

# QUERYING UML USING SESAME

Thesis submitted in partial fulfillment of the requirements for the award of degree of

**Master of Engineering**  
in  
**Computer Science Engineering**



**Thapar University, Patiala**

By:

**Harvinder**

**(80632005)**

Under the supervision of:

**Mrs. Shalini Batra**

**Lecturer (SS)**

**MAY 2008**

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT  
THAPAR UNIVERSITY  
PATIALA – 147004

## CERTIFICATE

---

I hereby certify that the work which is being presented in the thesis entitled, “**Querying UML using Sesame**”, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science & Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs. Shalini Batra*.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

**Harvinder**

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

**Mrs. Shalini Batra**  
Computer Science and Engineering Department  
Thapar University  
Patiala-147004

Countersigned by

(Dr. SEEMA BAWA)  
**Professor & Head**  
Computer Science & Engineering Department  
Thapar University  
Patiala

(Dr. R.K. SHARMA)  
**Dean (Academic Affairs)**  
Thapar University  
Patiala

## ACKNOWLEDGEMENT

---

First and foremost, I would like to express my sincere gratitude to my advisor **Shalini Batra**, Lecturer, Computer Science & Engineering Department for her immense help, guidance, stimulating suggestions and encouragement all the time. She always provide a motivating and enthusiastic atmosphere to work with, it was a great pleasure to do this thesis under her supervision.

I would like to thank **Dr. Seema Bawa**, Head, Computer Science and Engineering Department, Thapar University, Patiala for providing necessary facilities in the Department, to work. The most valuable thing I acquired from the two year studying Thapar University is to protect, survive and endure myself in this world with increased confidence and additional faith in my abilities to achieve. I am also thankful to the authors whose works I have consulted and quoted in this work.

I am very thankful to my respectable family for their fortitude. Nothing would have ever been possible without my parent's unconditional love and encouragement. Last but not the least I would like to thank God for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

**Harvinder**  
**(80632005)**

## Abstract

The vision of next generation web is semantic web where things are deciphered not only by their syntax but also by the meaning associated with them. For this to be successful various applications all over the world should be able to communicate with each other. RDF is one such means of communication, which is a computer-readable language and used to simplify communication between various applications. To ensure such communication a technology is needed, which can easily support the semantic, web concepts and implementations. One such language, which is widely adopted all over the world and in use currently, is UML. Unified Modeling Language (UML) has been widely adopted by the software engineering community and its scope is broadening to include more diverse modeling tasks. It is a framework for describing a set of models that capture the functional and structural semantics of any complex information system. On the other hand, RDF (Resource Description Framework) is a framework for describing resources on the web, which provides a model for data, and syntax so that independent parties can exchange and use it. RDF is a language, which can be used for inference by various applications on the web. This report puts forward using of the UML as a language for modeling resources in semantic web .The goal of this work is to make UML "RDF-compatible". This allows mixing and extending UML models and the language elements of UML itself on the Web in an open manner. Further, it explains the conversion of UML models into their respective RDFs and then querying those RDF to get the desired information. This makes easier to use the UML models of existing applications to be used for querying RDFs rather than making new RDFs of each application. In this report UML is used as a base, which is further, converted into in RDF. Sesame server is [usrd](#) for querying the RDFs obtained.

# Table of contents

---

Certificate.....	i
Acknowledgment.....	ii
Abstract.....	iii
Table of content.....	iv
List of figures.....	vi
Chapter 1: Introduction.....	1
1.1 Motivation.....	1
1.2 An overview of semantic web.....	2
1.3 Ontologies.....	5
1.3.1The need for ontologies for web.....	6
1.3.2 OWL as an RDF extension.....	6
1.4 Brief overview of UML (unified modelling language) .....	7
1.4.1 Benefits of using UML .....	8
1.4.2 Some reasons for why UML should be used to develop ontologies.....	9
1.5 An overview xml and xml schemas .....	11
1.5.1 Applicability of XML.....	12
1.5.2 Portability of XML .....	12
1.6 An overview of RDF (resource description framework) and RDFs	
1.6.1 RDFS ( Resource Description Framework Schema)..	16
1.6.2 Transforming RDF descriptions.....	17
1.7 Xpetal .....	17
1.8	
Sesame.....	17

1.8.1 The sesame library.....	18
1.8.2 The sesame server.....	18
1.8.3 Repositories and inferencing.....	18
1.8.4. An overview of the sesame architecture.....	19
1.8.5 Querying the RDF and storing the result .....	20
1.8.6 Deploying sesame on tomcat.....	20
Chapter 2 Literature review.....	24
Chapter 3 Problem statement.....	26
3.1 Problem definition.....	26
3.2 Methodology used to solve the problem.....	26
Chapter 4	
Implementation.....	29
Chapter 5 Conclusion and Future scope.....	39
References.....	40
List of Publications.....	42

## List of figures and tables

---

Figure 1.1: Semantic web layer cake.....	3
Figure 1.2: Graph representation of a "resource - property - value" – a triple.....	15
Figure 1.3: Sesame server.....	18
Figure 1.4: The sesame architecture.....	19
Figure 1.5: Apache tomcat where sesame server is deployed.....	21
Figure 1.6: Selecting the repository.....	22
Figure 1.7: Configuring the sesame repository.....	22
Figure 1.8: Logging in.....	23
Figure 1.9: Adding data to the repository in sesame.....	23
Figure 4.1: UML diagram of movie ontology.....	28
Figure 4.2: The Xpetal command.....	28
Figure 4.3: The RDF file generated after conversion from mdl file.....	29
Figure 4.4: Configuring tomcat.....	29
Figure 4.5: Deploying the sesame.war file on tomcat.....	30
Figure 4.6: Starting the sesame server configuration.....	30
Figure 4.7: Configuring sesame.....	31
Figure 4.8: Loading the configuration from a running server.....	31

Figure 4.9: Sending the configuration to sever.....	32
Figure 4.10: Setting the passwords.....	32
Figure 4.11: The "users" configuration tab.....	33
Figure 4.12: The "repository" tab.....	34
Figure 4.13: Configuring to change the access rights.....	35
Figure 4.14: Changing the access rights.....	35
Figure 4.15: The sesame entry page when logging .....	36
Figure 4.16: Adding data in sesame repository.....	37
Figure 4.17: Querying the RDF using se-rql language.....	37
Figure 4.18: Result in the subject-predicate-object form.....	38
Figure 4.19: Result in tabular form.....	38

# Chapter 1

## INTRODUCTION

---

### 1.1 Motivation

An ample amount of information is now a days available via the World Wide Web which has led to enhancement, effectiveness and selectivity of various search engines. The World Wide Web is possible only because of a set of broadly established standards that guarantee interoperability at various levels. Till now, the Web has been intended for direct human processing, but the next-generation Web called the “Semantic Web”, aims at machine-processible information. The word “Semantic” refers to “meaning” – a semantic web is actually a web of meaning or it can be said that it is a web that knows what the entities on the web mean and can make use of that knowledge efficiently [1]. It can only become possible if further levels of interoperability have been established. Standards have to be defined not only for the syntactic form of documents, but also for their semantic content. Semantic Web will enable intelligent services, which offer greater functionality and interoperability than current stand-alone services and for the semantic web to function properly and efficiently, computers must have an access to structured collections of information and sets of information and sets of inference rules that they can use to conduct automated reasoning of reference rules which can be used use to carry out automated reasoning. The aim of the semantic web is to make the WWW agents friendlier than their present scenario. The World Wide Web has become the largest database in the world. But finding relevant information is more and more difficult. To solve this problem, the W3C recommends the use of metadata and their encoding in RDF.

Basically, all the applications of real world are based on different models. An application is first designed and then finally modeled. The creation and maintenance of models does not take place in a vacuum. All the time we create models against the backdrop of secondary resources, which are necessary to understand the model correctly and also to interpret implications during its evolution. Currently, the only means of representing that information is to include it in the model in its entirety. All

reasoning on that additional information is meant to take place in the structure of the meta model. Unfortunately, there are no provisions in the UML standard that allow reasoning based on logic. Thus, the UML is lacking a defined mechanism to reason about the context of a model, to explore its background, check consistency and re-align it during evolution. Our goal is to augment the UML with loosely coupled background knowledge through the UML models, converting these models to a computer friendly language which is taken to be RDF here and with reasoning support on this knowledge performing queries using sesame server and different query languages.

## **1.2 An overview of Semantic Web**

Currently, the World Wide Web is primarily composed of documents written in HTML. Most of the information on the Web is designed only for human consumption. Inherent meaning is not shown in a way that allows their interpretation by computers. The information on the Web can be defined not only for display purposes, but also for interoperability and integration between systems and applications. One way to enable machine-to-machine exchange and automated processing is to provide the information in such a way that computers can understand it. The concept of semantic web is to extend the current web in such a way that information is given a well defined meaning so as to enable applications to work in cooperation. For this, various annotation languages have been developed to annotate information related to metadata. Meta data is actually data about data i.e one data giving information about another data. While meta data can be generated for the content of any kind of information source, development of ontologies may become a bottleneck. To overcome this, a large number of professional ontology modeling tools are required. The Semantic Web is an evolving extension of the World Wide Web in which web content can be expressed not only in natural language, but also in a format that can be read and used by automated tools, thus permitting people and machines to find, share and integrate information more easily.[2] Humans are capable of using the Web to carry out tasks such as finding the best engineering college to study or finding the best car according to mileage. However, a computer cannot accomplish the same tasks without human direction because web pages are designed to be read by people, not machines. The semantic web is a vision of information that is understandable by computers and thus

they can perform more of the tedious work involved in finding, sharing and combining information on the web.

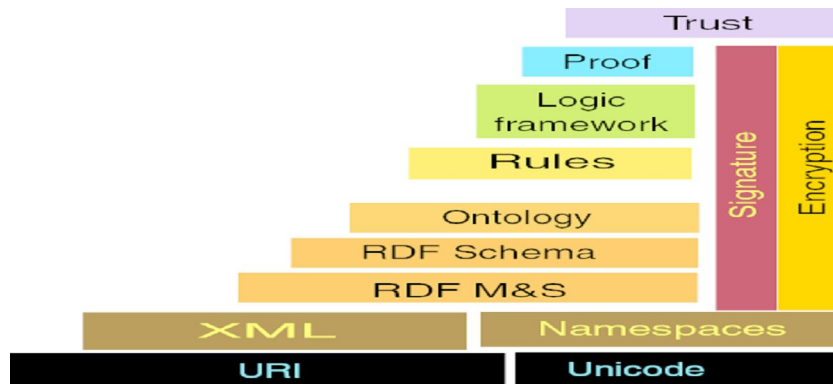


Figure 1.1 Semantic Web Layer Cake [3]

Here, RDF is layered on top of XML, which deals with data encoding while RDF deals with semantic processing. RDF represents information about resources on the Web, i.e., metadata about resources. A resource in RDF is mapped to a Uniform Resource Identifier (URI) and is described in terms of its properties. The purpose of RDF is to make statements about resources, which can be viewed as labelled edge (property) between two nodes (object and property value). The major purpose of RDF is to overcome introduce the semantic processing which was absent till the data communication took place via XML.

The description about each layer in the above semantic web layer cake is given as follows:

- **The Unicode and Uniform Resource Identifier (URI)** layers make sure that international characters sets are used and provide means for identifying the objects in the Semantic Web. The most popular URI's on the World Wide Web are Uniform Resource Locaters (URLs).
- **The extensible Markup Language (XML)** layer provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents. XML Schema is a language for restricting the structure of XML documents. This layer with namespace and schema definitions take care the Semantic Web definitions can integrate with the other XML based standards [4].

