

**Deanship of Graduate Studies  
Al-Quds University**



**The Semantic Web Framework (SWF)**

**Ahmad Yahya Mohammed Nada**

**M.Sc. Thesis**

**Jerusalem – Palestine**

**1431/2010**

# **The Semantic Web Framework (SWF)**

Prepared By:

**Ahmad Yahya Mohammed Nada**

Supervisor:

**Dr. Badie Sartawi**

A thesis Submitted in Partial fulfillment of requirements for the degree  
of Master of Computer Science Department/ Master of Computer  
Science Program/ Faculty of Graduated Studies -  
Al-Quds University

1431/2010

Al-Quds University  
Deanship of Graduate Studies  
M.Sc. In Computer Science



## **Thesis Approval**

### **The Semantic Web Framework (SWF)**

Prepared By: Ahmad Yahya Mohammed Nada  
Registration No: 20714011

Supervisor: Dr. Badie Sartawi

Master thesis submitted and accepted, Date: June 5, 2010.

The names and signatures of the examining committee members are as follow:

1- Head of committee	Dr. Badie Sartawi	Signature: .....
2- Internal Examiner	Dr. Raid Zaghal	Signature: .....
3- External Examiner	Dr. Muna Matar	Signature: .....

**Jerusalem – Palestine**

**1431/2010**

## **Declaration**

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this thesis (or any part of the same) has not been submitted for higher degree to any other university or institution.

Signed: .....

Ahmad Yahya Mohammed Nada

... /... /2010

## **Acknowledgments**

After thanking Allah .....

I would like to thank my advisor, Dr. Badie Sartawi, for his guidance throughout my years at university. He has prepared me well as a researcher in computer science, prepared me in computer science academics skills, and has worked hard to promote my work .....

Al-Quds University, I would like to thank my university specially the staff of Computer Science Department for their facilities over study period .....

Special thank to my father Dr. Yahya Nada, for his encouragement, support and advice to complete the master degree .....

Thanks to my honor students who have helped me in some stages of the work, Imad Kawasmi, and Mohamad Hamayel .....

I must thank my family, who has provided love and support over all of these years. My mother and father who have helped me get to where I am now. My brothers and sisters Somaya and her husband Ayob, Mohammed my reference in technology, Abd elnaser, Samya, Zakarya and Heba .....

Finally, most importantly, I want to thank my beautiful love Mryam for her love and supportive ideas and notes. She never leaves me when I worked in the preparation of this thesis, and always offered words of encouragement .....

## ملخص

من المهم ايجاد آلية لتنظيم موارد الويب واعادة هيكلتها بالطريقة الاكثر فعالية للحصول على افضل استفادة منها، هذه الاهمية مبينة على مدى شعبية هذا العالم الرقمي ومدى الاعتماد عليه من قبل مستخدميه الذين يمارسون انشطتهم من خلاله، سواء كانوا أناسا أو وكلاء الكترونيين أذكيا.

عدة حلول تم وضعها للوصول الى هذا التنظيم المهم، ومن بينها ما قام مخترع الويب بتقديمه وهي نظرية تعنى بتحول الويب الى قاعدة بيانات مترابطة مشتركة بين جميع شبكات العالم، تحت مسمى الويب الدلالي (Semantic Web)، وقام العديد من الباحثين في هذا المجال بتقديم جهود كبيرة لتحويل هذه النظرية الى واقع ولكن حتى الآن لم يتمكنوا من وضع اطار عام لها.

هدف الرسالة ايجاد اطار جديد وموحد للويب الدلالي يدعى (SWF)، SWF عبارة عن نظام يقدم اطار لادارة محتويات الويب والمساعدة في تنظيم موارده لتحقيق الحد الاقصى من اعادة استخدامها، اطار الويب الدلالي المقترح في هذه الرسالة يحل المشاكل المتعلقة بمجالات الويب المغلفة و لغات النمذجة المعقدة و مشاكل تحقيق محتويات الويب على انها كيانات ويب ومشاكل مشاركة البيانات الوصفية في الويب.

الخدمة الاولى التي يقوم الاطار بتقديمها هي Resource Description Framework (RDF) Generation وتهدف الى تحول عملية انشاء ملفات ال RDF الى عملية اوتوماتيكية يقوم بها النظام عن طريق الوصول الى الموقع الالكتروني و استخدام آلية تحديد المجال للموقع الالكتروني لايجاد البيانات المهمة و نبذ البيانات غير المهمة. آلية تحديد المجال للموقع الالكتروني تمكن النظام من تحديد نوع الموقع و استخدام المفردات الدلالية لاستخلاص النصوص من جميع صفحاته، ومن ثم تعبئتها بنموذج ال RDF المخصص للمجال والمحضر مسبقا في النظام والذي يمكن النظر اليه على انه قالب للبيانات. لقد تم المقارنة بين العملية الآلية والعملية التقليدية في خلق ال RDF وتم اكتشاف ان العملية الآلية تستهلك ثوان من الوقت بينما العملية التقليدية تستهلك دقائق من الوقت.

خدمة لغة الاستعلام عن الويب الذي يقدمها الاطار تسهل البحث في الويب باستخدام لغة SQL معدلة. لقد قمنا بتقليل الوقت الذي يستهلكه تعلم لغة استعلام جديدة للاطار عن طريق استخدام لغة ال SQL، واختيرت هذه اللغة بالتحديد لما تتصف به من عالمية واستخدامها من قبل العديد من العمليات التجارية واستخدامها على نطاق واسع أكثر من أي لغة أخرى. الجانب المهم في هذه الخدمة هو ان المستعلم غير مضطر لتزويد النظام بالكلمات الصحيحة في البحث، وذلك لان النظام المطبق يستخدم محلل لغوي ذكي Linguistics Analyzer لفهم اجزاء جملة ال SQL، كما ويستخدم نظام التعلم الذكي Learning system ليقوم بسؤال المستعلم عن الكلمات غير المفهومة او الواضحة في الجملة.

تصنيف ملفات ال RDF هي خدمة مهمة اخرى في نظام الويب الدلالي، نظام التصنيف مهم جدا لمحركات البحث في تحسين النتائج، وذلك لانه يساعد في استرجاع النتائج المتوقع انها الاقرب لمعايير البحث المطلوبة. الاطار يطبق صيغة

تصنيف جديدة تعتمد على محتويات صفحات الموقع وعلى تاريخ ملفات ال RDF الخاصة بالموقع، وتستخدم الصيغة الجديدة قيمة التصنيف القديم مضافا اليها قيمة معقولة وذلك لاحتواء التصنيف الجديد ضمن مدى محدد.

ينفذ نظام ال SWF ميزات ذكية مثل المحلل اللغوي الذكي الذي يقوم بفهم جمل البحث، ونظام التعلم الذكي الذي يطلب مساعدة المستعلم عندما لا يستطيع النظام تحديد بعض البيانات المهمة اثناء تنفيذ الخدمات التي تم ذكرها، والحالات التي يكلب فيه النظام المساعدة تضمن: المساعدة في تحديد مجال الموقع الالكتروني اثناء عملية اشاء ملفات ال RDF، والمساعدة في تحديد مرادفات كلمات جمل البحث عند عدم القدرة على ايجادها في قاعدة المعرفة الخاصة بالنظام، والمساعدة في توضيح الكلمات الغامضة في جمل البحث.

يقدم نظام ال SWF ميزة جديدة هي تلخيص مواقع الويب، الفكرة من هذه الخدمة هي تلخيص ملف ال RDF المنشأ للموقع وتقديم فقرة مختصرة وقصيرة عن اجزاء الموقع المهمة.

يمكن لنظام ال SWF ان يغطي كامل الويب، لكن في هذه الرسالة قمنا بالتركيز على مجال التعليم وجعله نطاق التنفيذ لدينا، وقمنا بتطوير تطبيق على شبكة الانترنت باستخدام Microsoft C#.Net 2008 و Microsoft SQL Server 2008.

## **Abstract**

It is important to find a mechanism that organizes and restructures the web resources in the most effective way to get the best benefit from it; this importance comes from the dependability and popularity of such digitalized world where human and intelligent agents practice their activities.

Many solutions were put to resolve this organization, one of the most valuable solutions was suggested by the WWW creator Tim Berners-Lee, and his vision is to convert the entire web into integrated database across the world's networks, under the name of "Semantic Web". Researchers who adopted this vision actively participated to make this dream true, but until now no standardized framework is globalized.

The purpose of this thesis was to find a new and standard Semantic Web Framework (SWF), the SWF is a system that provides a framework to manage the web contents and help in organizing its resources to achieve the maximum reusability of them; many researchers had worked on this field but it is agreed that no global framework is set yet. The suggested SWF covers the problems areas of closed domains, complexity of modeling languages, realizing web entities and Metadata sharing.

The Resource Description Framework (RDF) Generation in the main service in the SWF, aims to automate the tradition RDF creation process by accessing the website pages and uses domain-oriented approach to find relevant contents and eliminate unwanted data, the domain-oriented approach enables the system to detect the website type and use semantics to extract the text from all the website's pages, then fill the suitable prepared RDF model which can be considered as RDF document template. The automated and the manual generation of the RDF documents have been compared and discovered that the automated process consumes time in seconds but the traditional process consumes time in minutes.

The Web Query language is a service that is provided by this system facilitates the web searching using modified SQL. We had reduced the time to learn new framework query by using the SQL language; it was chosen because it is standardized, used by many business processes and widely used more than any other language, the important aspect of this service



that the SQL requestor do not have to provide the correct words to the system; because the applied SWF uses Intelligent Linguistics Analyzer system to understand the statement parts, also it uses intelligent Learning system to ask the requestor for help about the parts of the statement which it couldn't understand.

RDF Ranking System is another significant service in the SWF, the ranking is important for the search engines to optimize the results; because it helps in returning the results that are predicted to be the closest ones to the search criteria, the SWF applies new updating rank formula that depends on the website pages contents and on the RDF documents history, the formula includes the old rank added to it a reasonable value to limits the ranks in a managed range.

The SWF carries out intelligent features such as the Linguistics Analyzer that understand the users' SQL statements, and the Learning System that learns from the requester when the SWF system cannot determine some of core data during the services execution, the cases when the Learning system asks for help includes: asking about the domain in the RDF generating, asking for keywords synonyms when it does not find them in the knowledgebase through the SQL execution, asking for the unclear keywords meanings when executing the SQL statement.

The SWF brings out a new feature of Auto-Web Summarizing; the idea of this feature is to summarize the generated RDF document and provides a brief and short paragraph of the website main parts.

The SWF can cover the entire web, but in this thesis we took the educational domain as the implementation scope using Microsoft C#.Net 2008 and Microsoft SQL Server 2008 to develop a web-based application.

## Table of Contents

Declaration. . . . .	i
Acknowledgment. . . . .	ii
Abstract in Arabic. . . . .	iii
Abstract. . . . .	v
Table of Contents. . . . .	vii
List of Figures. . . . .	x
List of Tables. . . . .	xi
List of Appendices. . . . .	xii
Chapter 1 Introduction. . . . .	1
1.1 Introduction. . . . .	1
1.2 Problems and Motivation. . . . .	2
1.3 Contributions to the Field. . . . .	3
1.4 Outline. . . . .	4
Chapter 2 Background. . . . .	5
2.1 The Web. . . . .	5
2.2 The Semantic web. . . . .	6
2.3 The RDF. . . . .	8
2.4 SPARQL Query Language for RDF. . . . .	9
2.5 Web Ranking. . . . .	10
2.6 Related Work. . . . .	13
2.6.1 Related Projects. . . . .	13
2.6.2 Comparison between SWF and Other Frameworks. . . . .	15
2.6.3 Our Work. . . . .	16

Chapter 3	Semantic Web Framework Model. . . . .	17
3.1	Introduction. . . . .	17
3.2	SWF Overall Approach and Architecture. . . . .	17
3.3	The RDF Generator Model. . . . .	20
3.3.1	Introduction. . . . .	20
3.3.2	Websites type detection. . . . .	21
3.3.3	Websites classification. . . . .	21
3.3.4	RDF Models. . . . .	22
3.3.5	Semantic Extraction. . . . .	23
3.3.6	SWF Learning. . . . .	25
3.4	The RDF Query Language for SWF. . . . .	26
3.5	Ranking System for SWF. . . . .	28
3.6	Discussion. . . . .	29
Chapter 4	SWF System Design. . . . .	31
4.1	Introduction. . . . .	31
4.2	Overall Description. . . . .	31
4.3	Functional Requirements. . . . .	33
4.4	External Interface Requirements. . . . .	34
4.5	Non-Functional Requirements. . . . .	35
4.6	The System Design. . . . .	36
4.6.1	SWF Flowcharts. . . . .	36
4.6.2	SWF Use Case Diagram. . . . .	40
4.6.3	Narrative Use Cases. . . . .	41
4.6.4	Database Entity Relationship Diagram. . . . .	60
4.6.5	Database Entities Specifications. . . . .	60

Chapter 5	Results and Evaluation. . . . .	64
5.1	Results. . . . .	64
5.1	Evaluation. . . . .	65
Chapter 6	Conclusion and Future Works. . . . .	74
6.1	Conclusion. . . . .	74
6.2	Recommendations and Future Works. . . . .	75
	References. . . . .	76
	Appendices. . . . .	81
	Appendix 1: An Example of RDF Document for Alquds.edu. . . . .	81
	Appendix 2: W3C Document License. . . . .	86
	Appendix 3: SWF User Manual. . . . .	88
	Appendix 4: Publications. . . . .	98

