928/.

[22] History of ontology, available at: "http://ontology.buffalo.edu/.

[23] Matt. B, Joan. F, and Martin. S "Compliance Checking in the Policymaker Trust Management System", In Financial Cryptography, Second International Conference, volume 1465 of Lecture Notes in Computer Science, pages 254–274, Anguilla, British West Indies, February 1998. Springer.

[24] Andrzej. U, Jeffrey M. Bradshaw, Renia. J, Niranjan. S, Patrick J. Hayes, Maggie R. Breedy, Larry. B, Matt. J, Shriniwas. K, and James. L, "KAoS Policy and Domain Services: Toward a Description-logic Approach to Policy Representation, Deconfliction, and Enforcement". In POLICY, page 93, 2003.

[25] Lalana Kagal, Timothy W. Finin, and Anupam Joshi, "A Policy based Approach to Security for the Semantic Web". In The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings, Lecture Notes in Computer Science, pages 402–418.Springer, 2003.

[26] Rita Gavriloaie, Wolfgang Nejdl, Daniel Olmedilla, Kent E. Seamons, and Marianne Winslett, "No registration needed: How to use declarative policies and negotiation to access sensitive resources on the Semantic Web". In 1st European Semantic Web Symposium (ESWS 2004), volume 3053 of Lecture Notes in Computer Science, pages 342–356, Heraklion, Crete, Greece, May 2004. Springer.

[27] Moritz Y. Becker and Peter Sewell. Cassandra: “Distributed Access Control policies with tunable expressiveness”. In 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004), 7-9 June 2004, Yorktown Heights, NY, USA, pages 159–168. IEEE Computer Society, 2004.

[28] Piero A. Bonatti and Daniel Olmedilla, “Driving and monitoring provisional trust negotiation with metapolicies”. In 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005), pages 14–23, Stockholm, Sweden, June 2005. IEEE Computer Society.

[29] Berners-Lee, T. (1999). “Weaving the Web”, London: Orion Business Books

[30] Dieter. F: “Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce”, Feb 2000.

[31] Smith Barry. “Ontology: Philosophical and Computational” http://ontology.buffalo.edu/smith/articles/ontologies.htm;

[32] Quine WVO, “On What There Is. Review of Metaphysics”, 1948; p. 21–38.

[33] Gruber Thomas R, “A Translation Approach to Portable Ontology Specifications” Knowl Acquis 1993; 5(2):199–220.

[34] Ogden, C. K. and I. A. Richards, I. A. (1923), “The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism”, London:Routledge&KeganPaul.

[35] Peirce, C. S. (1931-1958). Collected Papers of C. S. Peirce ed. by C. Hartshorne, P.Weiss,&A.Burks,8vols.,HarvardUniversityPress,Cambridge,MA.

[36] Sowa, J. F. (2000), “Ontology, Metadata, and Semiotics”, Presented at ICCS'2000 in Darmstadt, Germany, on August 14, 2000. Published in B. Ganter & G. W. Mineau, eds., Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in AI #1867,Springer-Verlag,Berlin,2000,pp.55-81.Available at: http://users.bestweb.net/~sowa/peirce/ontometa.htm.

[37] Kopena Joseph, Regli William, “DAMLJessKB: A Tool for Reasoning with the Semantic Web”, IEEE Intelligent Systems 2003; 18(3):74–77.

[38] Carroll JeremyJ, Dickinson Ian, Dollin Chris, Reynolds Dave, Seaborne Andy, Wilkinson Kevin, “Jena: implementing the Semantic Web recommendations. In: WWW (Alternate Track Papers & Posters)”, 2004. p. 74–83.

[39] Tsarkov Dmitry, Horrocks Ian, “Implementing new reasoner with datatypes support. WonderWeb:Ontology Infrastructure for the Semantic Web Deliverable”,

2003.

[40] Horrocks Ian, “The FaCT System. In: Automated Reasoning with Analytic Tableaux and Related Methods”, International Conference Tableaux-98. Springer Verlag; 1998. p. 307–312.

[41] Haarslev Volker, Moller Ralf, “Description of the RACER System and its Applications”, In: Proceedings of the International Workshop in Description Logics 2001 (DL2001); 2001.

[42] Sirin Evren, Parsia Bijan, “Pellet: An OWL DL Reasoner. In: Description Logics”, 2004.