- **Resource Description Framework Schema (RDFS)** is a simple type modelling language for describing classes of resources and properties between them in the basic RDF model. It provides a simple reasoning framework for inferring types of resources [4].
- **The ontology layer** supports the evolution of vocabularies as it can define relations between the different concepts. In this layer knowledge is expressed as descriptive statements, stating some relationship exists between one thing and another.
- **A digital signature** is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document. The Digital Signature layer ensures that the original content of the message or document is unaltered.

The top layers Logic, Proof and Trust, are currently a topic of research and for the same, simple application demonstrations are being constructed. The Logic layer enables the writing of rules while the Proof layer executes the rules and evaluates together with the Trust layer mechanism for applications whether to trust the given proof or not. By adding semantic information to Web, a user can augment the effectiveness and selectivity of search engines and other automated document processing tools.

The Semantic Web enables computers to seek out knowledge distributed throughout the Web, mesh it, and then take action based on it. Resource Description Framework (RDF) is the W3C standard for encoding knowledge. [5] The knowledge available on the current web is very limited in scope and usage. For example a Wikipedia page, which might convey a lot of information to the human reader, but to the computer displaying the page all it sees, is presentation markup. To computers these are just pages of HTML, images, Flash, etc. The knowledge that these files are conveying to the human is absolutely opaque to the computers, displaying that information. The "semantic" in Semantic Web is not that computers are going to understand the meaning of anything written in various documents, but it simply mean that the logical pieces of meaning can be manipulated by a machine to useful *human* ends. Now a new web can be imagined where the actual content can be manipulated by computers intelligently. It can also be seen as a web of databases. Let's assume that one semantic website publishes a database about a product line, with products and descriptions,

while a second publishes a database of product reviews. A third site for a retailer publishes a database of products in stock. Some standards can make it easier to write an application to mesh dispersed databases together, so that a computer could use the three data sources together to help an end user make improved purchasing decisions.

### **1.3 Ontologies**

*Ontology is a formal, explicit specification of a shared conceptualization.* The idea is to devise new ways of constructing knowledge-based systems, such that these systems need not to be constructed from scratch, but the changes can be done only by assembling reusable component. Ontology consists of entities, attributes, and interrelationships between entities, domain vocabulary and factual knowledge—all connected in a semantic manner. Ontology is often viewed as the key means through which the Semantic Web vision can be realized. Ontology agrees upon unambiguous vocabulary to be used in reusable and sharable form. Originally introduced by Aristotle, *ontology is* a formal model about how we perceive a domain of interest and provide a precise, logical account of the intended meaning of terms, data structures and other elements modeling the real world. [6] Ontology provides a common understanding of a particular domain. Ontology provides interoperability among software systems and improves the design and the quality of software systems.

When we talk about ontology there are two or more agents communicating together for establishing a common terminology. Agents are nothing but applications which help in establishing friendly messages between two different types of terminologies. An efficient communication between two or more agents, in order to exchange knowledge, needs a “common understanding” of the concepts [7]. This means that agents does not need to agree about the terms used for the concepts described, but need to have an agreement about the meaning of these terms and the relationships among those terms. One major problem that might emerge is that two agents use the same term but with a different semantics. The two agents would believe they talk about the same thing, but they do not. Similar problems might emerge when two agents use different terms that have the same meaning. The agents would not recognize that they actually talk about the same thing. The aim of ontologies is to improve the efficiency and consistency in communication between agents. They do

this by defining hierarchical organized vocabularies, which describe the concepts of the domain of interest, the relationships among these concepts and their properties. The concepts are sometimes also called classes or categories used to model the world [7]. These categories and their relationships are clearly defined in the domain. Properties or slots, describe the “various features and attributes of the concept”. Typical relationships between concepts are is-a- and part-of –relationships. The first one can be compared with generalisation and inheritance from object-oriented programming, the later is sometimes referred to as the part-whole–relationship. Instantiation is another important relationship between ontologies [8] which describes the relation between a concept and the instances of that concept. Instances are “entities of the world”.

### **1.3.1 The Need for Ontologies on the Web**

Ontology is a description of a domain in terms of entities and relationships that exist between those entities. Ontologies provide a precise understanding of the domain, i.e. They provide an unambiguous and consistent interpretation of the domain in all scenarios. They enable web to share knowledge and provide reasoning support, i.e., verification of formalisms and rules. Checking for constraint violations in knowledge shared through the ontology can be automated.

### **1.3.2 OWL as an RDF extension**

The dual role given to core classes and properties makes the RDF model unique and difficult to formalize [8]. This makes it difficult to write ontologies using RDF. RDF is an odd fit in the Semantic web protocol stack where it is required to inter-operate with other knowledge representations and also interface with higher-level descriptive languages. OWL was proposed by the W3C Web Ontology Working Group as an extension to RDF with the aim of overcoming the limitations of RDF like lack of support to express class relations, lack of local restrictions on property values and lack of constructs to express property relations [9]. OWL is heavily influenced by RDF, RDF schema and the ontology description language DAML+OIL . OWL is similar to RDF schema where it structures the knowledge in a class-subclass hierarchy. Additionally, OWL allows classes to be specified as logical combination

(conjunction, disjunction and intersection) of classes or as enumerations (list, union, etc.) of specific objects. OWL extends the hierarchy to properties allowing relations between properties and sub-properties. OWL also allows domain and range constraints to be localized [10].

#### **1.4 Brief overview of UML (Unified Modelling Language)**

Modelling has been an essential part of engineering, art and construction for centuries. Complex software designs that would be difficult for a user describe textually can readily be conveyed through diagrams. Modelling provides three key benefits: visualization, complexity management and clear communication. The Unified Modelling Language is a visual language for specifying, constructing and documenting the artifacts of systems. A particular application can use UML with all processes, through out the development lifecycle, and across different implementation technologies. UML is a widely adopted modelling language, which is adopted (don't use adopted word twice in same line) by software professionals all over the world. In order to bring ontology development process closer to wider software engineering population, some authors propose usage of software engineering techniques, especially the UML since it is the most accepted software engineering standard. [11] It is the language for visualising, specifying, constructing, and documenting the artifacts of a software-intensive system. It can be used with many processes, throughout the software development life cycle, and across different implementation technologies. It is a language and associated graphical notation for object-oriented analysis and design. It provides a framework for describing a set of models that capture the functional and structural semantics of any complex information system. UML diagrams can also be used as a design tool (class diagram) to develop Semantic Web ontologies. With this viewpoint, UML can be looked upon as a suitable candidate for knowledge representation. There are several types of UML diagrams for modelling the static and dynamic behaviour of a system, but for application to knowledge representation and exchange the two relevant diagram types are class and object diagrams. [12]

### **1.4.1 Benefits of using UML**

The Unified Modeling Language was designed by Rational Software in order to provide a standard for software modeling languages. It is now owned and controlled by the Object Modeling Group, an independent organization. UML has quickly become the de-facto standard modeling language in the software industry. UML is a universal language because it can be applied in many areas of software development. Here are few reasons for why should UML should be used for software construction at a universal level:

i) Software construction generally needs a plan and there is a common perception among general management that software is both cheap to produce and easy to change. Software is extremely complex and, once a particular structure is in place, often long-winded and difficult to change. A building architect creates a drawing of the building from more than one direction, ensuring that the structure is appropriate for the purpose of the building and that the dimensions are all consistent. Only when this visual model of the proposed work has been approved is the job of laying bricks begun. Therefore UML as a modeling language is a useful tool for software construction.

ii) Visualization in multiple dimensions and levels of detail software is very abstract and hard to visualize. A visual modeling language, such as UML, allows software to be visualized in multiple dimensions, so that a computer system can be completely understood before construction begins. Furthermore, UML can be used to produce several models at increasing levels of detail. The overall scope of the software can quickly and easily be defined at the start of the project with a high level model allowing for accurate estimation. Increasing levels of detail can then be added to each part of the software as it is constructed, until finally the software appears as code. The code can then be tested against a test model that is derived from the original model of requirements.

iii) UML is appropriate for both new system developments and improvements to existing systems. It is a fallacy to use a new modeling technique on an old system, the old system will have to be completely 're-documented' in the new style in order for any change to take place. The truth is that only those parts that are affected by the change will need to be modeled. If they were badly documented or not documented at

all, then a lot of work would need to be done to understand them anyway in order to make the change.

iv) Undocumented or badly documented software has a reduced value as a company asset. With so much of the company's business process embedded in obtuse software code, a company is vulnerable to losing an understanding of and control over the business rules by which it operates. Gradual re-engineering of the company's bespoke computer software in a way that makes the functionality of the software transparent, brings those important assets back under control. It removes the company from dependence on key personnel who may leave at any time and improves understanding of the company's critical business processes. It even improves the quality of software engineering staff because those who want to do a proper job will look for companies that model with UML.

It includes user-definable extension mechanisms so that it can be adapted for specific environments. Industry standard extensions now exist for business modeling and web-based applications development.

v) The Use Case Driven nature of modeling with UML ensures that all levels of model trace back to elements of the original functional requirements. This traceability between models comes without the extra effort of creating and maintaining a 'traceability matrix' which can be a complex and time consuming job. The result is that the impact of a requested change can quickly be estimated. A change in requirements can be traced through analysis and design models into those components affected and even lines of code. The code affected can then be traced back to the requirements and total regression testing effort calculated. Accommodates Incremental Development and Re-Development UML models respond well to an incremental development environment. It is possible to develop only those parts of the model that are required to satisfy the new requirements, and the code needed to fulfill those requirements is in place. This approach ensures that only the work needed to fulfill the current requirement is done, while still developing in such a way that maximizes re-use, maintainability and extensibility.

#### **1.4.2 Some reasons for why UML should be used to develop ontologies**

Ontologies include class/subclass hierarchies, relationships between classes, class attribute definitions and axioms that specify constraints. In UML, this ontology

information is usually modeled in class diagrams and Object Constraint Language (OCL) constraints. Other UML diagrams such as state charts and activity diagrams are also useful for service and process-related ontologies.

There are a number of good reasons why UML is a promising notation for ontologies:

- UML is a graphical notation based on many years of experience in software analysis and design by a variety of companies in a wide spectrum of industries and domains.
- UML is an open standard maintained by the OMG.
- UML has standard mechanisms for defining extensions for specific application contexts such as ontology modeling.
- UML is widely adopted in industry and current techniques for ontology development are based on knowledge representations such as Knowledge Interchange Format (KIF), which are not widely known outside the AI research community.
- Real-world industrial agent-based systems need to interact with legacy enterprise systems, which often have existing UML models.[13]

### **1.5 An Overview XML and XML Schemas**

XML (eXtensible Markup Language) is an industry-standard, system-independent way of representing data [5]. Like HTML (HyperText Markup Language), XML encloses data in tags, but there are significant differences between the two markup languages. While XML tags relate to the meaning of the enclosed text, HTML tags specify how to display the enclosed text. Unlike HTML, the tag vocabulary of XML is not restricted and it enables users to create arbitrary tags that have arbitrary meanings. Because XML tags indicate the content and structure of the data they enclose, they make it possible to do things like archiving and searching. Another major difference between XML and HTML is that XML tags are extensible and allow writing own XML tags to describe the content. With HTML, only limited tags can be used that have been predefined in the HTML specification. With the extensibility that XML provides, the required tags can be created for a particular type of document. Tags are defined using an XML schema language, which describes the structure of a set of XML documents and can be used to constrain the contents of the XML documents. The most widely used schema language is the Document Type Definition schema language. A schema written in this language is called a DTD. The DTD

defines the tags used in the price list XML document. It specifies four tags (elements) and further specifies which tags may occur (or are required to occur) in other tags. The DTD also defines the hierarchical structure of an XML document, including the order in which the tags must occur.