## List of Figures

Figure 2.1	Semantic Web Layer Cake. . . . .	7
Figure 2.2	Basic form of a semantic triple. . . . .	8
Figure 2.3	RDF triples. . . . .	9
Figure 2.4	Sample RDF statement. . . . .	9
Figure 2.5	Hits Algorithm Authority and Hub. . . . .	10
Figure 2.6	Hits Update Roles. . . . .	11
Figure 3.1	SWF architecture. . . . .	18
Figure 3.2	The current status of the Web. . . . .	19
Figure 3.3	The Web after using the Semantic Web Framework. . . . .	19
Figure 3.4	RDF Generator model. . . . .	20
Figure 3.5	Update rank algorithm. . . . .	29
Figure 3.6	The SWF System. . . . .	29
Figure 4.1	RDF Flowchart. . . . .	36
Figure 4.2	Query Flowchart. . . . .	37
Figure 4.3	Web summary Flowchart. . . . .	38
Figure 4.4	Update rank. . . . .	39
Figure 4.5	SWF Use Case Diagram. . . . .	40
Figure 4.6	SWF ERD. . . . .	60

## List of Tables

Table 2.1	Comparison Between SWF and the Other Frameworks. . . . .	15
Table 4.1	User Classes and characteristics. . . . .	32
Table 4.2	RDFs Table. . . . .	60
Table 4.3	Domains Table. . . . .	61
Table 4.4	Properties Table. . . . .	61
Table 4.5	RDF_Properties Table. . . . .	62
Table 4.6	Domain_ Properties Table. . . . .	62
Table 4.7	KeyTypes Table. . . . .	63
Table 4.8	Knowledge base Table. . . . .	63
Table 5.1	Evaluation of Creating RDF Manually and Automatically. . . . .	65
Table 5.2	The distribution of the study sample according to the Academic Rank. . . .	67
Table 5.3	The distribution of the study sample according to the Number of Publications	67
Table 5.4	The study tool depending on the dimensions of the study. . . . .	67
Table 5.5	Balance of the percentages of responses. . . . .	69
Table 5.6	Averages and percentages of the first pivot. . . . .	69
Table 5.7	Averages and percentages of the second pivot. . . . .	70
Table 5.8	Averages and percentages of the third pivot. . . . .	71
Table 5.9	The results of (ANOVA) test to academic rank variable. . . . .	71
Table 5.10	The results of (ANOVA) test to Number of Publications variable. . . . .	71

## List of Appendices

Appendix 1	An Example of RDF Document for Alquds.edu. . . . .	81
Appendix 2	W3C Document License. . . . .	86
Appendix 3	User Manual. . . . .	88
Appendix 4	Publications. . . . .	98

# Chapter 1

## Introduction

### 1.1 Introduction

The web is a large and growing source of information. This information is not organized, therefore it difficult to reuse. Because reusing the web resources cannot be achieved, the need for a new technology to convert the huge amount of unorganized text into an organized database urgently emerged; Berenens-Lee, the World Wide Web creator, introduced a new vision to bypass what is called the “Semantic Web”. The Semantic Web has promised solutions to challenges of huge data integration, directed search, and knowledge management (Feigenbaum , and et al., 2007), because of this, the researchers and programmers have developed and built tools to realize that vision; until now the researchers face some difficulties and problems in the application of the semantic web. This thesis introduces a framework to facilitate the Semantic Web use, make it applicable, generalized, and standardized.

The Semantic Web is a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation of a large number of researchers and industrial partners. It is built on data that is represented in the Resource Description Framework (RDF). The RDF is a framework for representing the information on the Web. RDF has a simple data model that is easy for applications to process and manipulate (W3C, 2009).

Basically, the goal of the Semantic Web is to enable the machine to understand the web resources. Its vision involves the production and use of large amounts of RDF data; we can see it as a large knowledge base formed by sources that serve information as RDF files (Bojars, and et al., 2008), whereas RDF enables the description of anything. The Semantic Web has inspired a large community of researchers and practitioners who have achieved several early successes in the past years. Through these years the researches have introduced and developed tools and technologies to make the Semantic Web applicable.



This thesis is an attempt to make the Berners-Lee vision applicable by gathering researchers contributions in the Semantic Web field and introducing them in one framework. The proposed framework contains three parts; the first is RDF generator to generate a standard description (RDF) for web sites using predefined model for each web site type. The second part is a query system to retrieve the information from the RDF documents. The third part is an adaptive page ranking Algorithm to Semantic Web for using it in the search engines. These parts are the Semantic Web Framework (SWF) which is presented in this thesis. SWF implemented using web service to generate standard RDF documents for websites, and platform to retrieve the information from the RDF documents, using special query language to use it by users or search engines.

## **1.2 Problems and Motivation**

The Web is increasingly becoming a social place. There has been a shift from just existing on the web to participate on the web Community applications such as collaborative wikis which has become a very popular domain. This evolution is a new challenge to Semantic Web technologies. That required more intelligent tools and technologies to face this evolution in the web applications.

The Semantic Web is founded to create new facilities on top of the current Web in order to make the knowledge accessible and reusable. This attempt is still in its formative years and is driven by computer researchers and web programmers. The Semantic Web researchers and programmers assumed some hypotheses before doing Semantic Web researches or projects. This research is an attempt to make some of these hypotheses and problems possible to be solved and applied. The research problems are:

- The Semantic Web applications have assumed closed domains of manageable sizes; Berners-Lee introduces the Semantic Web vision to be a new version of the current humongous web. So we need to make the semantic web more general and standard.
- The complexity of the modeling languages, which is based on RDF documents, needs trained Knowledge engineers for modeling. But the Semantic Web researchers don't have time to build these models. That is why we need to provide standards and prepared models to facilitate the researchers work.

- The Semantic Web researcher imagines the web as entities to use it as database in creating, sharing and reusing on the Web. Advanced technology is needed to make this imagination applicable.

In this research, we aim to solve these problems by introducing a framework to generate RDF documents from Web pages, process the requester's queries, and introduce ranking for RDF documents to use it in search engines. The framework contains prepared models to analyze the Web pages, and create RDF files. Using this framework enabled us to avoid the closed domains problem, the complexity of modeling so that the programmers don't need to be trained in modeling languages. We realize the entity of creating, sharing and reusing the Metadata on the Web.

### **1.3 Contributions to the Field**

In this thesis, we contribute the following in the Semantic Web filed:

- The Semantic Web Framework (SWF) is a technique to make the semantic web applicable. The SWF has an RDF generator that generates RDF documents from any web page using predefined model for each web domain, query processor to retrieve the requested data from RDF document. The framework also has page ranking system to be used in the search engines.
- The RDF generator is a new idea to help in preparing RDF documents; it is a platform to generate a standard description (RDF) for internet web sites using predefined model for each web site type. The RDF generator classifies the web sites to help us while detecting the type of the web site and extract the description from the pages. The RDF generator depends on other sub-systems like the Semantics Matching System which is the basic intelligent method used to determine the type of webpage then to extract the needed information to generate the RDF.

## **1.4 Outline**

This Thesis is organized as follows. Chapter 2 provides the needed background about the Semantic Web and the related works. Chapter 3 presents a description of the SWF model in details. Detailed design of the model is presented in Chapter 4. Chapter 5 provides the implementation results and evaluation, conclusion and future works are presented in Chapter 6.

## **Chapter 2**

### **Background**

#### **2.1 The Web**

The World Wide Web (WWW) is a huge amount of documents. The content of these documents is usually natural language text whose semantics is mostly inaccessible to machines. This text is typically marked up using languages like Hypertext Markup Language (HTML). The markup basically describes the structure of the document. That is, it describes which part of the document is a heading, a paragraph, a hyperlink and so on. Web browsers can use this information to display the documents appropriately to humans. Therefore, the primary purpose of the markup is to explain the logical structure of a document and to leave the presentation details to the browser (Elliott, 2008).

HTML markup helps the machines to recognize different components of a document. But it does not help the machines to understand the content of these components and hence the document. In other words, HTML does not make the semantics or meaning of the document content accessible to the machines. In this sense, documents marked up by HTML are not different from those that are not marked up. We can refer to such documents as natural language documents or free-text documents or unstructured data.

The WWW has revolutionized the world. It is undoubtedly successful and has showed growth in the recent past. The underlying technology is simple enough to let the user to create and add documents to this repository. This has resulted in an explosion of information. As a result, finding and using relevant, precise, timely and reliable information on the Web is becoming more and more difficult. Search engines like Google help the users search for documents on the Web using keyword based queries. The problem with the search engines is that they retrieve too many documents. Also, search engines are not content aggregators. If the information that the user is looking for is distributed over several pages, then the user has to extract pieces of information from different documents and put them together.

## 2.2 The Semantic Web

*“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”*

– Tim Berners-Lee, James Hendler and Ora Lassila (Lopez, and et al., 2007)

The traditional Web technologies, which are based on a large amount of information, do not provide the semantic search and navigation in unorganized information. So the researchers are trying to find new technologies to navigate and search in the Web using semantic words to make these processes more intelligent. These efforts and technologies named The Semantic Web.

The Semantic Web initiative that led by World Wide Web Consortium (W3C) aims to make the content of the web more accessible to machines. The basic underlying data model of the Semantic Web is the RDF.

Because of the information exchange and collaboration is difficult aspect of making the Semantic Web a reality, a joint collaborative effort between multiple groups working in parallel is necessary. Various working groups can be established to work in parallel, allowing definitions to be produced more efficiently and a general consensus can be reached for each layer much more easily than if everyone had to swallow the whole cake at once (Antoniou, and et al., 2008). Figure (2.1) illustrates the “Semantic Layer Cake” as defined by the WWW Consortium (W3C) (W3C, 2007). Each layer is built upon the previous layer below it. At each layer the Semantic Web community is working to define and refine open standards such as XML, RDF, OWL, and SPARQL. Each of these open standards contributes in its own way to enable the communications necessary to realize the Semantic Web. At higher layers of the cake, logic and agent applications provide utilities to the end user.

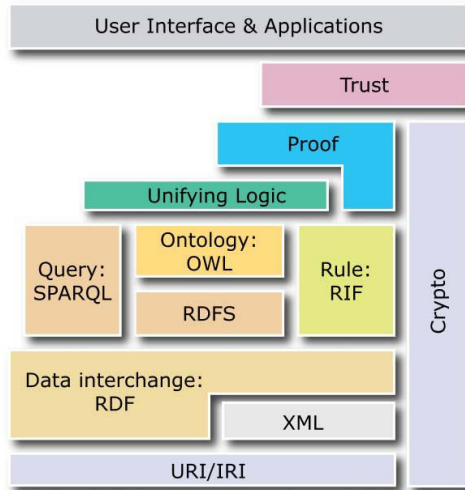


Figure 2.1: Semantic Web Layer Cake (W3C, 2007)

The Semantic Web layered cake presented as each layer is built on top of the other. RDF model is built upon the idea of URIs and it has XML syntax. RDF Schema and OWL are ontology languages which themselves are described using RDF. These languages map to well-understood formal models. Automated agents can use rules and ontological information to reason with the data and infer new information. The proof layer represents the fact that an agent must be able to justify how it has arrived at a given fact by keeping track of reasoning traces. It specifies proof representation and proof validation. The Semantic Web is like a huge database of facts. But the nature of the web is such that anybody can add new facts to this database. It is important for automated agents to be able to determine whether a fact is to be trusted or not (Immaneni, 2007).

XML (Extensible Markup Language) is a set of rules for encoding documents electronically it is a flexible markup language. It is a simple text format that was derived from ISO 8879. It originally achieved W3C recommendation status in 1998 (W3C, 1998).

RDF is a data model that enables one to convey semantic meaning alongside information in a standard fashion to software applications and services. RDF supports an XML-based syntax, allowing it to build off of the XML standardization efforts. The key elements of RDF are resources, properties, and statements which associate specific properties with resources. When coupled with an RDF, the model is able to convey semantic connotation in a machine comprehensible format (Aaron, 2002).

OWL, in recommendation status since 2004, enables a higher degree of machine understanding of Web content, building on XML, RDF. RDF is intentionally limited. Ontology languages are able to build off of them to suit their domain specific needs. OWL is intended by the W3C to eventually be a global ontology for the Semantic Web (W3C, 2004).

SPARQL has emerged as the standard RDF query language. It achieved recommendation status in January 2008 (W3C, 2008). Information contained in a RDF built upon XML syntax can be queried via XML query tools.

Software Agents are instances of programs with AI properties. They are able to work independently and proactively to solve problems. They use logic to apply principles of reason to solve problems based on input and pre-defined criteria or preferences and provide interfaces for human interaction. Software agents will play a major role in the Semantic Web. While they will not replace humans, they will augment human capability by supporting users with many of services (Thomas, 2008). Open standards must be used by technologies to achieve an infrastructure of applications and utilities that will ultimately enable the Semantic Web and beyond.

### 2.3 The RDF

RDF is the foundation Semantic Web data standard. RDF is a flexible and expressive graph-based data model whose properties make it a compelling choice for modeling data with rapidly changing structures, integrating data from multiple sources, and accessing data in a powerful and uniform manner. One of the primary strengths of RDF is that it can be used to represent nearly any kind of information.