[43] Kalyanpur Aditya, Parsia Bijan, Hendler James, “A Tool for Working with Web Ontologies”, In International Journal on Semantic Web and Information Systems. vol. 1, 2005.

[44] Jeen Broekstra, Michel Klein, Stefan Decker, Dieter Fensel, Ian Horrocks, “Adding formal semantics to the Web”, September 4, 2000. http://www.ontoknowledge.org/oil/papers/extending-rdfs.html.

[45] Grigoris. A, Frank. H, “A Semantic Web Premier”, The MIT press, London, England, 2004.

[46] Wikipedia,http://en.wikipedia.org/wiki/Tim_Berners-Lee.

[47] Ontology, available at: http://en.wikipedia.org/wiki/Ontologies.

[48] Undercoffer, J., Joshi, A., Finin, T. and Pinkston, J, “A Target-Centric Ontology for Intrusion Detection”, In 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2004.

[49] Semantic Web, available at: http://en.wikipedia.org/wiki/Semantic_Web.

[50] Anya kim, Jim Luo, Myong H. Kang, “Security Ontology for Annotating Resources”. OTM Conference 2005:1483-1499.

[51] Luciana A. F. Martimiano and Edson Moreira, “The Evaluation Process of a Computer Security Incident Ontology” in 2nd Workshop on Ontologies and their Applications, (WONTO2006).

[52] Artem Vorobiev, Jun Han, "Security Attack Ontology for Web Services", Second International Conference on Semantics, Knowledge, and Grid (SKG'06), 2006.

[53] M. Karyda, T. Balopoulos, S. Dritsas, L. Gymnopoulos, S. Kokolakis, C. Lambrinoudakis, S. Gritzalis, " An ontology for secure e-government applications". In Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06) 20-22 April 2006. pp 5 - ISBN: 0-7695-2567-9.

[54] "Information security risk assessment – practices of leading organizations", United States General Accounting Office (GAO), Executive Guide GAO/AIMD-00-33, November1999, http://www.gao.gov/special.pubs/ai00033.pdf.

[55] S.W.Lee, R.Gandhi, D.Muthurajan, D.Yavagal, and G.J.Ahn, "Building problem domain ontology from security requirements in regulatory documents", in Proceedings of international Workshop on Software Engineering for Secure Systems. ACM Press, 2006, pp.43-50.

[56] B.Tsoumas, S.Dritsas, and D.Gritzalis, " An ontology-based approach to information systems security management", in Proceedings of International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS), Lectures Notes in Computer Science, vol-3685. St.Petersburg, Russia: Springer Verlag, September 2005, pp 151-164.

[57] S.Liu and L.F.Kwok, "Data integration framework for knowledge model of organizational information security management", in Proceedings of 2nd Secure Knowledge Management Workshop (SKM), September 2006.

[58] B.Tsoumas and D.Gritxalisi, "Towards an ontology-based security management", in Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AIINA), Vienna, Austria, April 2006, pp. 985-992.

[59] M.Klemen, E.Weippl, A.Ekelhart and S.Fenz, "Security ontology: Simulating threats to corporate assets", in Proceedings of 2nd International Conference on Information System Security (ICISS). Springer 2006, pp. 249-259.

[60] A.Simmonds, P.Sandilands and L.van Ekert, "An ontology for network security attacks", in Proceedings of Asian Applied Computing Conference (AACC), Lecture Notes in Computer Science, vol. 3285. Kathmandu, Nepal: Springer Berlin, October 2004, pp. 317-323.

[61] P.Mitra, C.Pan, P.Liu and V.Atluri, "Privacy-Preserving semantic interoperation and access control for heterogeneous databases", in Proceedings of Symposium on Information, computer and communications security. ACM Press, 2006, pp. 66-77.

[62] H.Li, X.Zhang, H.Wu, and Y.Qu, “Design and application of rule based access control policies”, in Proceedings of Semantic Web and Policy Workshop, Galway, Ireland, November 2005, pp. 34-41.

[63] A. Toninelli, R. Montanari, L.Kagal and O. Lassila, “A semantic context-aware access control framework for secure collaborations in pervasive computing environments”, in Proceedings of 5th International Semantic Web Conference, Lecture Notes in Computer Science, vol. 4273. Athens, GA: ACM Press, November 2006, pp. 473-486.