### **1.5.1 Applicability of XML**

XML allows any type of data to be encoded, as there are no restrictions on XML tags. The only constraint is that a well-formed and valid XML document conforms to a DTD or schema, that is it requires the XML file adhere to a pre-defined grammar. The XML schema provides a syntactic description of data and does not provide semantics to the data. If web users all over the world will use the same terms to describe entities, then parsing a XML file will yield both syntactic and semantic information. XML Schemas use a restricted tag set to describe the structure of an XML document and are syntactically constrained to form a balanced tree (i.e., every starting tag must have a corresponding closing tag). RDF overcomes the semantics limitations to a great extent and can represent semantics more efficiently than XML [4].

### **1.5.2 Portability of XML**

A DTD, such as the university DTD, gives XML data its portability. If an application is sent a university document in XML format and has the university DTD, it can process the document according to the rules specified in the DTD. For example, given the university DTD, a parser will know the structure and type of content for any XML document based on that DTD. If the parser is a validating parser, it will know that the document is not valid if it contains an element not included in the DTD, such as <student>, or if the head element precedes the name element. Other features also contribute to the popularity of XML as a method for data interchange. It is written in a text format, which is readable by both human beings and text-editing software. Applications can parse and process XML documents, and errors can be easily detected and processed as XML is human readable and easily understandable. Another feature is that because an XML document does not include formatting instructions, it can be displayed in various ways.

## 1.6 An Overview of RDF (Resource Description Framework) and RDFS

The Resource Description Framework (RDF) is an up-coming standard for expressing simple meta-data content. RDF Schema provides a simple mechanism for defining meta-data schemas or ontologies. As its name signifies, it is not a language as such but a model for representing data about “things on the Web “. This type of data about data is called metadata. The “things” are resources in RDF vocabulary. RDF’s basic data model is simple: besides resources, it contains properties and statements. A property is a specific aspect, characteristic, attribute, or relation that describes a resource. A statement consists of a specific resource with a named property plus that property’s value for that resource. An RDF description is a list of triples: an object (a resource), an attribute (a property), and a value (a resource or free text). [14]

RDF was originally created in 1999 as a standard on top of XML for encoding metadata--literally, data about data. The most exciting uses of RDF aren't in encoding information about web resources, but information about and relations between things in the real world: people, places, concepts, etc. This mechanism for describing resources is a major component in proposed by the W3C's Semantic Web activity. This activity is an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity. As mentioned above, RDF provides a general, flexible method to decompose any knowledge into small pieces, called triples, with some rules about the semantics (meaning) of those pieces. [15]

Most of the abstract model of RDF comes down to four simple rules:

1. A fact is expressed as a Subject-Predicate-Object **triple**, also known as a **statement**. It's like a little English sentence.
2. Subjects, predicates, and objects are given as names for **entities**, also called **resources** (dating back to RDF's application to metadata for web resources) or

**nodes** (from graph terminology). Entities *represent something*, a person, website, or something more abstract like states and relations.

3. Names are **URIs**, which are global in scope, always referring to the same entity in any RDF document in which they appear.
4. Objects can also be given as text values, called **literal values**, which may or may not be typed using XML Schema data types.

The main elements of RDF are the root element, <RDF>, and the <Description> element, which identifies a resource. The document written in RDF is written as follows:

```
<xml version =”1.0”?>
<RDF: RDF
xmlns: RDF =http://www.w3.org/1999/02/22-RDF-syntax-ns#>
.....
.....
.....
</RDF: RDF>
```

the <rdf: RDF> Element:

<rdf: RDF> is the root element of an RDF document. It defines the XML document to be an RDF document. It also contains a reference to the RDF namespace.

The <RDF: Description> Element:

The <RDF: Description> element identifies a resource with the about attribute. It gives the detailed description of what the particular RDF is all about.

Main points about RDF are:

RDF is Designed to be Read by Computers-RDF was designed to provide a common way to describe information so it can be read and understood by computer applications.RDF descriptions are not designed to be displayed on the web.

- RDF is written in XML – The documents in RDF are written in XML. The XML language used by RDF is called RDF/XML. By using XML, RDF information can easily be exchanged between different types of computers using different types of operating systems and application languages.
- RDF is a W3C Recommendation- A W3C Recommendation is understood by the industry and the web community as a web standard. A W3C Recommendation is a stable specification developed by a W3C Working Group and reviewed by the W3C Membership.

The base is breaking knowledge down into a directed labeled graph. Each edge in the graph represents a fact, or a relation between two things. The edge in the example from the node subject is related to predicate i.e. Actor AB starred in the movie movie1. A fact represented in this manner has three parts: a subject, a predicate (i.e., verb), and an object. The subject is what's at the start of the edge, the predicate is the type of edge (its label), and the object is what's at the end of the edge. Most of the abstract model of RDF comes down to four simple rules. A fact is expressed as a Subject-Predicate-Object **triple**, also known as a **statement**. It's like a little English sentence.

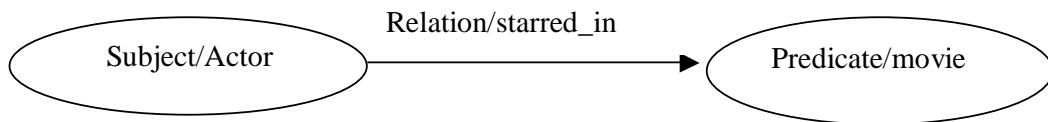


Figure 1.2 Graph representation of a "resource - property - value" – a triple

The Resource Description Framework (RDF) is a language for representing meta-data. The *RDF data model* defines the structure of the RDF language. The data model consists of three data types.

**Resources:** All data objects described by a RDF statement are called resources. For example, resources are web sites or books.

**Properties:** A specific aspect, characteristic or relation of a resource is described by a property. For example, properties are the creation date of a web site or the author of a book.

**Statements:** A statement combines a resource with its describing property and the value of the property. RDF statements are the structural building blocks of the language.

A RDF statement is typically expressed as “resource - property - value” - triple, commonly written as P(R, V): A resource R has a property P with value V. These triples can also be seen as object-attribute-value triple. Statements can also be expressed as graphs with nodes for resources and values where directed edges represent the properties. Figure 1 shows the graph of the resource R with an edge for the property P directed to the property value V. Resources are represented in the graph as circles, properties are represented by directed arcs, (property-) values are represented by a box. These values are called graph end node and these values can also become resources if they are described by further properties, i.e., if a value forms a resource in another triple. They are then represented by a circle.

### **1.6.1 RDFS ( Resource Description Framework Schema)**

RDF Schema provides a simple mechanism for defining meta-data schemas or ontologies. Basically, RDF Schema is a simple type system for RDF. It gives a mechanism to define domain-specific properties and classes of resources to which those properties can be applied. The RDF data model does not make any assumptions about the application area in which the data is used. There are no reserved terms to model the data. Additionally, the RDF data model has no mechanism to define names for properties or resources. For that purpose, the *RDF schema* is needed to define resource types and property names. Different RDF schemas can be defined and used for different application areas. The W3C proposed a RDF schema called *RDFS* [15]. RDF schema statements are valid RDF statements because their structure follows the structure of the RDF data model. The only difference to a resource - property - value” - triple is, that an agreement about the specific meaning for reserved terms and statements has been made. Thus, the RDF schema provides a vocabulary for defining the semantics of RDF statements. RDF asserts statements about resources in terms of properties, the description of these properties is provided by the RDF schema specification, i.e. RDF schemas add semantics to RDF. They provide a type system for resources and define relationships between resources and can be encoded using RDF resources, such as Class, SubclassOf, Property, etc. which are defined in the namespace “rdfs”. Unlike XML schemas, they do not impose syntactic constraints on the RDF document but provide a mechanism to describe metadata about resources. Every entity in RDF is a resource and a class represents a group of resources. Every

resource in RDF is an instance of at least one class. RDF schemas use a core set of classes and properties to describe metadata about web resources, i.e., there exists a subset of RDF resources using that all other RDF schema constructs can be defined. The core classes in RDF schema are `rdfs:class`, `rdfs:resource`, `rdfs:property` and `rdfs:constraintresource`. `rdfs:class` represents the set of entities used to describe concepts in a schema, `rdfs:resource` represents the set of all resources, `rdfs:property` represents the set of all RDF properties and `rdfs:constraintresource` represents the set of all constraints applicable to RDF resources. Core properties include properties of the class `rdfs:property` namely, `rdf:type`, `rdfs:subclassof`, `rdfs:range` and `rdfs:domain`. The RDF model assigns a dual role to these core classes and properties, which is used to define other schema constructs and can be used in the specification of an RDF schema itself that causes self-referentiality in the RDF model .

### **1.6.2 Transforming RDF descriptions**

Multiple RDF schemas can be found for the same domain (education for example) resulting in the lack of interoperability. So, translating RDF descriptions is mandatory if a user wants to request multiple descriptions using multiple schemas even if one doesn't know that different schemas exist. RDF descriptions are based on RDF schemas so, the translations of descriptions may be deduced from the translations of schemas. But, transforming one schema into another must be done while preserving the meaning of the schemas.

### **1.7 Xpetal**

It generates an RDF schema from a rational Rose petal file. This is used to generate an RDF equivalent of UML class diagram. The generated RDF schema can be efficiently used in various WWW applications for number of purposes. The input file and output file arguments are file names in the normal format for the platform. The RDF files used here normally end in `.ptl` or `.mdl` extension. The output file is an RDF schema file. The schema is constructed according to the UML to RDF mappings in the OMG (OMG is a consortium of around 800 member companies and institutions involved in software engineering) specification.

## **1.8 Sesame**

Sesame is an open source Java framework for storing, querying and reasoning with RDF and RDF Schema. It can be used as a database for RDF and RDF Schema, or as a Java library for applications that need to work with RDF internally. For example, suppose a big RDF file is to be read, find the relevant information for the application, and use that information. Sesame provides the necessary tools to parse, interpret, query and store all this information, embedded in the application if you want, or, if you prefer, in a separate database or even on a remote server. In short it can be said that, Sesame provides an application developer a toolbox that contains useful hammers screwdrivers etc. for doing 'Do-It-Yourself' with RDF.[16]

### **1.8.1 The Sesame library**

The Sesame library consists of a set of java archives:

- Sesame.jar. The Sesame core classes.
- rio.jar. Rio (RDF I/O) is a set of parsers and writers for different RDF serialization formats (RDF/XML, Turtle, N-Triples).
- openRDF-model.jar. Shared interfaces and classes for the RDF model.
- openRDF-util.jar. Shared utility classes.

These archives (which are located in the lib/ directory) contain Java classes ready for use in your own application.

### **1.8.2 The Sesame Server**

Sesame can be used as a Server with which client applications (or human users) can communicate over HTTP. Sesame can be deployed as a Java Servlet Application in Apache Tomcat, a webserver that supports Java Servlets and JSP technology.

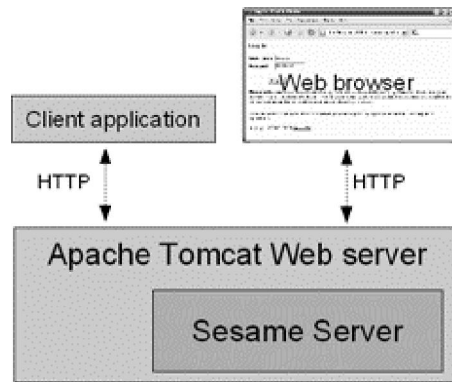


Figure 1.3 Sesame Server [17]

### 1.8.3 Repositories and Inferencing

A central concept in the Sesame framework is the *repository*. A repository is a storage container for RDF. This can simply mean a Java object (or set of Java objects) in memory, or it can mean a relational database. Whatever way of storage is chosen however, it is important to realize that almost every operation in Sesame happens with respect to a repository: when RDF data is added, it is added to a repository. When a user does a query, actually he queries a particular repository.