The RDF defines an abstract syntax for making assertions (or statements) about resources in a subject-predicate-object form. These statements are also known as *RDF triples*.

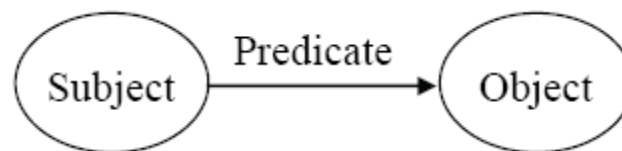


Figure 2.2 Basic form of a semantic triple

As seen in the Figure (2.2) above, this abstract syntax is most often visualized as a directed graph, where the nodes in the graph are the resources and the links between them are the statements. Meaning is captured by the combination of all the statements in an RDF graph.

The nodes in an RDF graph are one of three kinds: a URI, an RDF literal, or a *blank node*. An RDF literal is a scalar value associated with the combination of the subject and predicate. A blank node is “just a unique node that can be used in one or more RDF statements, but has no intrinsic name” (Klyne, and et al., 2004). For example, in RDF a statement about the W3C’s Architecture of the World Wide Web recommendation can be asserted in RDF in the way in Figure (2.3):

```
http://www.w3.org/TR/webarch/ (subject)  
http://www.w3.org/1999/02/22-rdf-syntax-ns#type (predicate)  
http://xmlns.com/foaf/0.1/Document (object)
```

Figure 2.3 RDF triples

The same statement can be visualized using the RDF abstract syntax, which represents each of these statements as links in an RDF graph. The URIs used for the predicate and object terms are shown in Figure (2.4) an abbreviated form for the sake of brevity:

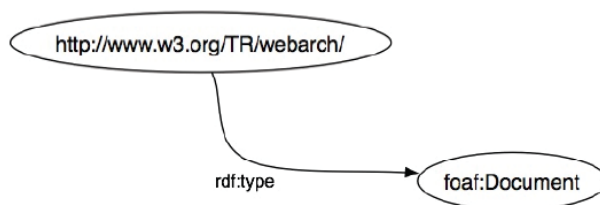


Figure 2.4 Sample RDF statement

## 2.4 SPARQL Query Language for RDF

SPARQL is the standard query language of the Semantic Web (W3C, 2008). It is designed from the ground up to be capable of simultaneously querying and integrating the results from multiple data sources scattered across the web, SPARQL allows clients to choose exactly the data that they want, giving them great flexibility for data access. Traditional data access APIs are limited and return proprietary structures, if they exist at all, trapping data in a single place or making it cumbersome to retrieve (Feigenbaum, and et al., 2007).



SPARQL is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. Prior to the development of the SPARQL specification, there was no formal, publicly-standardized language for querying RDF data, although several others had been proposed. In addition, there was no standardized protocol for interacting with remote or local RDF storage servers (Kendall, 2005). SPARQL was designed to address this shortcoming.

## 2.5 Web Ranking

With the rapid growth of the Web, providing relevant pages of the highest quality to the users based on their queries becomes increasingly difficult. The reasons are that some web pages are not self-descriptive and that some links exist purely for navigational purposes. Therefore, finding appropriate pages through a search engine that relies on web contents or makes use of hyperlink information is very difficult.

To solve the problems mentioned above, the Search Engines display search results in ascending order, starting with the best results to worst results. The Search Engines use ranking for each page by use algorithms for ranking the results. There are some of prepared ranking algorithms to sort the search engines results. The most two popular algorithms for ranking which are Hyperlink-Induced Topic Search (HITS) and Google PageRank.

### The HITS Algorithm:

This algorithm used for rating a set of search engine results, which it determine two values for each page in the set of results the Authority and Hub. The Authority of page ( $Auth(p)$ ) is a page has in links, while a hub of page ( $Hub(p)$ ) has out links, See Figure (2.5).

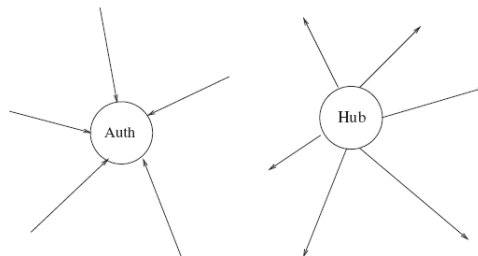


Figure 2.5: Hits Algorithm Authority and Hub.

Authorities and hubs exhibit a mutually reinforcing relationship: A good hub page is one that points to many good authorities; a good authority page is one that is pointed to by many good

hub pages. Page p has both an authority value  $Auth(p)$  and a hub value  $Hub(p)$ , the  $Auth(p)$  is the sum of scaled hub values that point to page p, while the  $Hub(p)$  the sum of scaled Authority values of the pages it points to.

The Hits Algorithm in the begin set  $Auth(p)=1$  and  $Hub(p)=1$  for each page. Then do two updates on all pages values recursively; the first one Authority Update Role to update each page's, Authority value to be equal the sum of the Hub values of each node that points to it. And the second one is Hub Update Role is to update each page's, Hub value to be equal to the sum of the Authority values of each node that it points to, see Figure (2.6).

Authority Update Role for all n pages : 
$$\sum_{i=1}^n hub(p)$$

Hub Update Role for all n pages : 
$$\sum_{i=1}^n auth(p)$$

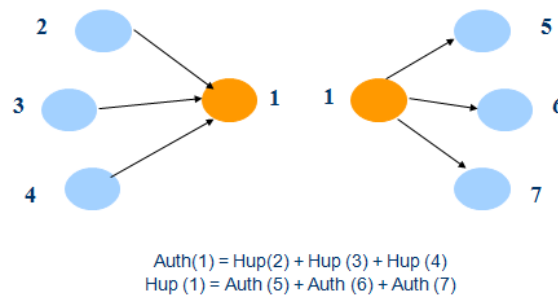


Figure 2.6: Hits Update Roles

The algorithm, in general, do the following: for all s pages set  $Auth(p)=1$  &  $Hub(p)=1$  after that for all s pages apply Authority Update Rule and apply Hub Update Rule. [3] present an algorithm to do that and the following is briefly and clear algorithm:

S: set of pages  
 N: the number of pages in S  
 For all pages in S do  
      $Auth(p) = 1$   
      $Hub(p) = 1$   
     For  $i = 1, 2, 3, \dots, n$   
         Apply the Authority update role  
         Apply the Hub update role

### The PageRank Algorithm:

The PageRank Algorithm is another method for rating the pages, which used by Google. This algorithm used numerical weighting for each page in linked pages set. The PageRank Algorithm depends on incoming links; the page that linked from many pages has high rank, that mean this algorithm just use the authority. The PageRank of a page is a recursive definition, so computation necessarily requires iteration.

The PageRank of page (PR(p)) is dependent on the PageRank values for each page (PR(p')) out of the linked set, divided by the number of links from page p'.

$$PR(p) = \sum_{p' \in B_p} \frac{PR(p')}{L(p')}$$

B<sub>p</sub>: the set of all pages linking to p. The page algorithm holds a damping factor 'd' and can be set between 0 and 1. So we can calculate th PR(p) using the following equation:

$$PR(p_i) = d \sum_{p' \in B_p} \frac{PR(p')}{L(p')} + 1 - d$$

Now the algorithm, in general, do the following: for all S pages set PR(p)= 1/N, after that for all s pages apply the recursively calculating of the last equation, and the following is briefly and algorithm:

S: = set of pages  
N: =the number of pages in S  
For all pages in S do  
    PR(p) = 1/N  
For i: =1, 2, 3, ..., N  
    PR(pi) = (1-d) + d (PR(p'1)/L(p1) + ... + PR(p'N)/C(p'N))

### Our Ranking Algorithm:

In this research we develop a new adaptive ranking algorithm for RDF documents, it depends on the related content in the page and the RDF document history, in addition to the above algorithms all links equally when distributing rank scores but our algorithm has been developed to improve the performance of ranking by adding new parameter to give the weight of each result on the basis of content.

## 2.6 Related Work

Our research introduces a framework for Semantic Web. There have been many attempts to introduce frameworks for Semantic Web. This section gives an overview of some of the existing related projects followed by an overview about our project

### 2.6.1 Related Projects

Hewlett-Packard (Hewlett-Packard, 2009) (Carroll, and et al., 2004) proposed the Jena Framework which is a Java framework for querying RDF data. RDF data can be stored either in a database or in-memory. The components of the framework include APIs for processing RDF (reading, writing, navigating the graph, querying etc.), and support for querying. The first version of Jena (Jena1) used two different database schemas- one for relational databases and a special one for Berkeley DB (BDB). The schema for relational databases consisted of a statement table, a resources table, and a literals table. Every resource and literal was given a unique ID. Resources were stored in the resources table (ID, Value) and the literals were stored in literals table (ID, Value).

Statements were stored in the statement table (subject, predicate, ObjectURI and ObjectLiteral). To distinguish object literals from object URIs, two columns were used for the object of the statement. As is obvious, every find operation would require multiple joins between the statement table and the literals table. The schema for BDB stored all parts of the statement in a single row. But each statement was stored three times indexed by subject, predicate and object respectively, thereby increasing storage cost. It was found that data was accessed much faster from BDB suggesting that denormalized relational schema might reduce response times.

Redland (Beckett, 2005) (Beckett, 2001) is another RDF application framework like Jena that allows RDF data to be stored, queried and manipulated in C, Perl, Python, TCL and other languages. It provides APIs for manipulating and querying RDF. RDF graphs can be stored in-memory or in a database (BDB, MySQL etc.) It provides support for querying using RDQL and SPARQL. Redland stores RDF statements (in-memory or persistent storage) using hashes. A hash is the mapping from a key to a value with duplicates allowed. Statements are stored

using three hashes corresponding to the following keyvalue pairs: (Subject&Predicate, Object), (Predicate&Object, Subject) and (Subject&Object, Predicate).

Oracle 11g (Oracle, 2009) introduces scalable, secure and reliable RDF management platform. Based on a graph data model, RDF triples are persisted, indexed and queried, similar to other object-relational data types. The Oracle 11g RDF database ensures that application developers benefit from the scalability of Oracle 11g to deploy scalable and secure semantic applications.

Sesame (Immaneni , 2007) (Sesame, 2009) is a Java framework for storing and querying RDF. It can be deployed on relational databases, in-memory, file systems and supports both local and remote access. It supports querying using several different query languages including SeRQL and SPARQL. Like other systems, it provides APIs to manipulate RDF graphs.

KAON (KAON, 2005) (Volz, and et al., 2003) is a Java based suite of tools that focuses on scalability and efficient reasoning with ontologies. The ontology language that is supported is based on RDFS with proprietary extensions. The infrastructure targets business applications and gives importance to security, consistency, extensibility etc. API for manipulating RDF with support for both in-memory and persistent storage (any SQL2 compatible database) is provided.

rdfDB (Guha, 2007) is a database for RDF. It supports C and Perl access. The database can be queried using a high level SQLish query language provided. It supports RDFS along with some basic inferencing. The system uses three Sleepycat BTree databases to index each statement using the following key value pairs (Subject&Predicate, Object), (Object&Predicate, Subject) and (Predicate, Subject).

RDFStore (RDFStore, 2006) is a Perl/C toolkit to process RDF. Data can be stored and retrieved from in-memory hashed storage or from the local disk using BDB. Querying can be done using RDQL query language. The programming API supports contexts, reification, multi-lingual literals, etc.

Kowari (Adams, and et al., 2005) is a transaction-safe storage infrastructure for RDF data. Kowari supports iTQL and RDQL query languages. Kowari stores RDF in its native transactional data store. The system is designed such that other data stores can be plugged into

it. The native data store consists of a Statement Store and a String Pool. Statements are stored as a quad of nodes consisting of the subject, predicate, object and meta nodes. Each node is assigned 64-bit integers and statements are therefore stored as a quad of integers. The URIs and literals are stored in the String Pool.

## 2.6.2 Comparison between SWF and Other Frameworks

The previous projects are an overview of some exiting frameworks for Semantic Web. The main purpose for these projects is to store the RDF document in relational databases and retrieve the requested data using the implemented query languages for RDF and databases. In our research will use the features in the previous projects and add new features like system to generate the RDF documents.

We develop a new platform which is not found in the previous frameworks; it is a mechanism for creating RDF files automatically instead of using prepared files, and this is a quantum leap in this area compared to what preceded it. In addition to develop a new adaptive ranking algorithm for RDF documents to improve the performance of ranking.

The following table illustrates the main aspects of comparison between SWF and the other frameworks.

Table 2.1: Comparison Between SWF and the Other Frameworks

	Generate RDF	RDF Query	Ranking System
Jena Framework	X	√	√ (in 2009)
Redland	X	√	X
Oracle 10g	X	√	X
Sesame	X	√	X
KAON	X	√	X
rdfDB	X	√	X
RDFStore	X	√	X
Kowari	X	√	X
SWF	√	√	√

### **2.6.3 Our Work**

The SWF is a system for creating, querying, and sorting the RDF document. The SWF introduces a new platform for generating the RDF document using predefined model for each website type. It builds the RDF documents automatically by determines the website type using special knowledge base, then extracting the related semantics to prepare the RDF document. This platform used by users or other websites like search engines.