[64] A. Toninelli, J.M. Bradshaw, L.Kagal and R. Montanari, “Rule-based and ontology-based policies: Toward a hybrid approach to control agents in pervasive environment”, in Proceedings of the Semantic Web and policy workshop, Galway, Ireland, November 2005, pp. 42-54.

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

[66] A. Uszok, J.M. Bradshaw, M. Jhonson and R. Jeffers, “Kaos policy management for semantic web services”, in Proceeding of IEEE 4th International Workshop on Policy. IEEE Computer Society, July/August 2004, pp. 32-41.

[67] Finin, T., Lalana Kagal and A. Joshi, “A policy language for pervasive computing environment”, in Proceeding of IEEE fourth International Workshop on Policy. Italy: IEEE Computer Society, June 2003, pp. 63-76.

[68] Robert C. Seacord, Allen D. Householder, “A Structured Approach to Classifying Security Vulnerabilities”, January 2005, Technical Note CMU/SEI-2005-TN-003.

[69] L. F. Cranor, “Web Privacy with P3P”, New York, 1967.

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

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

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

[73] 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.

[74] 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).

[75] 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.

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

[77] 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/.

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

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

[80] 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/.

[81] 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 for Web Services”, in Proceedings of the International Semantic Web Working Symposium (SWWS), July 2001.

[82] 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”, Pages: 663-671.

[83] 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, Pages: 14- 21.

[84] Jeen Broekstra, Arjohn Kampman, Frank van Harmelen. “Sesame: a Generic Architecture for Storing and Querying RDF and RDF Schema”. 1st International Semantic Web Conference (ISWC2002), June 9-12, 2002. Sardinia, Italy.

[85] B. McBride. "Jena: Implementing the RDF Model and Syntax Specification". In Proceedings of the Second International Workshop on the Semantic Web-SemWeb2001. May 2001.

[86] Jena2 Database Interface - Database Layout: Available at: http://jena.sourceforge.net/DB/layout.html.

[87] Siegfried Handschuh, Alexander Maedche, Ljiljana Stojanovic and Raphael Volz, "KAON-The Karlsruhe Ontology and Semantic Web Infrastructure". White paper available at http://kaon.aifb.uni-karlsruhe.de/white-paper

[88] Aimilia. M, Grigoris. K, Ta Tuan Anh, Vassilis. C, Dimitris. P, "Ontology storage and querying", technical report no. 308, April 2002.

[89] RDFStore API, http://rdfstore.sourceforge.net/documentation/api.html.

[90] Daniel Olmedilla, "Security and privacy on the Semantic Web". In Milan Petkovic and Willem Jonker, editors, Security, Privacy and Trust in Modern Data

Management, Data-Centric Systems and Applications. Springer, 2007.

[91] "Information security and risk assessment, practices of leading organizations", 1999. Available at: www.**gao**.gov/special.pubs/ai00033.pdf.

[92] The Apache Cocoon Project, available at: http://cocoon.apache.org/.

[93] Ahmed et al., "SemanticLIFE' - A Framework for Managing Information of A Human Lifetime", Proc. of the Int. Conf. on Information Integration, Web-applications and Services (IIWAS'04, Indonesia).

[94] Latif, K., Mustofa, K., and Tjoa, A. M. (2006), "An Approach for a Personal Information Management System for Photos of a Lifetime by Exploiting Semantics". In Proceedings of DEXA, volume 4080) of LNCS, pages 467-477, Krakow, Poland. Springer - Berlin / Heidelberg.

[95] XML-based page description language for SWT, http://xswt.sourceforge.net/.

[96] Apache Axis WSDD, http://xml.apache.org/axis/wsdd/.

[97] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana, Web Services Description Language (WSDL), http://www.w3.org/TR/wsdl.

[98] Ian. H, "DAML+OIL: a Description Logic for the Semantic Web", IEEE computer society technical report, 2001.

www.cs.man.ac.uk/~horrocks/Publications/download/2002/ieeede2002.pdf

[99] Abdelmounaam. R, Athman. B, Mohamed. Y, “Privacy on the web: facts, Challanges and Solutions“, IEEE computer Society 2003.

[100] Mind Dictionary, available at:
“http://philosophy.uwaterloo.ca/MindDict/ontology.html”.