Sesame, as mentioned, supports RDF Schema inferencing. This means that given a set of RDF and/or RDF Schema, Sesame can find the implicit information in the data. Sesame supports this by simply adding all implicit information to the repository as well when data is being added.

### 1.8.4. An Overview of the Sesame Architecture

In Figure, an overview of Sesame's overall architecture is given.

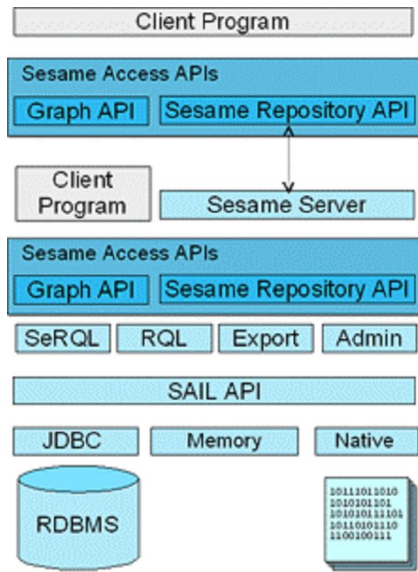


Figure 1.4 The Sesame architecture

Starting at the bottom, the Storage And Inference Layer, or *SAIL API*, is an internal Sesame API that abstracts from the storage format used (i.e. whether the data is stored in an RDBMS, in memory, or in files, for example), and provides reasoning support. SAIL implementations can also be stacked on top of each other, to provide functionality such as caching or concurrent access handling. Each Sesame repository has its own SAIL object to represent it. On top of the SAIL, there are Sesame's *functional modules*, such as the SeRQL, RQL and RDQL query engines, the admin module, and RDF export. Access to these functional modules is available through Sesame's *Access APIs*, consisting of two separate parts: the *Repository API* and the *Graph API*. The Repository API provides high-level access to Sesame repositories, such as querying, storing of RDF files, extracting RDF, etc. The Graph API provides more fine-grained support for RDF manipulation, such as adding and removing individual statements, and creation of small RDF models directly from code. The two APIs complement each other in functionality, and are in practice often used together.

The Access APIs provide direct access to Sesame's functional modules, either to a client program (for example, a desktop application that uses Sesame as a library), or to the next component of Sesame's architecture, the *Sesame server*. This is a component that provides HTTP-based access to Sesame's APIs. Then, on the remote

HTTP client side, APIs can be accessed again, which can again be used for communicating with Sesame, this time not as a library, but as a server running on a remote location.

### **1.8.5 Querying the RDF and storing the result**

After getting the RDF of the particular UML file, the generated RDF is used for querying. Since the output generated in RDF is as such not a human readable language, some kind of software or tool is needed which can take the input as RDF, understands it and then gives the output in the desired format. The desired format can be further an RDF file which can be used elsewhere in the web or it can be a file which is human readable means output is displayed in a user friendly manner in that case. So, for that purpose a java framework tool known as Sesame is used. This framework once established can be used to query the RDFs and obtain the result in required form. For using Sesame the first thing that is required is Apache tomcat (where this framework is deployed).

### **1.8.6 Deploying Sesame on Tomcat**

**Apache Tomcat** is a web container, or application server developed at the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server. [18]

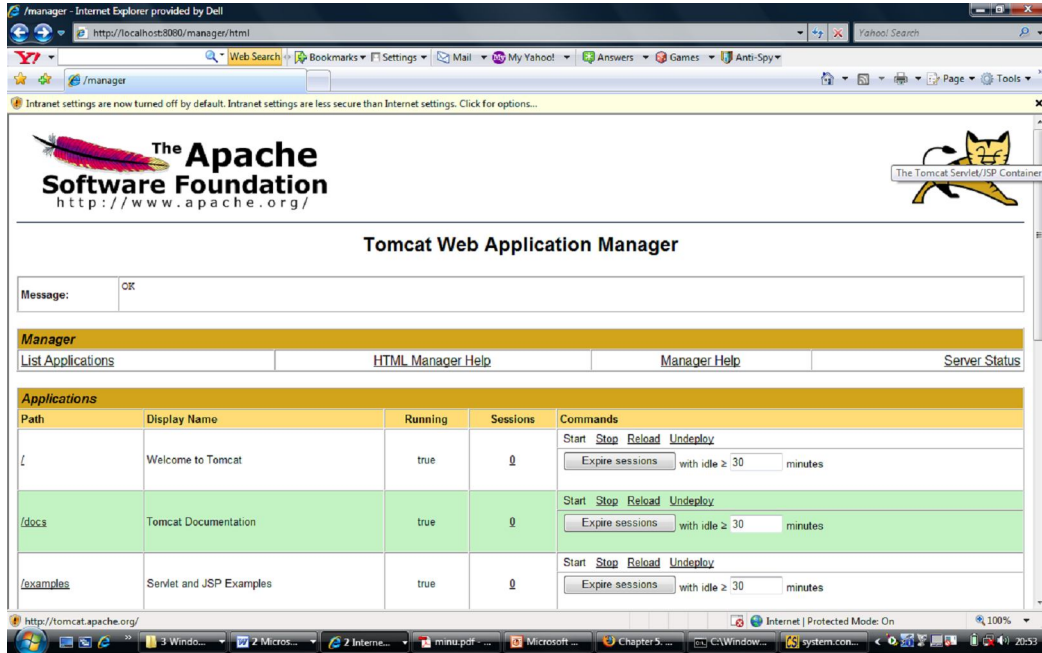


Figure 1.5 Apache Tomcat where Sesame server is deployed

### The Sesame entry Page

Once the Sesame server is deployed on Tomcat it runs at localhost, port 8080. The war file is to be deployed and then configuration of Tomcat is to be done using configsesame.bat utility.

After this file is run successfully the access to the Sesame repository will be obtained.