SWF uses modified SQL for querying the RDF documents, it retrieves the requested data from the RDF documents after modifying the requested SQL using linguistic analysis to retrieve the best result. It is used to request RDF document, to request data from any RDF document, and to refresh RDF document. We use the standard SQL because it is used widely between researchers and programmers without having to learn another query language.

SWF can sorts the results using a new ranking system for RDF by adding a parameter in each RDF document. The SWF ranking system used a new formula to calculate the ranking for each RDF document; it recalculates the ranking for the documents automatically after each searching process. SWF ranking system formula depends on the related content in the page and the RDF document history to improve the performance of ranking.

The SWF framework was implemented as a web service. We have divided the implementation into three parts; databases, control panel, and users web service. Our database was implemented using Microsoft SQL Server 2008 to store the knowledge base and the RDF documents, and the web applications were implemented using Microsoft C#.Net 2008.

## **Chapter 3**

### **The Semantic Web Framework Model**

#### **3.1 Introduction**

The SWF is an innovation that provides a platform to put the Semantic Web theories into practice and gather the related researchers' works into one and unified outcome. As mentioned before, the SWF is made up of three main parts; the RDF Generator for generating RDF documents from any web page using predefined model for each web domain, query processor to retrieve the requested data from RDF document, and page ranking system to provide it to the search engines, in this chapter we introduce the SWF content and functions in details.

#### **3.2 SWF Overall Approach and Architecture**

The SWF is a system developed to automatically prepare RDF documents from websites and store these documents in the database to retrieve them when required. Figure (3.1) shows the SWF functions and its main components.



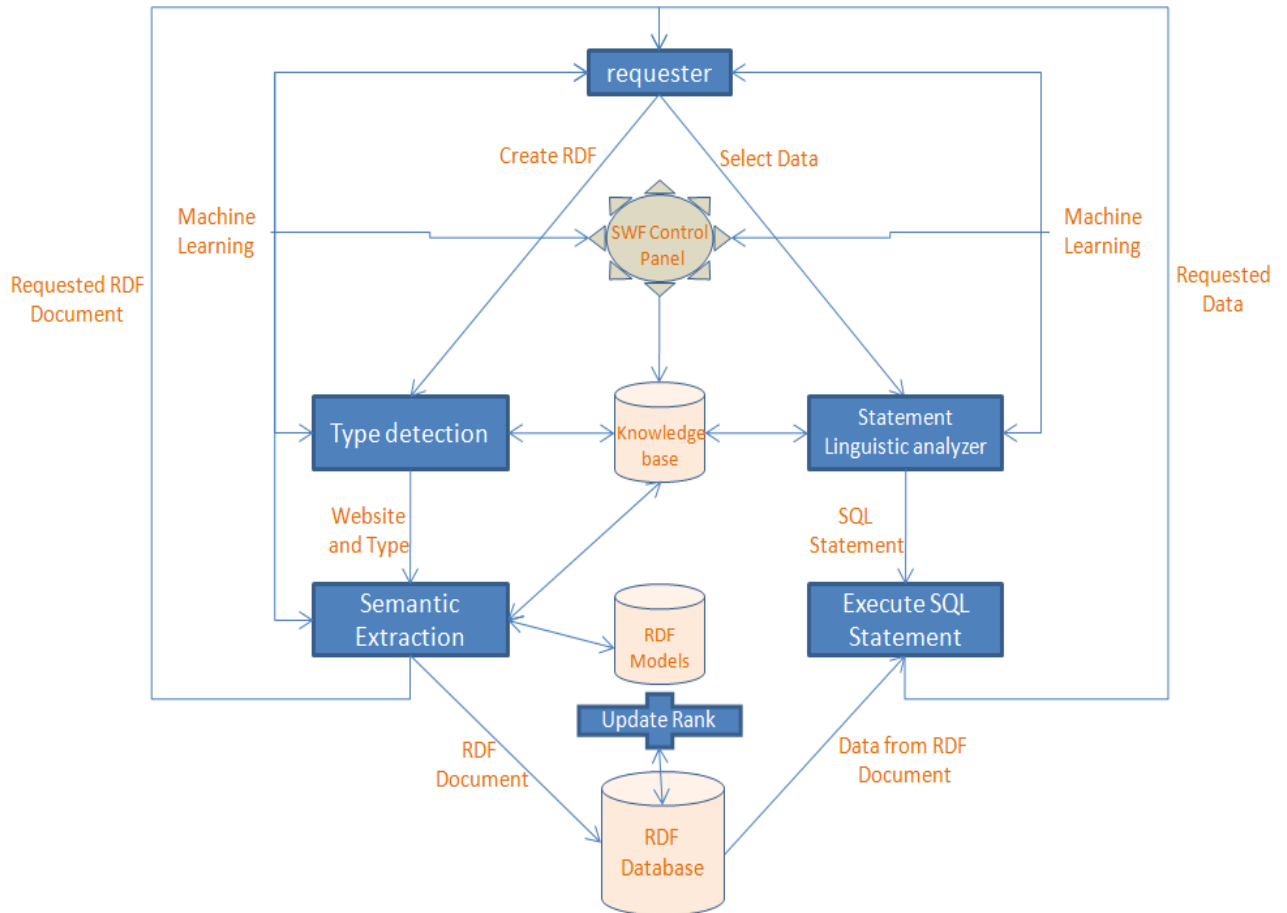


Figure 3.1: SWF architecture

When the requester needs to create an RDF document, the type detection used to determine the type of the website using some semantics from the knowledge base, if the type detection can't detect the type it uses the SWF learning to ask the requester about the type. After detecting the type the RDF generator extract the needed data from the website based on semantic from knowledge base to fill the RDF model, in the case the requested data is not found; the RDF generator use the SWF learning model to ask the requester. The system publishes the new RDF document after extracting the semantics.

When the requester needs to retrieve data using 'select' statement, the linguistic analyzer is used to prepare correct SQL statement to execute in the RDF database. The linguistic analyzer use the knowledge base to find the keywords meanings, if the linguistic analyzer can't understand the meaning it uses the SWF learning to ask the requester about the keywords meaning. After preparing the real SQL statement the system executes it and sends the

retrieved data to the requester. The SWF updates the RDF rank after each ‘select’ statement using the rank updater process. The SWF control panel determined if we can add the requester notes and knowledge to the knowledge base and use it in the future.

This research is a new contribution in the Semantic Web area, which is a new direction in Semantic Web researches area. Figure (3.2) describes the current Semantic web which is used in Swoogle search engine and other search engines like Google and Yahoo.

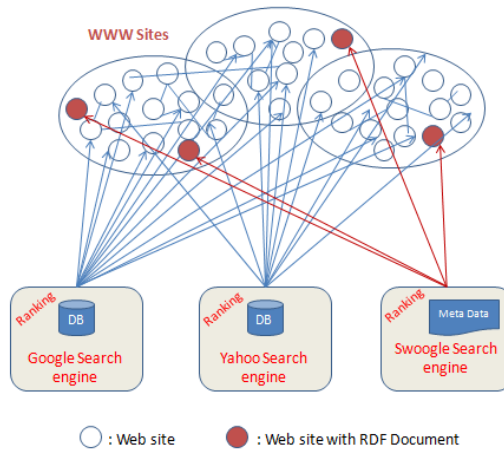


Figure 3.2: The current status of the Web.

In Figure (3.2) the Swoogle is a search engine that use semantic web technologies. The RDF document was created in manually in each web site. But in our proposed framework the RDF documents are created using the RDF generator. Figure (3.3).

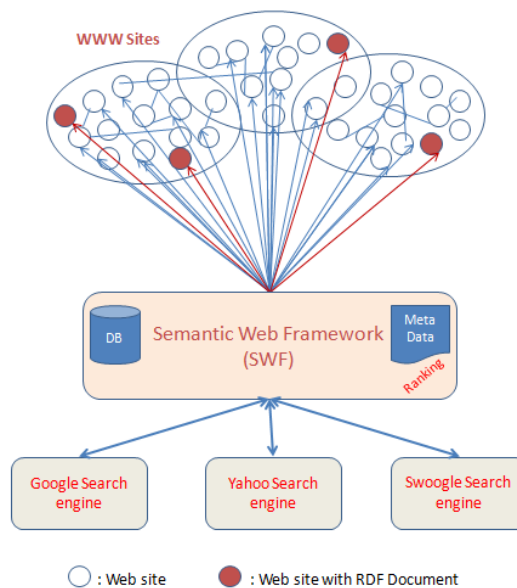


Figure 3.3: The Web after using the Semantic Web Framework.

The next sections describe the SWF architecture main parts in details, The RDF generator, RDF Query System, and Ranking System. RDF generator is the first part in our framework used to generate the RDF document from web pages using predefined model for each web domain. The RDF generator uses knowledge base while generating the RDF document, when the web page is not clear or the model of this page is not defined in the knowledge base; the RDF generator uses the SWF learning to generate the documents. The second part of our framework is RDF query system; this part is a system to retrieve the information from the RDF documents using Query language like SQL. The ranking system is the third part used for ranking web pages to provide it to the search engines and to show the results sorted, this content uses special formula for ranking algorithm to Semantic Web.

### 3.3 The RDF Generator Model

#### 3.3.1 Introduction

The RDF Generator is a platform generates the RDF documents from any web page using predefined models for each internet domain using special web pages classification system. RDF generator aimed to standardize the researchers efforts in Semantic Web by classifying the internet sites to domains, and preparing special RDF model for each domain. RDF generator used web intelligent methods for preparing RDF documents such as ontology based semantics matching system to detect the type of web page, and knowledge base SWF learning system to create the RDF documents accurately and according to the standard models. The following Figure shows the main contents in the RDF generator architecture.

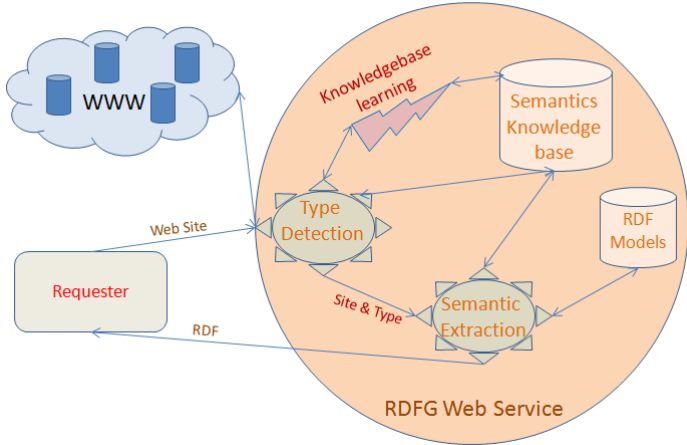


Figure 3.4: RDF generator model.

The RDF generator contains three main parts, the first is website type detection to decide the suitable RDF model for this type, second is semantic extraction to get the data from the websites, and the third is SWF learning system to be more intelligent. In the following sections we introduce these parts in details.

### **3.3.2 Websites type detection**

Based on special web sites classification, RDF generator determined the website type semantically based on semantics on each page which helps the RDF generator to determine the website type. The RDF generator looks for some semantic for each type in the website title, in the description, or in the head and body of the page. Websites type detection depends on knowledge base semantics for each web classification; this knowledge base depends on SWF learning system to be more intelligent and dynamic. In section 3.3.4 we introduce the SWF learning system. Websites type detection used to determine the website type to select the suitable RDF model.

### **3.3.3 Websites Classification**

The web is continues to grow hugely, and the search process in the web is becoming more complex also the direct access to what we need become impossible, so we need methods to arrange the web and help us while using it. The search engine can help us to search in the web but we face another problem which is the need to search in the search engines results. In our framework we propose an idea to classify the web pages to categories, to help us while detecting the type of page, and to help the search engines while preparing the search results and categories it. This idea is a new Websites classification based on our requirements in RDF generating to create standard RDF model for each category. RDF generator uses this classification while detecting the type of web page and select the suitable RDF model to extract the pages description.

Websites are already classified into domains like .com, .net, .info and other, but these domains aren't related to website content or type, so we need another classification strategy to categorize the websites based on pages content. Our classification depends on the website content. The Classification of Websites Based on SWF requirements to generate standards RDF files we classified the websites into the following categories:

1. Forums or Blogs sites: websites used to post topics online, like discussions. It is a general term for an online bulletin board. It is also known by different terms such as discussion group, discussion forum, message board, and online forum.
2. Commerce sites or ecommerce sites: This type is defined as the exchange of money, information, services, and products between economic entities over the Internet, networks, and other digital technologies. E-Commerce will allow an organization to expand into new markets without building expensive facilities. It allows customers to receive information and order products 24 hours a day, 7 days a week.
3. Corporate website: Used to provide background information about a business, organization, or service.
4. Universities and Educational.
5. Download websites: Used for downloading files, such as software, game, and others.
6. Search engine websites: Websites provide a gateway to lookup for other sites. Search engines work by sending out a spider to fetch as many data as possible and store it.
7. Wikis websites: websites which users collaboratively edit, created by volunteer.
8. Web portals: Websites that provide dynamic linking between pages on the Internet.
9. Warez websites: Websites contains illegal content and downloads.
10. Payments site: Websites use sensitive information, such as passwords and credit card details, used by trustworthy person or business such as PayPal.
11. Personal homepage: Websites contain personalized information about people.
12. News websites: This type of websites purpose is reporting public news.
13. Development websites: This type of websites purpose is to provide information and resources related to software development, Web design and the like.
14. Games websites.
15. Other: like Job placement, fun, Erotica and others.

#### **3.3.4 RDF models**

The RDF is a general-purpose language for representing information in the Web (Klyne, and et al., 2004). In RDGF platform we prepare RDF templates for each website type to fill them after extracting the required information. The RDF templates (models) are the same W3C

RDF syntax in addition to adding new parameters to be more general in the web. We add the following parameters in each RDF model:

- Domain.
- Website parts: add the main website parts with links.
- Main paragraph: Description about the website.
- Rank.
- Preparation date.
- Preparation comments: Some notes, errors, face the RDF generator.
- Programmers Hints: Notes for the programmers like the RDF syntax to use it while prepare SQL statements.

And there are other parameters based on model domain such as in universities we add faculties' information, professors' information, and university country and contact.

### **3.3.5 Semantic Extraction**

After determining the website type and select the suitable RDF model we need to extract the information from the website to prepare the RDF. Web information extraction involves locating documents and identifying and extracting the wanted data within the documents. Information extraction systems usually rely on extraction rules that are tailored to a particular information source.

The most challenging aspect of these systems is they must be able to recognize the wanted data among many other pieces of text, for example, mark-up tags, inline code, and navigation hints, among others. The simplest information extraction systems utilize extraction rules that are constructed manually. These systems require a human developer to create new models (patterns) for each information source or for information sources that are structurally changed.

This limits users to accessing information only from predefined information sources. Currently there are two principal methods for identifying interesting data within Web pages are ontology-based extraction and position-based extraction.

- **Ontology-based Extraction.** - Ontology-based information extraction tools feature many of the properties desired for an adaptive Web information extraction system. An ontology-based tool uses domain knowledge to describe data. This includes relationships, lexical appearance, and context keywords. (Embley, and et al.,1998). However, ontology-based tools require that the data be fully described using page-independent features. This means the data must either have unique characteristics or be labeled using context keywords. Unfortunately, all interesting Web data does not necessarily meet these requirements. Some data is freeform and cannot be identified using a specific lexical pattern and also is not labeled. This type of data can only be extracted using its specific location in the HTML page.
- **Position-based Extraction** relies on inherent structural features of HTML documents to accomplish data extraction. Under a position-based extraction system, a HTML document is fed to a HTML parser that constructs a parsing tree that reflects its HTML tag hierarchy. Extraction rules are written to locate data based on the parse-tree hierarchy. If a collection of items is to be retrieved (as from a search results page), a regular expression is constructed to allow multiple items to be retrieved for a hierarchical pattern.

Position-based extraction lacks the resilience of ontology-based extraction. When there are changes to the structure of the target Web pages, it frequently fails. However, it does guarantee a high accuracy of information extraction, with precision and recall being at least 98% (Chidlovskii, 2002). Position-based Extraction can be appropriate when the data to be extracted can only be identified based on its location within a Web page and not on domain information.

In our research we present a domain-oriented approach to automatically web data extraction. This approach is based on a highly efficient patterns structure analysis and allows the extraction of relevant text passages from the pages of a given Web site, and fetching the entire Web site content, and the extraction of the relevant text passages discarding non-useful material.

The data extractor pulls the specific data from the HTML pages. A data extraction process is initiated to locate the tokens within the parsed HTML document. The system searches the set of location key content-text pairs generated by the HTML parser for any of the keywords defined. If a keyword represents a path expression, this indicates that position based extraction is being used and the location key is used for the search.

Otherwise, the data is being extracted using ontology based extraction and the content-text is searched to locate the keyword. Once a keyword is located, the RDF generator data extraction system searches the content text before and after the keyword location to find data that matches a pattern defined in the domain RDF model. When Web data containing an appropriate pattern is found, the data type is used to extract the desired token from the content text.

### **3.3.6 SWF Learning**

We use SWF learning in our framework to learn the knowledge base with semantics related to domains or to help the system while preparing the requested results. SWF uses SWF learning when the requester waiting the results and that in the following position:

- When the RDF generator can't determine the website domain, the SWF learning system asks the requester to select the type and note how can anyone know what type of site, this note used by the SWF control panel to determine if this case is general and add it to the knowledge base or it is just for this website.
- When the RDF generator can't find the needed data to create the RDF document. The SWF learning system asks the requester to write the needed data and note where can anyone find the wanted data in the website paragraphs after what, or before what or in which page, this note used by the SWF control panel to determine if this case is general to add it to the knowledge base or it is just for this website.
- When the SWF Query Language do linguistic analysis for the query statement, if there are some keywords in the statement not found in our knowledge base, the SWF learning asks the requester what the meaning of the Incomprehensible keywords and looks for the keyword another time to correct the query statement.



### 3.4 The RDF Query Language for SWF

In our framework we will use the standard query language (SQL) which used in relational databases to retrieve data from RDF documents. The reason for using the SQL is to use it by programmers without having to learn another query language, and the portability between applications and agents. Our query language uses linguistic analysis to enable the users to write what they want and the system analyze and understand the query. Our RDF query language for SWF depends on the SWF learning system to modify the queries if it has errors or ambiguities.

- **Request Data from RDF documents.**

As mentioned previously we will use the SQL statements to retrieve the requested data from RDF Documents. We will use standard *'select ... from'* to retrieve data from the RDF documents. Because of the data is heterogeneous and the requester does not know how is designed; our system uses linguistic analysis when executing the select statement to enable the requester query about what he wants without the design limitation. So the websites developers or the search engines can use the flexible select statement.

**Main Syntax:** *Select [Key-word1, Key-word2, ...] from web [Where domain= specific-domain or Site =Website-Address or other custom filters];*

#### **Examples:**

- *'Select professors from web where site='www.alquds.edu';'* to request the available professors information in the selected site,
- *'Select professors from web where domain= 'universities';'* to request the available professors information in the selected domain.
- *'Select professors from web where domain= 'universities' and major='Computer';'* to request the available professors information in the selected domain and related to major = computer.
- *'Select professors from web where city="Palestine";'* to request the available professors information in the web and in Palestine.

## The Algorithm:

- Read the first two words in the statement,
  - If not 'select rdf' and not 'refresh rdf' goto 'select something' case,
  - Check if there where statement and Read the condition after where,
  - If the domain selected: look for the domain in the domains table if available set the domain parameter equal it, if not available look for it in the knowledge base to get another word to determine the domain, if there is no synonyms in the knowledge base: if the requester is user ask him about the domain, if the requester is a service set the domain parameter is all and note in the query report about this thing.
  - If the site selected: set the site parameter equal it.
  - Read the words between select and from –the keywords-,
  - Separate the keyword by comma,
  - Search about each keyword in the knowledge base and replace each one,
  - Execute the query to retrieve the data from RDF documents. //here we have correct SQL statement with correct parameters so we execute it using ADO.NET.
- **Request RDF Document.**

SWF query language allows users to request or create RDF document for any website. The websites developers can use this statement while programming to get any RDF directly without facing the trouble of creating RDF from scratch. This statement similar to select ...from statement, as it is in the following statement:

**Main Syntax:** *Select RDF from web where site=Website-Address;*

**Example:** 'select *RDF* from web where site='www.alquds.edu';' to request the RDF document for Al-Quds University website.

**Execution:** SWF executes this select statement by finding the website address in the metadata and returning the RDF document as '.rdf' file after checking if the website address is correct. If the website address is not created in the SWF metadata the system generates the RDF document using RDF generator and returns the created RDF.

- **Refresh RDF statement.**

The SWF refresh the RDF documents periodically, in addition to that the websites developers can add a statement to them applications to refresh the RDF document after each update in the website. This statement has one parameter to pass the website address as it is shown in the following statement:

**Syntax:** *Refresh RDF Website-Address;*

**Example:** ‘*refresh RDF www.alquds.edu;*’ to refresh the RDF document for alquds university website.

**Execution:** The SWF executes the refresh statement by regenerate the RDF document using RDF generator after checking if the website address is correct. The refresh statement doesn’t return any values.

### 3.5 Ranking System for SWF

Ranking is the process of assigning a rank number to the RDF documents according to the importance of the RDF. All search engines use algorithms to calculate the ranking based on the users hits or the number of in and out links. In this thesis we propose a ranking system to sort and introduce the results based on the RDF document history and the related content in the page. Our algorithm has been developed to improve the performance of ranking by adding new parameter to give the weight of each result on the basis of content.

Our Ranking System uses the following formula to update the old RDF rank.

$$\underline{\underline{NewRank = (OldRank) + (((Count\ of\ results * MaxRank) + Weight) / 1000\%)}}$$

*OldRank* is the stored in the RDF document we use it to increase the RDF rank based on previous rank, *MaxRank* is the maximum rank stored in the system and we use it to keep the ranks in reasonable range, the *MaxRank* is useful when the results count is very large it keep the range of ranks reasonable between the *OldRank* and *NewRank*. The weight aims to help in identifying the site ranks in terms of contents; it is calculated by giving the count of similarities in the page contents and the text you searched for.

For example if the results count is 750 and the *MaxRank* is .003251 , then using the ranking formula we will add .00024 to each rank of the result.

The SWF Framework updates automatically the ranking for the results after each searching process using the following algorithm:

```

UpdateRank (RDF[] results) {
    For(i=0;i<results.length;i++) {
        results[i].setRank(results[i].getRank()+(results.length * maxRank+weight)/1000%)
    }
}

```

Figure 3.5: Update rank algorithm

In the figure UpdateRank method receives the results lists as an array of RDF class; the RDF class has setRank method to change the rank of the RDF document directly, getRank to give us the rank of the selected RDF document.

### 3.6 Discussion

We have introduced a system used to prepare RDF documents from websites automatically and store these documents in the database to retrieve them when it required. This system is called the SWF. Figure (3.7) shows the SWF system.

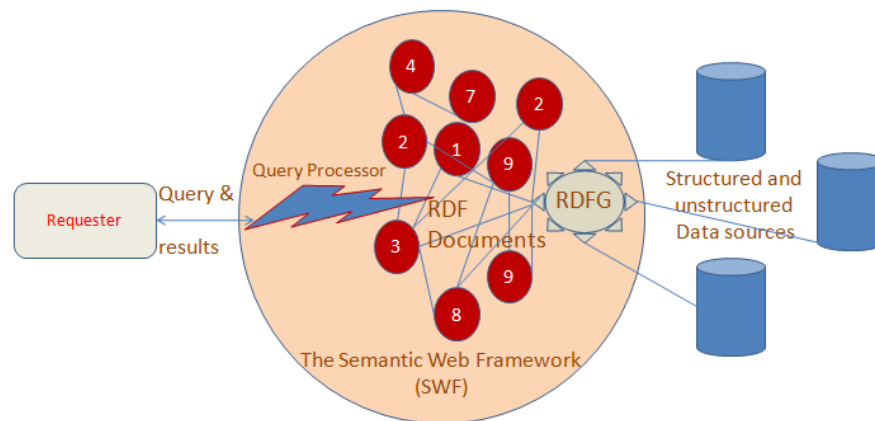


Figure 3.6: The SWF System

The SWF inputs are the structured and unstructured data from the web (websites). The RDF generator process these websites by determining the type of them using some semantics from the knowledge base, if the type detection can't detect the type detection uses the SWF learning to ask the requester about the type. After detecting the type the RDF generator extracts the needed data from the website based on semantic from knowledge base to fill the RDF model,

in the case of not finding the wanted data, the RDF generator uses the SWF learning to ask the requester, after extracting the semantic the system publishes the new RDF document.

After the RDF documents were prepared using RDF generator; the SWF can prepare the outputs for the requesters, when any requester requests data from the SWF system; the SWF processes the request as follow: the linguistic analyzer is used to prepare correct SQL statement from the requester request to execute in the RDF database. The linguistic analyzer uses the knowledge base to find the keywords meanings, if the linguistic analyzer can't understand the meaning it uses the SWF learning to ask the requester about the keywords meaning. After preparing the real SQL statement the system executes it and sends the retrieved data to the requester. After each select statement the rank updater updates the rank for the used RDF documents in the results.

## **Chapter 4**

### **SWF System Analysis and Design**

#### **4.1 Introduction**

This chapter includes the SWF analysis and design details, focusing on the three main parts of the application:

- RDF generation.
- Query Execution (selects data, select RDF, refresh RDF, website summaries..)
- Website Ranking.

The SWF is intended to cover all the web areas and classification, this thesis and its application specialized in the Educational area of the entire web.

#### **4.2 Overall Description**

##### **SWF Perspective**

The SWF is a new version of the web, it is totally based on the web information and relations but in new structure that solves the reusability and organization of them, it is based on other researcher's contributions and brings a solution to the World-Wide-Web.

##### **SWF Features**

SWF generates automatic RDF Documents and returns results to the user queries even if the select statements were not written correct, and provides the search engines with page ranks; also the framework has learning abilities which helps in building its knowledgebase.