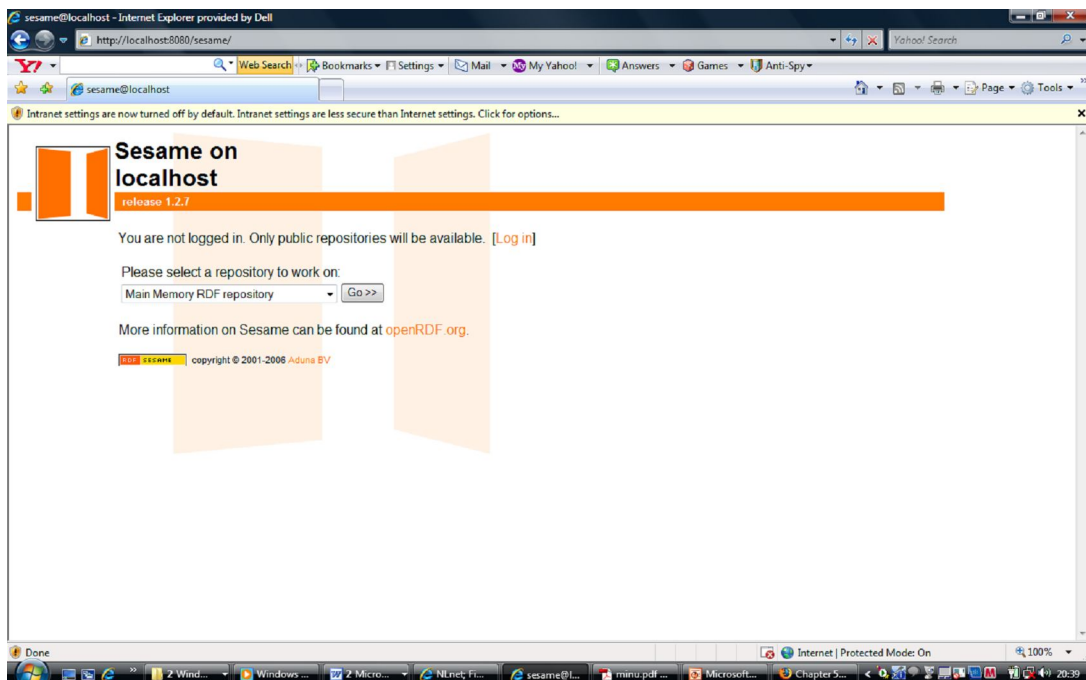


Figure 1.6 Selecting the repository

The Sesame login page before configuring the repository

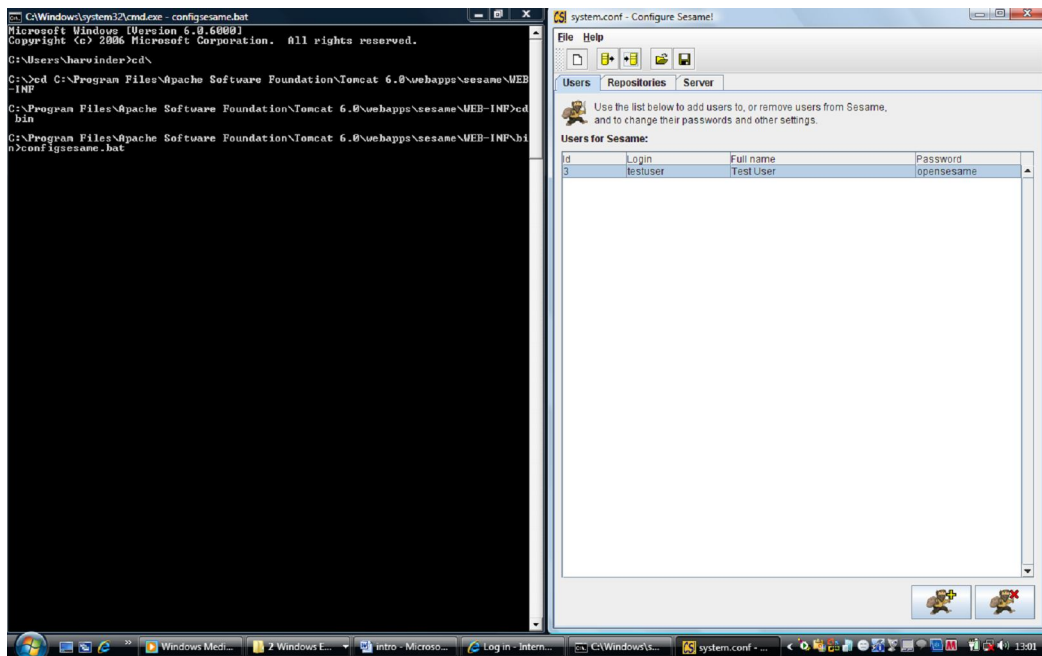


Figure 1.7 Configuring the Sesame repository

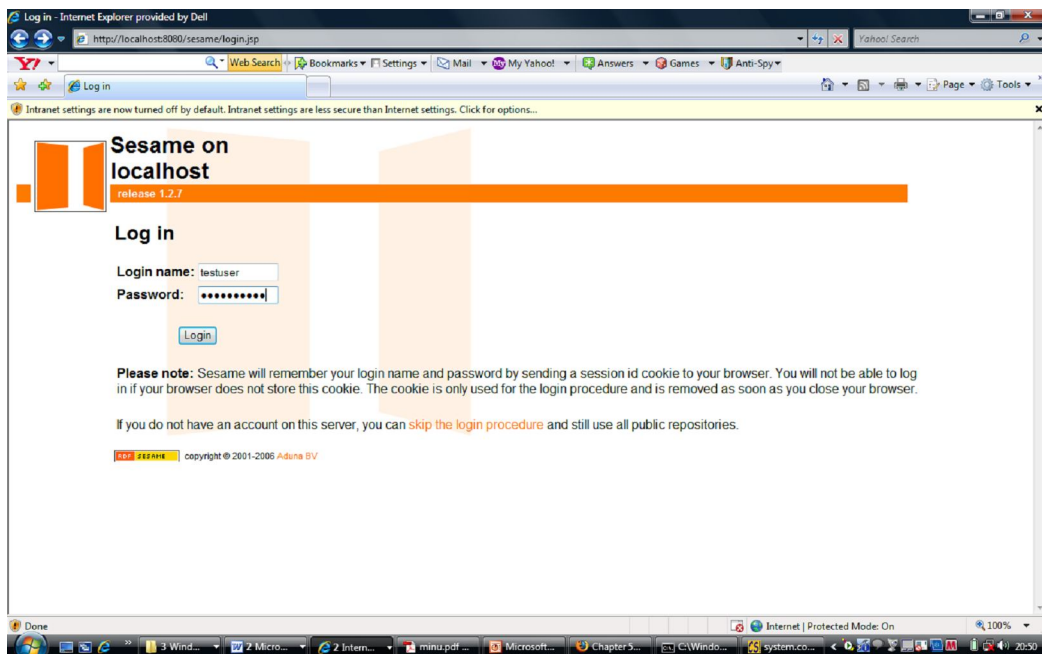


Figure 1.8 Logging in

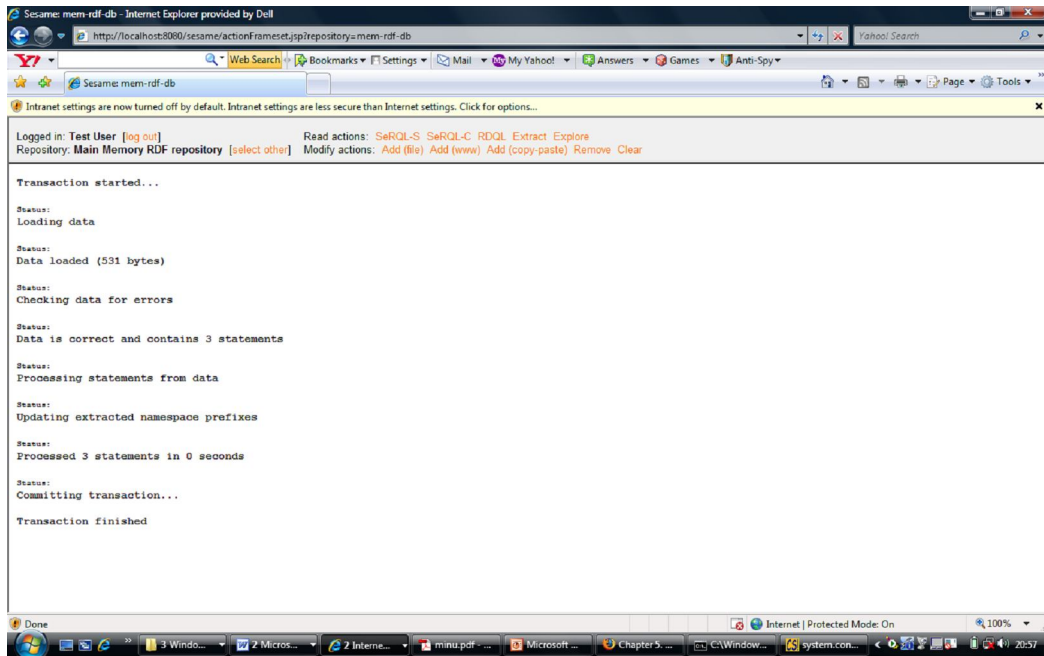


Figure 1.9 Adding data to the repository in Sesame

## Chapter 2

### Literature Review

---

Today, the World Wide Web is a network of different information resources. The content, usually marked up in HTML, is kept in the nodes of this huge heterogeneous network. It was designed almost solely for humans to read and understand. Increase in size of web combined with the availability of new technologies and applications, a strong need arises for computer programs to access information “stored” in the World Wide Web.

A lot of knowledge is available on the web but this knowledge sometimes is not utilized fully by human beings. The web was designed as an information space, with the goal that it should be useful not only for human-human communication, but also that machines would be able to participate and help. One of the major obstacles to this has been the fact that most information on the Web is designed for human consumption, and even if it was derived from a database with well defined meanings for its columns. The Semantic Web approach instead develops languages for expressing information in a machine processable form. The Semantic Web is a web of data, in some ways like a global database. The question was how to access this vast amount of information and further making this information available in a computer readable form.

Extensible Markup Language (XML) and data-language standards such as Resource Description Framework (RDF) and Web Ontology Language (OWL), two World Wide Web Consortium’s (W3C) standards are components of Semantic Web Stack. These standards and descriptors enable Web developers to add layers of meaning to Web documents, supplying a framework for defining how data is linked and how its intended relationships are expressed .

The Extensible Markup Language (XML) is a generic, very flexible but simple text format, which was derived from the SGML standard. By utilizing element tags and attributes, it is used for marking up semi-structured data. It allows adding arbitrary structure to a human- and machine-readable document. Therefore, XML can be used as data exchange format. Since XML is capable of defining the rules for such tree-structured documents, it is called a meta language.

XML Schema is a schema definition language expressed in XML syntax . It describes the structure of an XML document and provides means to constrain the content, i.e., names and values of elements and attributes. A schema is an XML document itself and supplies methods to define the structure, content, and semantics of XML documents.

RDF, The Resource Description Framework (RDF) has been developed by theW3C to encode metadata. It can be used to describe any kind of resource. These resources can be accessed and uniquely identified through a URI. This allows for the distribution of a resource's description over different nodes. RDF can be represented in many ways, e.g., with a graph, with triples, or in an XML mark-up called RDF/XML .

RDF terminology has three components:

- a subject, which is a resource.
- a predicate, which is a property of the resource.
- and an object, which is the value of the property.

In RDF, information is simply a collection of statements, each with a subject, verb and object. Everything, be it subject, verb, or object, is identified with a Uniform Resource Identifier. This is something like `<http://www.w3.org/>` or `http://www.w3.org/2000/10/swap/test/s1.n3#includes` .

The RDF Vocabulary Description Language, RDF Schema (RDFS), adds semantics to an RDF data model by defining its vocabulary. It is similar to XML and XML Schema [39], in a way that it transfers a general RDF concept to a specific application domain. Written in RDF, the schema description provides possibilities to characterize groups of related resources and their relationships. Moreover, it allows for inference with the use of metadata.

Xpetal is used for conversion of UML files into RDFs and Sesame is a server deployed on tomcat in order to query the rdf files for the use of applications.

---

### **3.1 Problem definition**

During the literature review it was found that implementation of semantic web can be done using some kind of inferences between various applications on the World Wide Web. There are numerous amount of information available on net and to use it in a user friendly manner a common terminology is needed amongst all the applications. To solve this purpose ontologies are developed but the problem that arises from ontologies is that the terminologies of different people all across the world are different. It has been widely accepted in the computer research community that ontologies will serve the basis for the fulfillment of vision of Semantic Web, i.e. a web with reasoning, intelligence and vision but the major problem faced is that ontologies can be represented in various data formats, and terminologies and mapping them is a major problem. To overcome this a solution is opted for a world wide accepted standard format: UML. RDF serve as a base for adding intelligence to the data by means of triplets i.e. subject, object and predicate but again the problem foreseen was how to convert a particular UML file into an RDF file and generate the output in a computer readable, application friendly format which can be used for the communication between various applications on the vast web. The RDF file obtained should be comfortably queried as per user's requirements. The solution of the problem is sorted by using xpetal to convert UML to RDF and further used Sesame to query the RDF in a human readable form.

### **3.2 Methodology used to solve the problem**

Here is the step by step procedure followed to solve the above problem:

- The UML class diagram of the particular ontology is generated using rational rose(an .mdl file will get generated as output)
- Xpetal tool is installed in order to convert this particular file into RDF format.
- Java and Tomcat server are installed. After configuring the server, it is run on port 8080 and then Sesame is deployed on it.

- To configure the repository of Sesame (where all data is stored) the bat file of Sesame is run at command prompt. This utility will run at the backend till the time we are working on Sesame server.
- Querying of database is possible once the user log on to Sesame server using username and password.
- The output generated can be obtained as a table or RDF file or html file.

## Chapter 4 Implementation

---

The implementation starts with making an UML diagram. An UML diagram representing a movie model is presented here. In this ActorAB is a class which is associated with movie1 and movie2. ActorAB starred in a movie and a TV show. The name of TV show is KBC and movie1 has a similar plot to movie2.

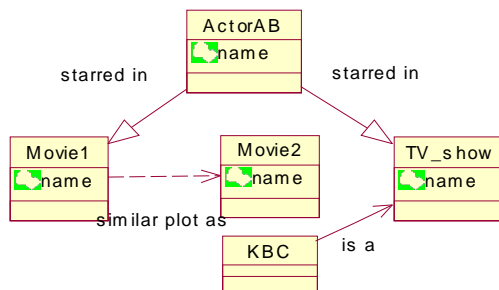


Figure 4.1 UML diagram of movie ontology

Once UML diagram is obtained, the next step is to convert it to a file which can be used for further querying. This is done with the help of Xpetal command. This command is run after we've successfully installed jre environment and configured Xpetal properly. After all configurations are set run the following command.

```
CA Command Prompt
C:\xpetal>java -jar xpetal.jar -rdf c:\xpetal\movie.mdl try
C:\xpetal>_
```

Figure 4.2 The Xpetal command

After this a file named try will contain the output file of the input file which was movie.mdl. The mdl file is thus converted into an RDF file. This file is not human readable. This file is meant for applications or computers. The RDF files generated named try looks as follows:

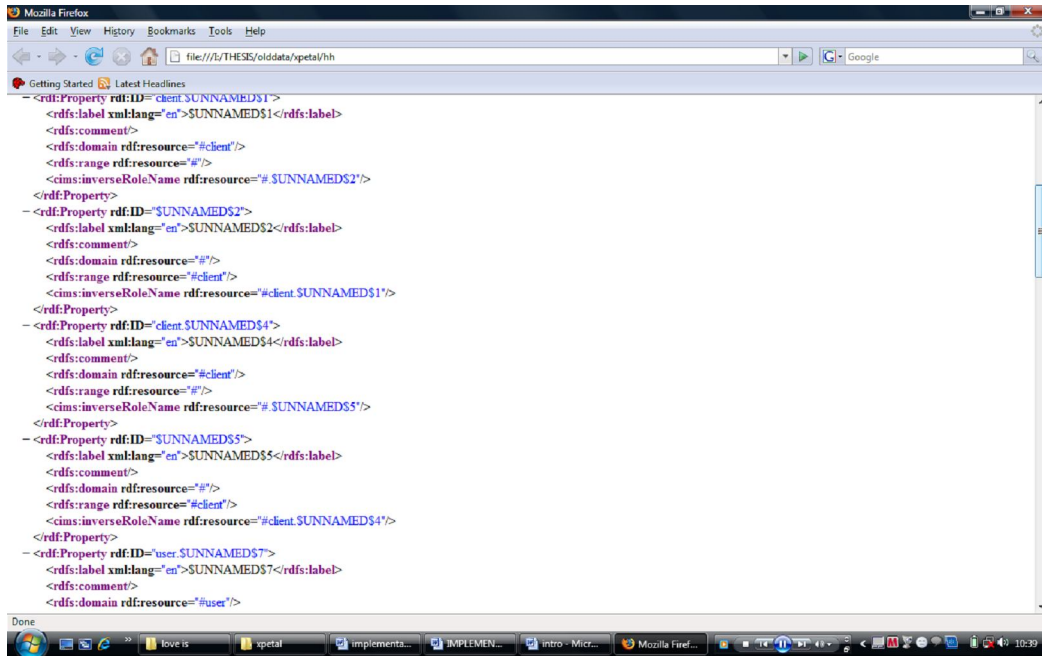


Figure 4.3 The RDF file generated after conversion from mdl file

Once the RDF is obtained next is to query the RDF file.