## User Classes and Characteristics

Table 4.1: User Classes and Characteristics

User Class	SWF Functions	Class Skills
Researchers	<ul style="list-style-type: none"><li>- Request automatic RDF documents</li></ul>	<ul style="list-style-type: none"><li>- Expert knowledge in Semantic Web Field</li></ul>
Developers	<ul style="list-style-type: none"><li>- Request automatic RDF document using web coding-</li><li>- Executing SQL statements using web coding</li></ul>	<ul style="list-style-type: none"><li>- knowledge in Semantic Web field</li><li>- Web development skills</li><li>- Database skills</li></ul>
End-users	<ul style="list-style-type: none"><li>- Web searching</li><li>- Web summarizing</li><li>- Web Directories</li></ul>	<ul style="list-style-type: none"><li>- Browsing skills</li><li>- Little SQL skills</li></ul>

### Operating Environment

The SFW application is a web-based one that can run at any operating system that support web browsing including WINDOWS, UNIX and others.

### Design and Implementation Constraints

- The RDF generation in the current time depends on English semantics to detect the webpage positions and extract the data, the websites that are developed using other languages in defining theses positions cannot be correctly handled in the RDF generation process.
- The websites that include components such as flashes in main positions in the WebPages limits the output of full RDF documents.
- The security level in the websites, some websites may prevent the SWF from accessing the source code and creating the RDF or website summary.
- The last modification dates of websites are missed in some websites; this is handled by regular RDF updating.

## **User Documentation**

User manual is prepared to help in using the SWF online application, refer to appendix 3.

## **Assumptions and Dependencies**

- The requested website main contents are English-based.
- The SWF can find synonyms to the ambiguous words in the query statements from the knowledgebase or the requester.
- The requested website will be finally implemented on one of the RDF models based on specific website classification.

## **4.3 Functional Requirements**

1. The framework shall provide platform to create managed web structure.
  - 1.1. It shall solve the closed domains, modeling languages complexity, web entities realization and metadata sharing problems
2. The framework shall generate automatic RDF documents for the requested websites:
  - 2.1. It shall determine the website type based on the framework knowledgebase classifications.
  - 2.2. It shall use the type predefined modeling language and semantics to create the website RDF
  - 2.3. It shall learn by asking the user for more details if it can't determine the type
3. The framework shall save the generated RDF documents in accessible knowledgebase
4. The framework shall keep up-to-date RDF documents for the websites
  - 4.1. It shall check the RDF document creation time against the website last modification time
  - 4.2. It shall weekly update the RDF documents for the websites that do not contain modification time
5. The framework shall summarize the requested websites for the users
6. The framework shall enable the user to search the web using basic and not well-defined select statements
  - 6.1. It shall automatically correct the user queries before start searching



- 6.2. It shall learn by asking the user for clarifications if it failed to correct the query statement
7. The framework shall enable the admin to add new knowledge from users to its knowledgebase
8. The framework shall update the WebPages based on specific algorithm to optimize the search results

#### **4.4 External Interface Requirements**

##### **User Interfaces**

Users of the SWF application are researcher, developers, end users and the application admin, each of them has services they can use, as follows:

- Researchers, developers and end users: can use all the SWF application services without the need for accounts.
- Admin: can manage the application and its knowledge base such as database management, logs management and the learning system by special account.

##### **Hardware Interfaces**

The hardware used to access the SWF online application is the user's computer connected to the internet through wired or wireless connection.

##### **Software Interfaces**

The operating system needed to access the application can be of any type and any version that support the network connection with web browsing capabilities, IE 8 and Firefox are recommended as the browsers.

##### **Communications Interfaces**

The SWF application is totally dependable on the internet connections; to use the application and to execute request s and provide services.

## **4.5 Non-Functional Requirements**

### **Extensibility**

The SWF can cover the entire web areas, but in this thesis we implement an application to cover the educational area, the application is fully extensible to cover the other web areas classifications.

### **Portability**

The SWF can work on any operating system that includes or can operate web browser.

### **Maintainability**

- Correct defects: can be solved and maintained easily and quickly, but requires uploading work
- Adding new feature: need new development lifecycle to cover any other web area, it is hard not quick.

### **Usability**

The SWF can application tools such as generating RDF and summarize websites and execute queries are easily to be used by inexperience users.

### **Accessibility**

- The current SWF application can be used by user who has the abilities to use computers screens and keyboards, no sound facilities are supported.

## 4.6 The System Design

### 4.6.1 SWF Flowcharts

#### - RDF Flowchart

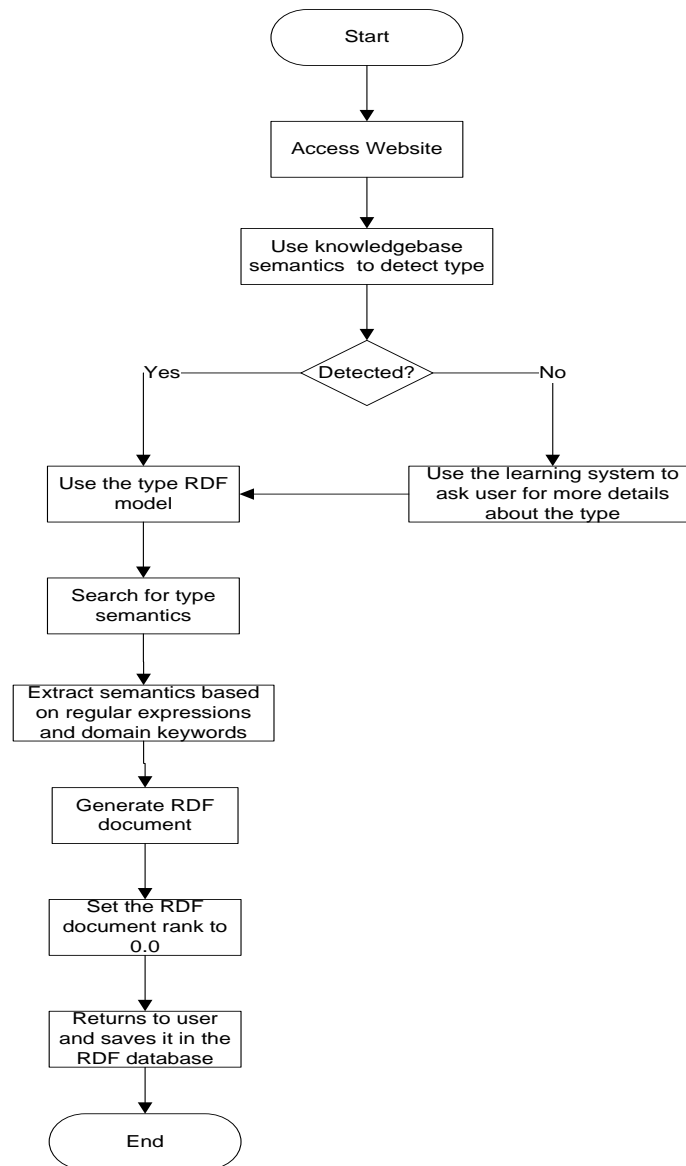


Figure 4.1: RDF Flowchart

## - Query Flowchart

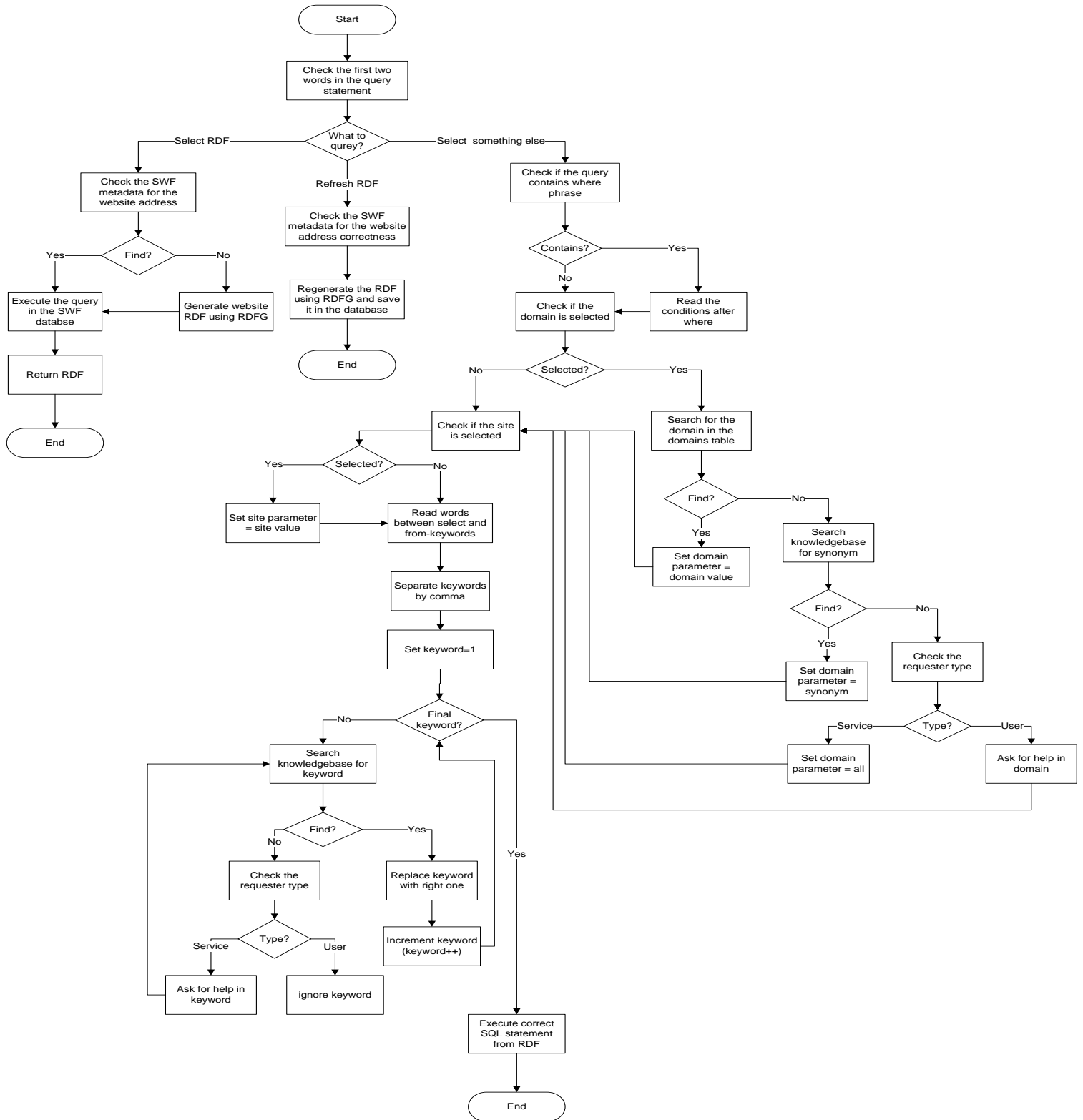


Figure 4.2: Query Flowchart

- Website Summary Flowchart

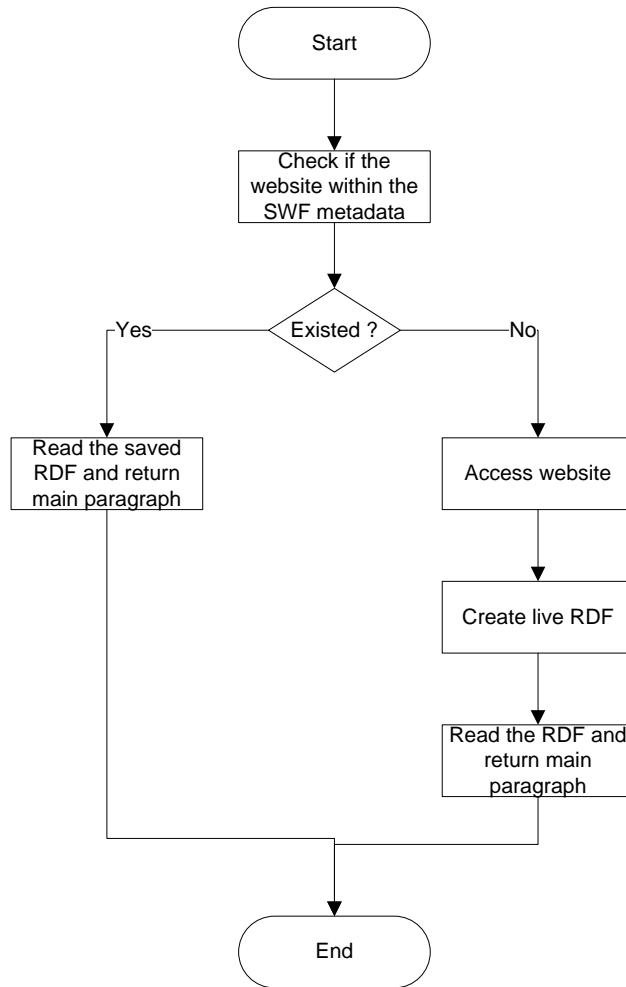


Figure 4.3: Web Summary Flowchart

- **Update Rank Flowchart**

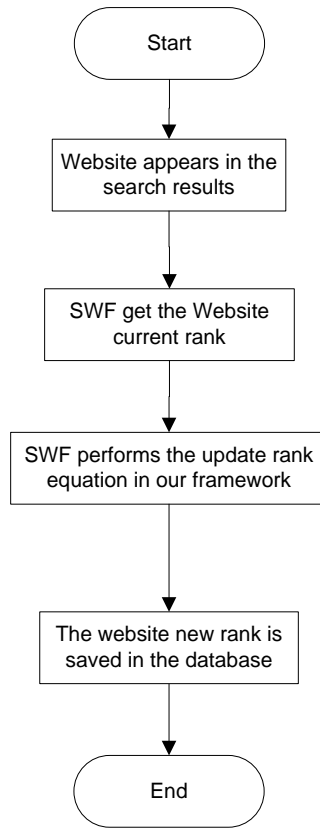


Figure 4.4: Update Rank Flowchart

### 4.6.2 SWF Use Case Diagram

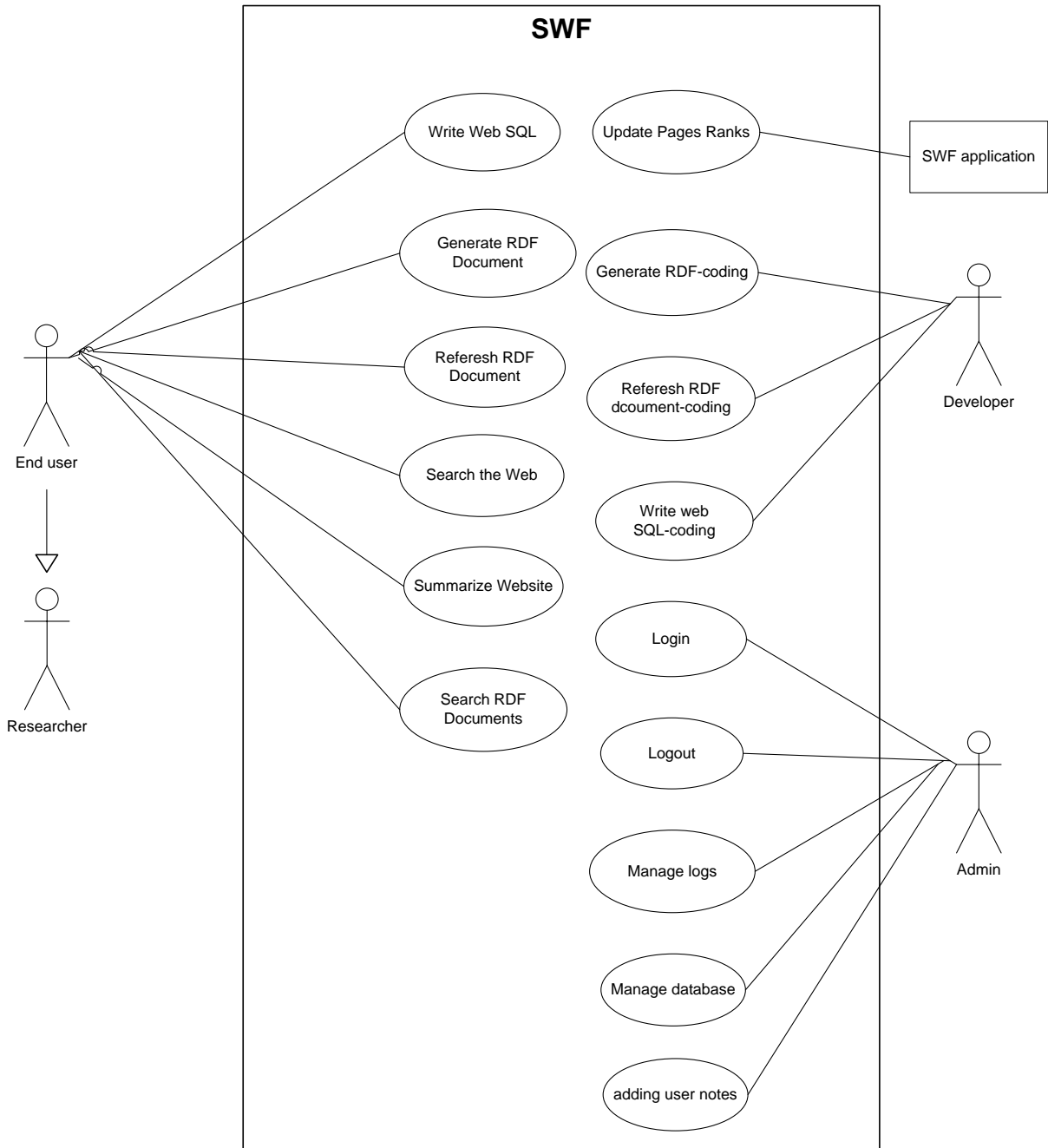


Figure 4.5: SWF Use Case Diagram

### 4.6.3 Narrative Use Cases

#### *Generate RDF Document Use case*

Use Case ID:	1		
Use Case Name:	Generate RDF Document		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	End user, Researcher, Web service
Description:	The use case describes the creating of automatic RDF document using the SWF online application
Preconditions:	<ol style="list-style-type: none"> <li>1. The actor provides website URL</li> <li>2. The website main contents are English-based</li> <li>3. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Return RDF document to the actor</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The End user/ Researcher/ Web service opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End user/ Researcher/ Web service types <a href="http://www.nadasys.com/swf">http://www.nadasys.com/swf</a> in the address bar and presses enter</li> <li>4. The browser opens SWF website</li> <li>5. The End user/ Researcher/ Web service chooses the link named “create RDF”</li> <li>6. The web browser returns the requested page</li> <li>7. The End user/ Researcher/ Web service types the targeted website URL in the website URL field and clicks Generate RDF button</li> <li>8. The application checks the website name against the knowledgebase</li> <li>9. The application accesses the website and search for semantics to detect its type</li> <li>10. The application uses the suitable RDF model for the website from the</li> </ol>



	<p>application database</p> <p>11. The application starts to detect main position in the website pages and extract data</p> <p>12. The application prepare the RDF document from the extracted data and set the website ranks to 0.0</p> <p>13. The application returns the RDF document to the requester</p>
Alternative Flows:	<p><b>A1:</b> the website URL is not correct</p> <p>This alternative starts at point 8:</p> <p>9. The application ask the user to provide correct website URL, the sequence returns to point 7</p> <p><b>A2:</b> the website domain cannot be detected</p> <p>This alternative starts at point 9:</p> <p>10. The application ask the user to provide a help on detecting the domain, the sequence returns to point 10</p>
Errors:	<p><b>E1:</b> the website main data cannot be extracted</p> <p>This alternative starts at point 11, the application return no result and the sequence ended</p> <p><b>E2:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended</p>

*Search RDF Documents Use case*

Use Case ID:	2		
Use Case Name:	Search RDF Documents		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Researcher, Web service
Description:	The use case describes the selecting of existed RDF document using the SWF online application
Preconditions:	<ol style="list-style-type: none"> <li>1. The actor know how to write select RDF statement</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Return RDF document to the actor</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The Researcher/ Web service opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The Researcher/ Web service types <a href="http://www.nadasys.com/swf">http://www.nadasys.com/swf</a> in the address bar and presses enter</li> <li>4. The browser opens SWF website</li> <li>5. The Researcher/ Web service chooses the link named “Web SQL”</li> <li>6. The web browser returns the requested page</li> <li>7. The Researcher/ Web service types the select statement in the Enter SQL field and clicks Execute button</li> <li>8. The application searches for the RDF in the saved RDF documents in the application database</li> <li>9. The application return the RDF document to the requester</li> </ol>
Alternative Flows:	<p><b>A1:</b> the website RDF is not existed</p> <p>This alternative starts at point 8:</p> <ol style="list-style-type: none"> <li>9. The application access the website and create live RDF document, the sequence returns to point 9</li> </ol>
Errors:	<p><b>E1:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

***Refresh RDF Document Use case***

Use Case ID:	3		
Use Case Name:	Refresh RDF document		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Researcher
Description:	The use case describes the refresh the existed RDF document to make it up-to-date with the website contents
Preconditions:	<ol style="list-style-type: none"> <li>1. The actor know how to recreate the RDF document</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Having new RDF document copy</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The Researcher opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The Researcher/ Web service types <a href="http://www.nadasys.com/swf">http://www.nadasys.com/swf</a> in the address bar and presses enter</li> <li>4. The browser opens SWF website</li> <li>5. The Researcher chooses the link named “Web SQL”</li> <li>6. The web browser returns the requested page</li> <li>7. The Researcher types the refresh statement in the Enter SQL field and clicks Execute button</li> <li>8. The application searches for the RDF in the saved RDF documents in the application database</li> <li>9. The application recreates the RDF document and saves it</li> </ol>
Alternative Flows:	<p><b>A1:</b> the website RDF is not existed</p> <p>This alternative starts at point 8:</p>

	10. The application access the website and recreate, the sequence returns to point 9
Errors:	<b>E1:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.

*Write Web SQL Use case*

Use Case ID:	4		
Use Case Name:	Write Web SQL		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	End user, Researcher
Description:	The use case describes how the user can write web SQL and how the application returns the results
Preconditions:	<ol style="list-style-type: none"> <li>1. The actor has experience in writing SQL statements</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns the requested query results</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The End user/ Researcher/ Web service opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End user/ Researcher/ Web service types <a href="http://www.nadasys.com/swf">http://www.nadasys.com/swf</a> in the address bar and presses enter</li> <li>4. The browser opens SWF website</li> <li>5. The End user/ Researcher/ Web service chooses the link named “Web SQL”</li> <li>6. The web browser returns the requested page</li> <li>7. The End user/ Researcher/ Web service types the select statement and then presses the Execute button</li> </ol>

	<p>8. The application reads the first two words</p> <p>9. The application check if the where phrase exists</p> <p>10. The application reads the conditions after it</p> <p>11. The application reads the domain</p> <p>12. The application searches the domains table</p> <p>13. The application set the domain parameter equals the domain value</p> <p>14. The application reads the site</p> <p>15. The application reads the keywords between select and from</p> <p>16. The application separates keywords by a comma</p> <p>17. The application searches the knowledgebase to find synonyms</p> <p>18. The application replaces each keywords with its synonym</p> <p>19. The application executes the corrected SQL statement</p> <p>20. The application returns results to the requester</p>
<p>Alternative Flows:</p>	<p><b>A1:</b> the application didn't find the where phrase This alternative starts at point 9: The sequence continue at point 11</p> <p><b>A2:</b> the application didn't read the domain This alternative starts at point 11: The sequence continue at point 14</p> <p><b>A3:</b> the application didn't find the domain in the domain table This alternative starts at point 12: 13. The application looks for it in the knowledgebase to find another word 14. The application determine the domain, and the sequence continue at point 13</p> <p><b>A4:</b> the application didn't find the another word for the domain the knowledgebase This alternative starts at point 12: 13. The application checks the requester type 14. The application asks the requester if it is a user to help in detecting the domain and set the domain parameter to the detected domain, or sets the domain</p>

	<p>parameter to all if the requester is a service, and the sequence continue at point 14</p> <p><b>A5:</b> the application didn't find the synonyms for keywords in the knowledgebase This alternative starts at point 17:</p> <p>18. The application asks the user for the meanings of the ambiguous words, the sequence continue at point 17</p>
Errors:	<p><b>E1:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

***Search the web Use case***

Use Case ID:	5		
Use Case Name:	Search the Web		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010
Actors:	End user		
Description:	The use case describe searching the web for quick information about sites, their links and ability to read their RDF documents		
Preconditions:	<ol style="list-style-type: none"> <li>1. The End user types the website URL correctly</li> <li>2. Internet connection</li> </ol>		
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns quick search results</li> </ol>		
Normal Flow:	<ol style="list-style-type: none"> <li>1. The End user opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End user types <a href="http://nadasys.com/swf">http://nadasys.com/swf</a> in the address bar and presses enter</li> <li>4. The browser opens SWF website</li> <li>5. The End user types the website URL in the "search the web" field</li> </ol>		

	6. The web browser returns the search results to the requestor
Alternative Flows:	<b>A1:</b> the website URL is not correct This alternative starts at point 5: The sequence continue at point 5
Errors:	<b>E1:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.

*Summarize Website Use case*

Use Case ID:	6		
Use Case Name:	Summarize Website		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	End user
Description:	The use case describe searching preparation of websites summaries
Preconditions:	1. The End user types the website URL correctly 2. Internet connection
Post-conditions:	1. Returns quick search results
Normal Flow:	1. The End user opens the web browser 2. The web browser returns the computer homepage 3. The End user types <a href="http://nadasys.com/swf">http://nadasys.com/swf</a> in the address bar and presses enter 4. The browser opens SWF website 5. The End user chooses the link named “Web Summary”

	6. The web browser returns the requested page 7. The End user types the website URL in the website URL field 8. The application accesses the website and create RDF document 9. The application returns main paragraph the RDF
Alternative Flows:	<b>A1:</b> the website URL is not correct This alternative starts at point 8: The sequence continue at point 7
Errors:	<b>E1:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.