Configuration of Tomcat server before deploying Sesame server on it is :

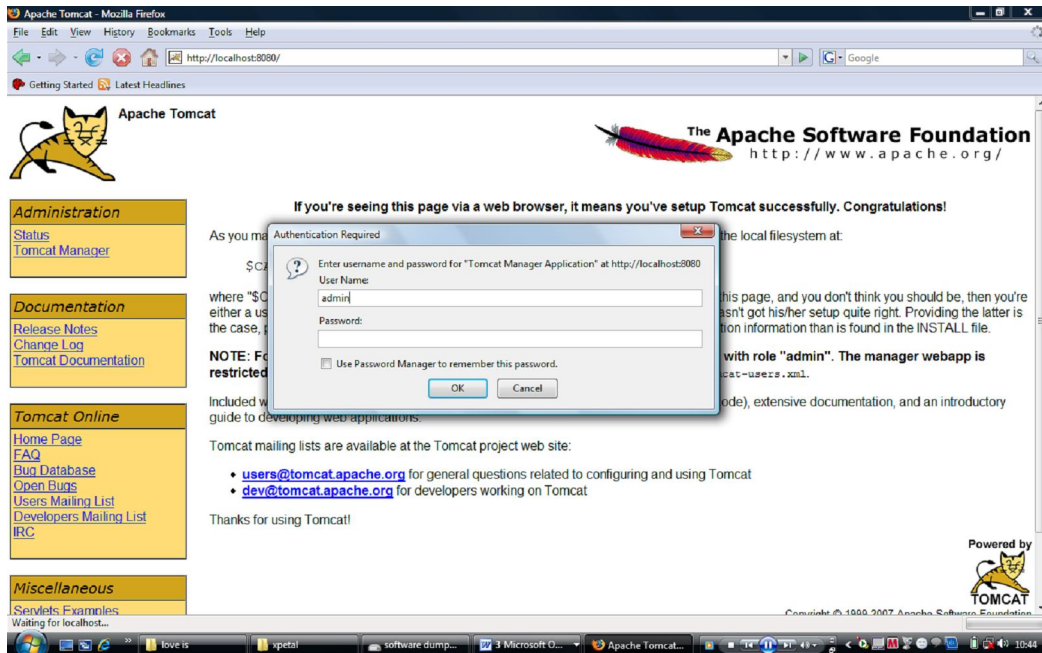


Figure 4.4 Configuring Tomcat

After configuring Tomcat, sesame server is configured. First of all, Sesame server is deployed on Tomcat by deploying the war file onto the Tomcat server.

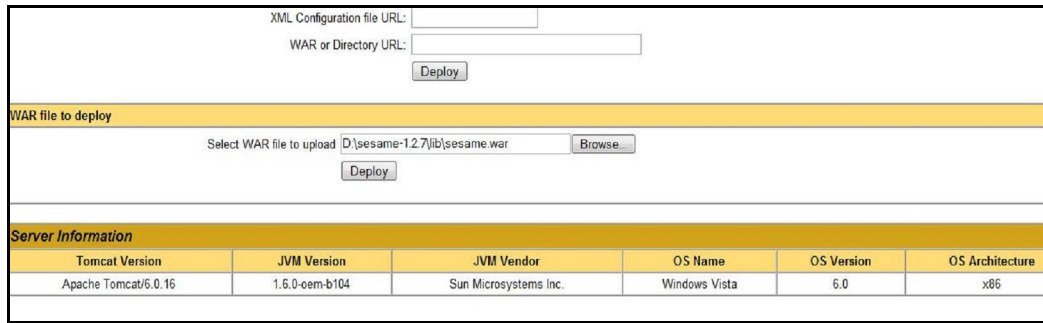


Figure 4.5 Deploying the Sesame.war file on Tomcat

After deploying this file Sesame is configured to start the task of querying.

It is done by running the config.bat file located at C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\Sesame\WEB-INF\bin.

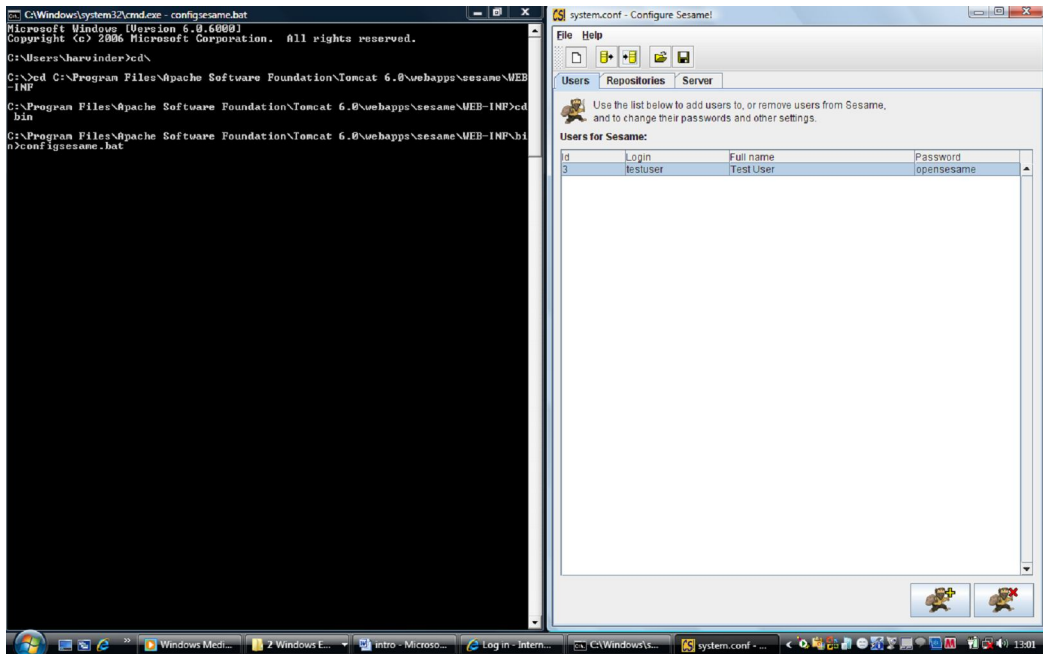


Figure 4.6 Starting the Sesame server configuration

Loading a system configuration

Sesam server is cnfigured via :



Figure 4.7 Configuring sesame

When we start Configure Sesame, the file system.conf is loaded

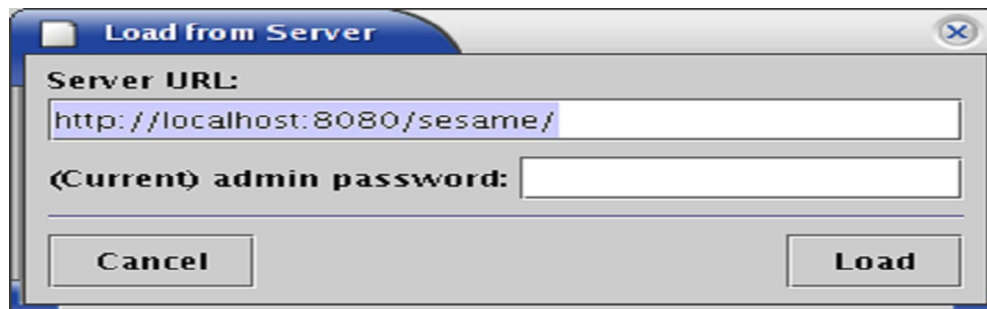


Figure 4.8 Loading the configuration from a running server

If this is a fresh installation of Sesame, the admin password is still the default password. In this case, admin is the Current admin password to load the configuration. If this is not a fresh installation of Sesame admin password has to be entered.

Storing a system configuration

When the configuration is modified, it is stored on disk (menu option [File]-->[Save file as...]), or it can be sent directly to a running Sesame server (menu option [File]-->[Send to server...]). Using this option the changes are immediately sent to the sever.

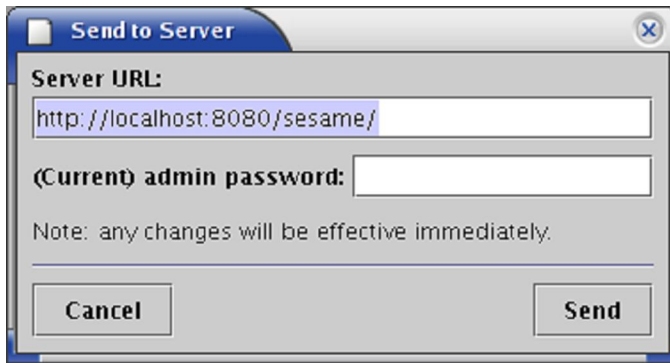


Figure 4.9 Sending the configuration to sever

### Setting the admin password

Admin password is set in server tab.

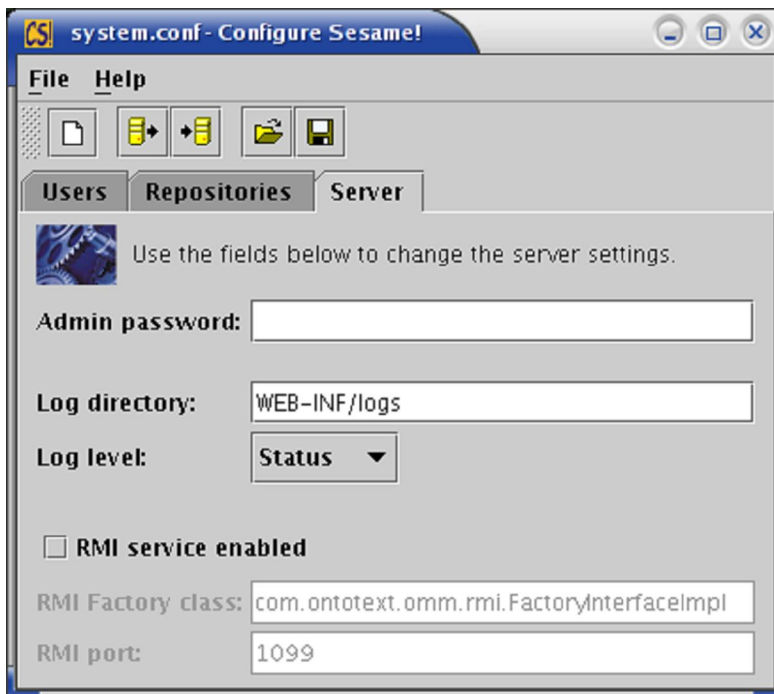


Figure 4.10 Setting the passwords

### Adding and removing user accounts

User account is added to sesame by following these steps:

Open the Tab "Users" and there we can configure the users which can use the sesame server. Only using this user name and password can we login to the server. By default one user is created with username testuser and password "opensesame".

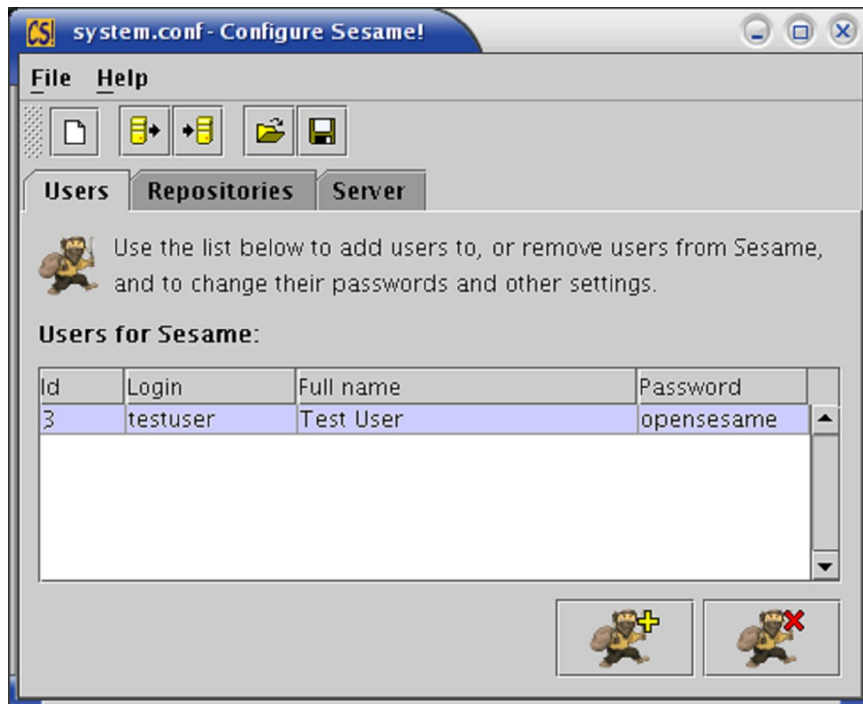


Figure 4.11 The "Users" configuration tab

1. By clicking the "Add user" icon on the bottom of the window, a new entry is added to the overview.
2. A new entry is added to the overview by using "Add User icon" on the bottom of the window.
3. Credentials for the new user are added by hitting Enter button.

To remove a user account, simply select the user you wish to remove and click the "Remove user" button in the bottom right of the window.

### Configuring repositories

Repositories are configured in order to add or remove the required data in the database. Existing repositories can be configured and new ones can be added using Configure Sesame utility.

### Editing an existing repository configuration

1. "Repositories" tab is opened and repository you wish to configure is selected and then click the "Repository details" button on the bottom left.

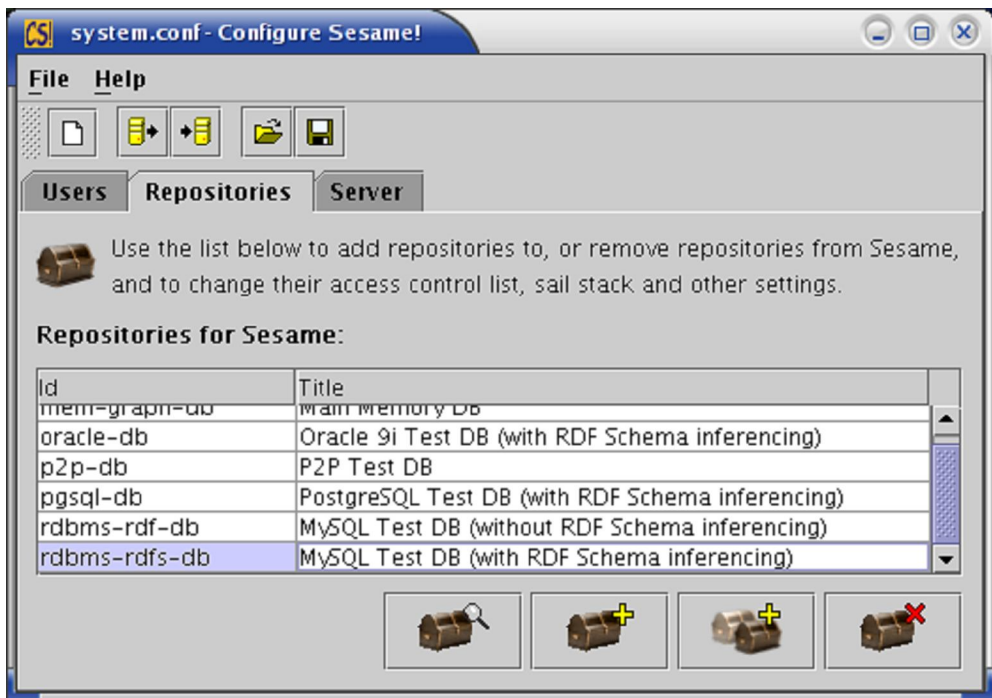


Figure 4.12 The "Repository" tab

In the Repository details window, we can edit as many repositories as we want.

## Editing the access rights

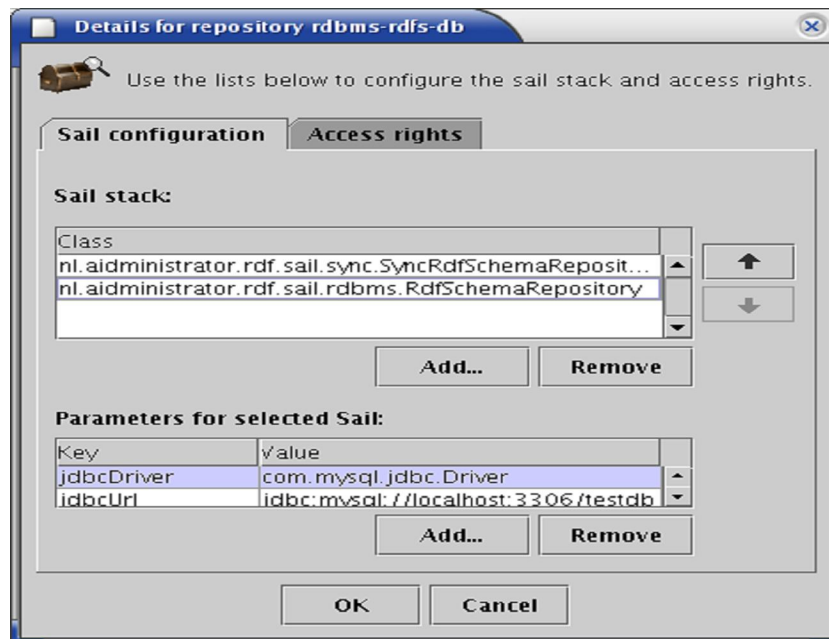


Figure 4.13 Configuring to change the access rights

The Repository details window also allows to change the access rights of users to a repository.

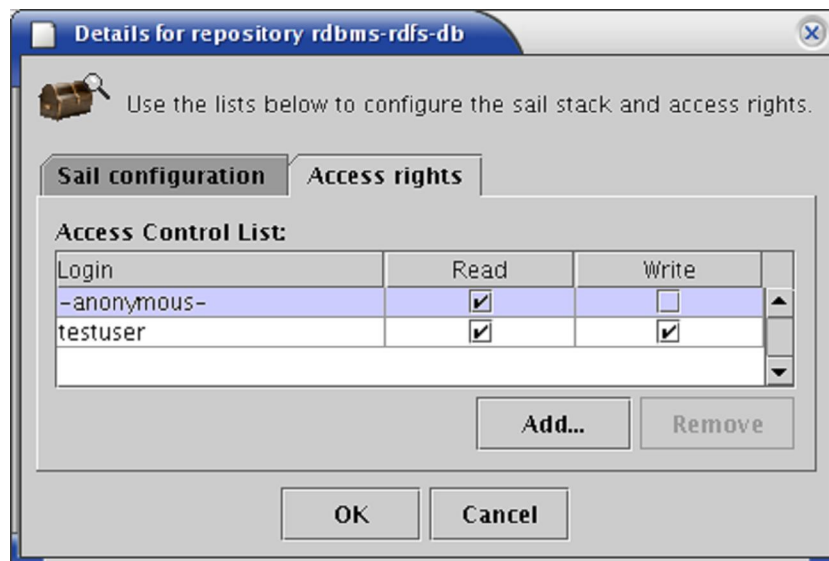


Figure 4.14 Changing the access rights

Once the Sesame is configured a Sesame entry page is shown, where the user has to put in user name and password. Generally, the Sesame entry page is located at `http://[MACHINE_NAME]:8080/Sesame/`.

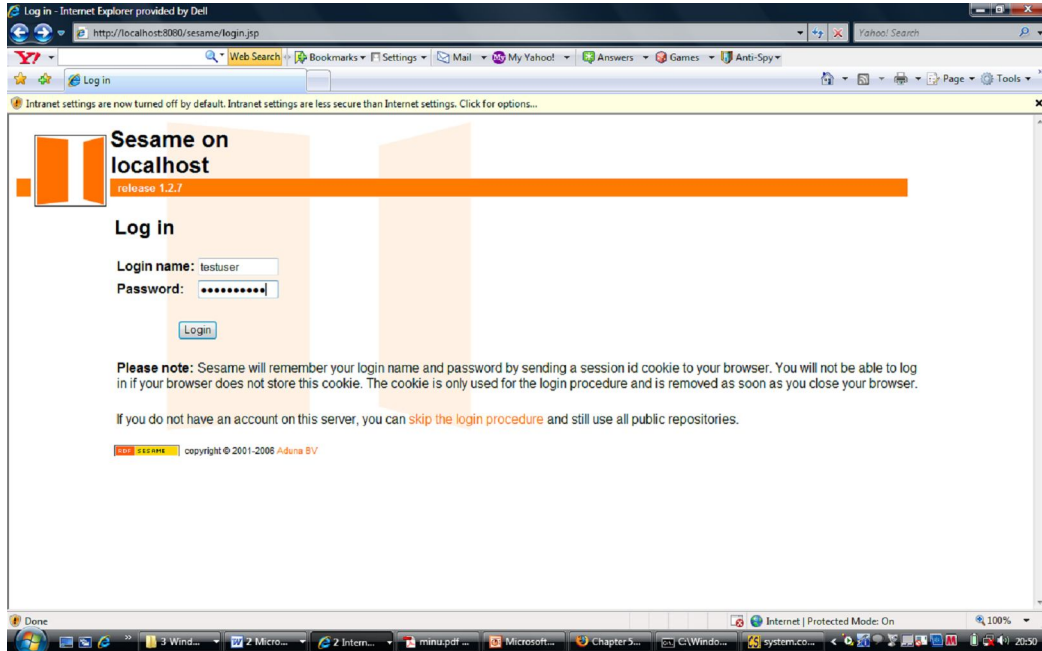


Figure 4.15 The Sesame entry page when logging in

After logging in the RDF results are added in the Sesame repositories which were obtained earlier to query it further. The query can be done in various languages namely RQL, Se-RQL, RDQL. It is up to the user which language to use.

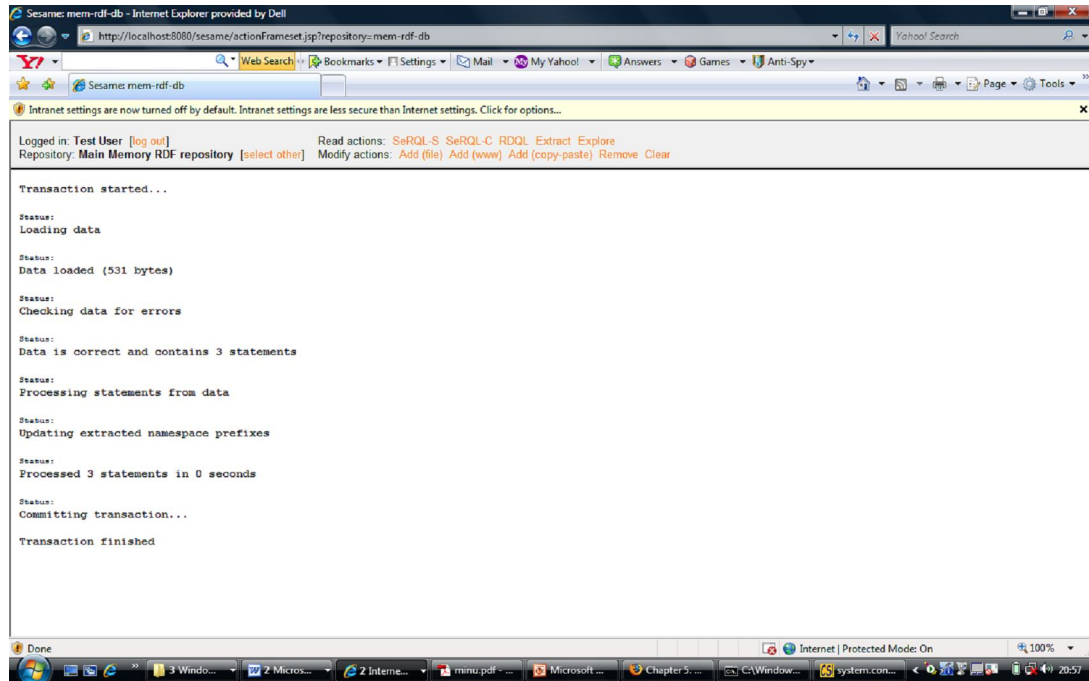


Figure 4.16 Adding data in Sesame repository

Results in various forms:

### 1 Using Se-RQL language

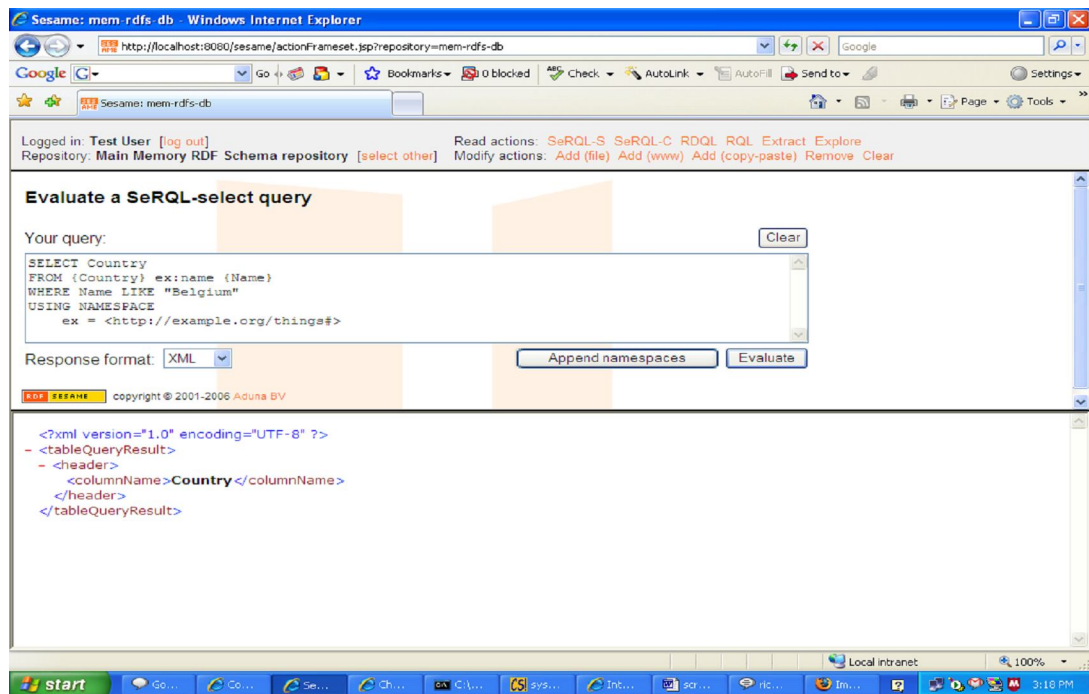


Figure 4.17 Querying the RDF using Se-RQL language

## 2. Using subject-predicate-object form

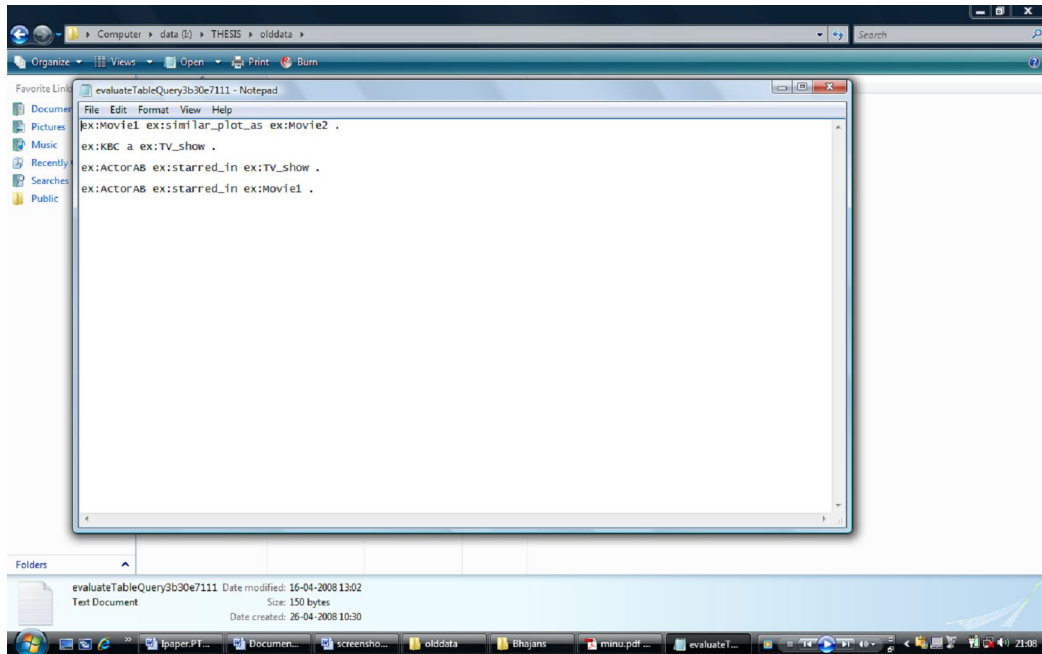


Figure 4.18 Result in the subject-predicate-object form

## 3. Results in tabular form

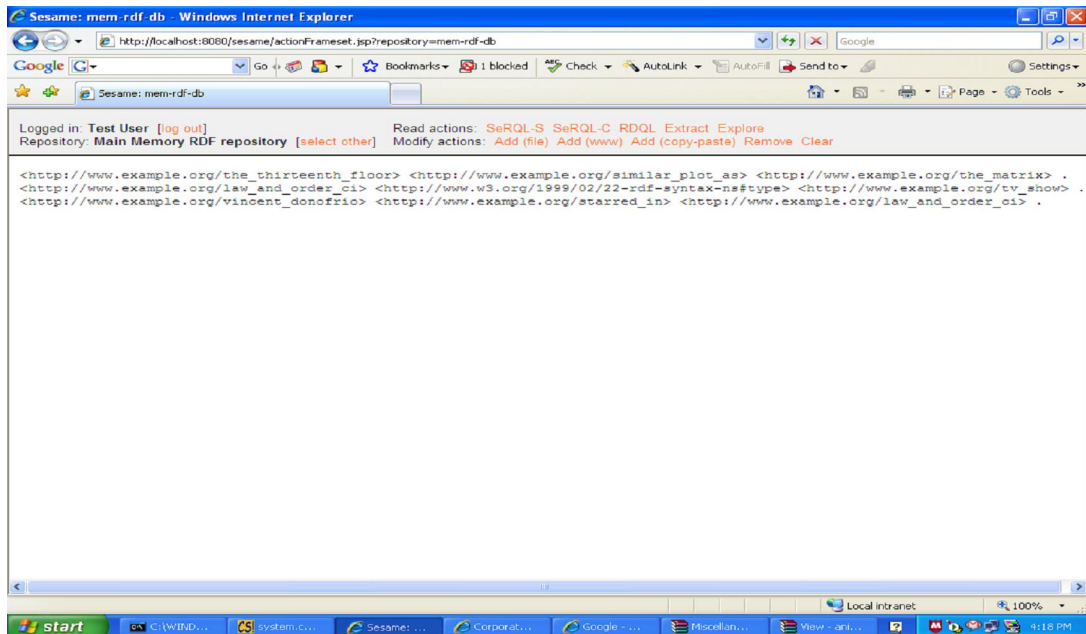


Figure 4.19 Result in tabular form

## Chapter 5

### Conclusion and Future Scope

---

RDF provides a standard model to explain facts about Web resources, which gives some interpretation to the data. However, to realize the Semantic Web vision, it will be necessary to express even more semantics of data, so further extensions are needed. UML is a good candidate for representing knowledge. Whenever we want to build a system, which can enable conversation between, various applications without the human intervention then UML can be prove beneficial approaches. It is so because of its wide acceptance, reliability and various other features described above. For systems built using other techniques, more traditional ontology and knowledge representation formalisms might be better suited. Various other models like UML can also be supported UML would be a good candidate for representing ontologies and knowledge. Whenever we want to built a system which can enable conversation between various applications without the human intervention then UML can be prove beneficial approaches. It is so because of its wide acceptance, reliability and various other features described above. For systems built using other techniques, more traditional ontology and knowledge representation formalisms might be better suited. for efficient querying in RDF models.

## References

---

- [1] T. Berners-Lee, *Weaving the Web*, Harper, San Francisco, 1999.
- [2] S. Cranfield. UML and the Semantic Web. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, 2001.  
<http://www.semanticweb.org/SWWS/program/full/paper1.pdf>.
- [3] A.J. Gerber, A. Barnard, and A.J. van der Merwe (South Africa): Towards a Semantic Web Layered Architecture From Proceeding (552) *Software Engineering - 2007*
- [4] [Decker et al., 2000] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks: *The Semantic Web: The Roles of XML and RDF*, *IEEE Internet Computing*, Vol. 4(5), pp. 63-74, Sept/Oct 2000.
- [5] <http://www.xml.com/pub/a/2001/01/24/rdf.html>
- [6] Frank van Harmelen, et al. (2001). 'Ontology-based Information Visualisation'. In *Proceedings of the workshop on Visualisation of the Semantic Web (VSW'01)*, London. in conjunction with the 5th International Conference on Information Visualisation.
- [7] N. F. Noy & D. L. McGuinness (2001). 'Ontology Development 101: A Guide to Creating your First Ontology'. Tech. rep., Stanford Knowledge Systems Laboratory.
- [8] [Nedjl et al., 2000] W. Nejdl, M. Wolpes, and C. Capelle: *The RDF schema specification revisited*. In *Modellierung 2000*.
- [9] [Dean et al., 2003] M. Dean, G. Schreiber (eds.): F. van Harmelen, J. Hendler,

I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein: OWL Web Ontology Language Reference. <http://www.w3.org/TR/2003/WD-owl-ref-20030331>.

[10] Frank van Harmelen, et al. (2001). 'Ontology-based Information Visualisation'. In Proceedings of the workshop on Visualisation of the Semantic Web (VSW'01), London. in conjunction with the 5th International Conference on Information Visualisation.

[11] Object Management Group. Unified Modeling Language (UML), <http://www.omg.org/uml/>.

[12] Booch, G., Jacobson, I., and Rumbaugh, J. (1998) The Unified Modeling Language User Guide (Addison-Wesley)

[13] K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, and M. Aronson. Extending UML to Support Ontology Engineering for the Semantic Web.

[14] [Lassila & Swick, 2003] Resource Description Framework (RDF) Model and Syntax Specification, O. Lassila and R. Swick (eds.): World Wide Web Consortium Recommendation, February, 1999. <http://www.w3.org/TR/1999/REC-rdfsyntax-19990222>.

[15] [Brickley & Guha, 2000] D. Brickley, R. Guha (eds.): Resource Description Framework (RDF) Schema Specification, W3C Candidate Recommendation, March, 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>.

[16] <http://www.openrdf.org/documentation.jsp>

[17] <http://www.openrdf.org/doc/sesame/users/ch01.html>

[18] <http://tomcat.apache.org/>

## List of Publications

---

1. Harvinder and Shalini Batra “Modelling of UML to RDF” in National conference at NCCN'08 (SLIET Longowal) on 27-28 March.
2. Harvinder and Shalini Batra “Transformation approach for UML based Semantic web” at National conference at Conference-08 (AICACCC-KM) on 19-20<sup>th</sup> Feb,08
3. Harvinder and Shalini Batra “Classification of various edge detectors” at National conference at Commune 08 on 5 April,08.