***Login Use case***

Use Case ID:	7		
Use Case Name:	Login		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin login process to the SWF online application
Preconditions:	1. The Admin has private account on the application 2. Internet connection
Post-conditions:	1. Returns Control Panel page
Normal Flow:	1. The Admin opens the web browser 2. The web browser returns the computer homepage 3. The Admin types <a href="http://nadasys.com/swf">http://nadasys.com/swf</a> in the address bar and presses enter



	<ol style="list-style-type: none"> <li>4. The browser opens SWF website</li> <li>5. The Admin chooses the link named “Login”</li> <li>6. The web browser returns the requested page</li> <li>7. The Admin types his username in the username field and his password in the password field and press login button</li> <li>8. The application checks the entered values against the login information</li> <li>9. The application returns the Control Panel page to the admin</li> </ol>
Alternative Flows:	<p><b>A1:</b> the username or password or both are wrong</p> <p>This alternative starts at point 8:</p> <ol style="list-style-type: none"> <li>1. The application returns the admin to the Login page</li> <li>2. The applications asks the admin to retry the login process, the sequence continue at point 7</li> </ol>
Errors:	<p><b>E1:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

***Logout Use case***

Use Case ID:	8		
Use Case Name:	Logout		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin logout process from the SWF online application
Preconditions:	<ol style="list-style-type: none"> <li>1. The Admin already logged in the application</li> <li>2. Internet connection</li> </ol>

Post-conditions:	1. Returns SWF application homepage
Normal Flow:	1. The Admin clicks the “logout” link in any page in the application 2. The application kills the active session 3. The applications return the homepage to the admin
Alternative Flows:	
Errors:	

***Manage Logs Use case***

Use Case ID:	9		
Use Case Name:	Manage Logs		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin management of logs, this use case extends the Login Use Case
Preconditions:	1. The Login Use Case is preformed 2. Internet connection
Post-conditions:	1. Returns SWF application logs pages
Normal Flow:	1. The Admin chooses the “logs” link 2. The application access the database and the logs information 3. The application selects all the logs information 4. The application returns the logs in the logs page to the admin
Alternative Flows:	

Errors:	<p><b>E1:</b> the application cannot access the logs information</p> <p style="padding-left: 40px;">This sequence starts at point 2, the application returns error and the sequence stops</p> <p><b>E2:</b> the internet connection failed</p> <p style="padding-left: 40px;">This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>
---------	--

***Manage RDF Generator Use case***

Use Case ID:	10		
Use Case Name:	Manage RDF Generator		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin management of generated RDF documents, this use case extends the Login Use Case
Preconditions:	<ol style="list-style-type: none"> <li>1. The Login Use Case is preformed</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns SWF application RDF generator Management page</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The Admin chooses the “RDF generator Management” link</li> <li>2. The application access the database and the saved RDF documents</li> <li>3. The application selects all the RDF documents in the database</li> <li>4. The application returns the RDF documents as a list in the left side of the page, and the RDF contents in the right side</li> </ol>
Alternative Flows:	

Errors:	<p><b>E1:</b> the application cannot access the RDF documents</p> <p style="padding-left: 40px;">This sequence starts at point 2, the application returns error and the sequence stops</p> <p><b>E2:</b> the internet connection failed</p> <p style="padding-left: 40px;">This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>
---------	---

***Manage Database Use case***

Use Case ID:	11		
Use Case Name:	Manage Database		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin management of the database through SQL statements, this use case extends the Login Use Case
Preconditions:	<ol style="list-style-type: none"> <li>1. The Login Use Case is preformed</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns SWF application Database Management page</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The Admin chooses the “Database Management” link</li> <li>2. The application returns the requested page</li> <li>3. The Admin writes the SQL statement and press Execute button</li> <li>4. The application executes the SQL by accessing the database contents and perform the requested operation all</li> </ol>
Alternative Flows:	

Errors:	<p><b>E1:</b> the admin SQL statement is wrong</p> <p style="padding-left: 40px;">This sequence starts at point 4, the application returns error and the sequence stops</p> <p><b>E2:</b> the internet connection failed</p> <p style="padding-left: 40px;">This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>
---------	--

*Adding User Notes Use case*

Use Case ID:	12		
Use Case Name:	Adding User Notes		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Admin
Description:	The use case describes admin management of the knowledgebase, this use case extends the Login Use Case
Preconditions:	<ol style="list-style-type: none"> <li>1. The Login Use Case is preformed</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns SWF application Knowledgebase Management page</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The Admin chooses the “Knowledgebase Management” link</li> <li>2. The application returns the requested page</li> <li>3. The Admin checks the new user notes</li> <li>4. The admin decides what to add to the knowledgebase</li> <li>5. The admin add new knowledge and eliminate unnecessary notes</li> <li>6. The application executes the admin actions on the knowledgebase</li> </ol>

Alternative Flows:	
Errors:	<p><b>E1:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

***Generate RDF-coding Use case***

Use Case ID:	13		
Use Case Name:	Generate RDF-coding		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Developer
Description:	The use case describes how developers can use coding to generate the RDF documents without using the “Generate RDF” facility
Preconditions:	<ol style="list-style-type: none"> <li>1. The developer have good experience in writing RDF generating code and using URL parameters</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns the requested website RDF document to the developer</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The developer opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End developer types <a href="http://nadasys.com/swf/RDFG.aspx">http://nadasys.com/swf/RDFG.aspx</a> in the address bar and followed by the URL parameter that equals the website and then presses enter</li> <li>4. The application returns the requested RDF document to the developer</li> </ol>
Alternative Flows:	

Errors:	<p><b>E1:</b> the developer wrongly write the RDF generation coding in the address bar This alternative starts at point 3, the web application returns error in generating the RDF and the sequence ended.</p> <p><b>E2:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>
---------	--

***Refresh RDF Document-coding Use case***

Use Case ID:	14		
Use Case Name:	Refresh RDF Document-coding		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Developer
Description:	The use case describes how developers can use coding to refresh RDF documents without using the “Web SQL” facility
Preconditions:	<ol style="list-style-type: none"> <li>1. The developer have good experience in writing RDF refresh code and using URL parameters</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Having new copy of the RDF document</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The developer opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End developer types <a href="http://nadasys.com/swf/SQL.aspx">http://nadasys.com/swf/SQL.aspx</a> in the address bar and followed by the SQL parameter that equals the refresh statement for the website and then presses enter</li> <li>4. The application notifies the developer with the completion of the refreshment</li> </ol>

Alternative Flows:	
Errors:	<p><b>E1:</b> the developer wrongly write the RDF refresh coding in the address bar This alternative starts at point 3, the web application returns error in refreshing the RDF and the sequence ended.</p> <p><b>E2:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

*Write Web SQL-coding Use case*

Use Case ID:	15		
Use Case Name:	Write Web SQL-coding		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	Developer
Description:	The use case describes how developers can use coding to select data from the web without using the “Web SQL” facility
Preconditions:	<ol style="list-style-type: none"> <li>1. The developer has good experience in writing SQL statements and using URL parameters</li> <li>2. Internet connection</li> </ol>
Post-conditions:	<ol style="list-style-type: none"> <li>1. Returns the query results to the developer</li> </ol>
Normal Flow:	<ol style="list-style-type: none"> <li>1. The developer opens the web browser</li> <li>2. The web browser returns the computer homepage</li> <li>3. The End developer types <a href="http://nadasys.com/swf/SQL.aspx">http://nadasys.com/swf/SQL.aspx</a> in the address bar</li> </ol>



	<p>and followed by the SQL parameter that equals the query statement from the web and then presses enter</p> <p>4. The application returns results to the developer</p>
Alternative Flows:	
Errors:	<p><b>E1:</b> the developer wrongly write the SQL statement and coding in the address bar This alternative starts at point 3, the web application returns error in querying from the web and the sequence ended.</p> <p><b>E2:</b> the internet connection failed This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

### *Update Page Rank Use case*

Use Case ID:	16		
Use Case Name:	Update Page Rank		
Created By:	Ahmed Nada	Last Updated By:	Ahmed Nada
Date Created:	7-6-2010	Date Last Updated:	7-6-2010

Actors:	SWF application
Description:	The use case describes the process of updating the websites ranks after searching the web
Preconditions:	1. Internet connection
Post-conditions:	1. The new rank for the website is greater than the old rank
Normal Flow:	<p>1. The user opens the web browser</p> <p>2. The web browser returns the computer homepage</p>

	<ol style="list-style-type: none"> <li>3. The user types <a href="http://nadasys.com/swf">http://nadasys.com/swf</a> in the address bar</li> <li>4. The web browser returns the home page for the SWF online application</li> <li>5. The user chooses the link named “Web SQL”</li> <li>6. The application returns the requested page</li> <li>7. The user types the search statement</li> <li>8. The application returns the results to the user</li> <li>9. The application searches the SWF database for the websites RDF documents</li> <li>10. The application updates the websites ranks for the websites that appears in the search results</li> </ol>
Alternative Flows:	<p>A1: the website RDF in not existed in the database</p> <p>This sequence starts at point 9:</p> <ol style="list-style-type: none"> <li>10. The application creates RDF document and set the rank to 0.0, the sequence returns to point 10</li> </ol>
Errors:	<p><b>E2:</b> the internet connection failed</p> <p>This alternative starts at any point, the web browser returns connection error and the sequence ended.</p>

#### 4.6.4 Database Entity Relationship Diagram

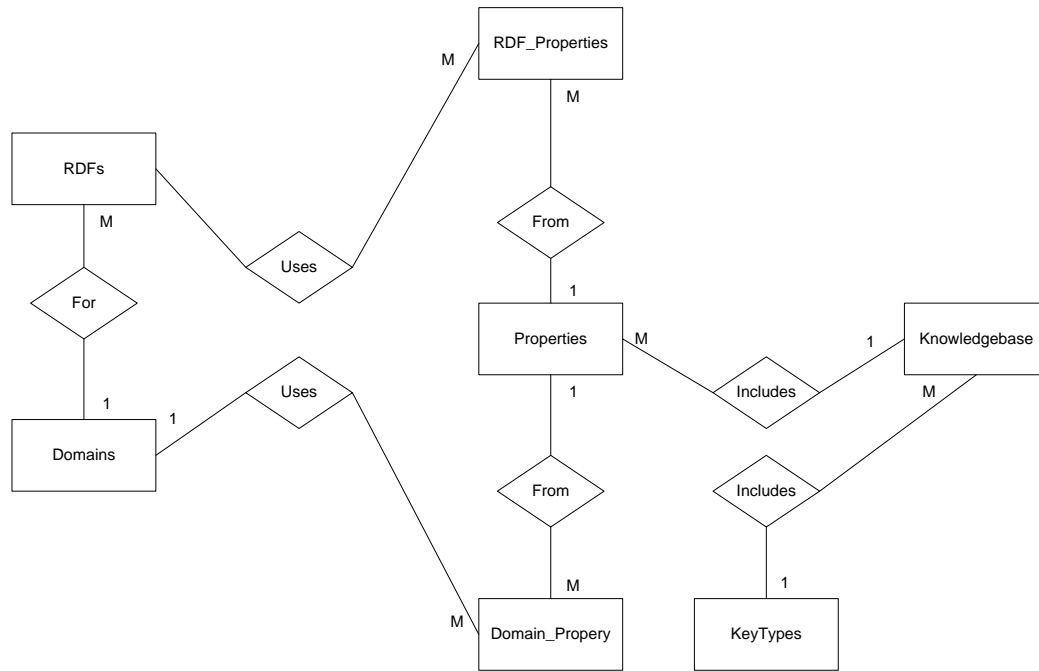


Figure 4.6: SWF ERD

#### 4.6.5 Database Entities Specifications

\*Create SWFDB database user

Table 4.2: RDFs Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> RDFs			<b>Page:</b> 1 Of 1
Field Name	Format			Validation Rules
	Type	Length	Null	
RDF_ID	N	6	No	*primary key
Domain_ID	N	6	No	
Site	V	100	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **RDFs** table is a base table that contains the RDF documents information.

Table 4.3: Domains Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> Domains			<b>Page:</b> 1 <b>Of</b> 1
<b>Field Name</b>	<b>Format</b>			<b>Validation Rules</b>
	<b>Type</b>	<b>Length</b>	<b>Null</b>	
Domain_ID	N	6	No	*primary key
Domain_Name	V	30	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **Domains** table is a base table that contains the domains classifications information.

Table 4.4: Properties Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> Properties			<b>Page:</b> 1 <b>Of</b> 1
<b>Field Name</b>	<b>Format</b>			<b>Validation Rules</b>
	<b>Type</b>	<b>Length</b>	<b>Null</b>	
Property_ID	N	6	No	*primary key
Property_Name	V	30	No	
Key_ID	N	6	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **Properties** table is a base table that contains all the SWF properties.

Table 4.5: RDF\_Properties Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> RDF_Properties			<b>Page:</b> 1 Of 1
Field Name	Format			Validation Rules
	Type	Length	Null	
RDF_ID	N	6	No	*primary key
Property_ID	N	6	No	
Property_Value	N	5	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **RDF\_Properties** table is a transaction table that contains the RDF properties.

Table 4.6: Domain\_Property Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> Domain_Property			<b>Page:</b> 1 Of 1
Field Name	Format			Validation Rules
	Type	Length	Null	
Domain_ID	N	6	No	*primary key
Property_ID	N	6	No	
Purpose	V	200	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **Domain\_Property** table is a transaction table that contains the domains properties.

Table 4.7: KeyTypes Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> KeysTypes			<b>Page:</b> 1 Of 1
Field Name	Format			Validation Rules
	Type	Length	Null	
Type_ID	N	6	No	*primary key
Type_Name	V	30	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **KeysTypes** table is a base table that contains the keys types' information; it is a part of the SWF knowledgebase.

Table 4.8: Knowledgebase Table

<b>Project</b> Control Flow	<b>Author</b> Al-Quds University Software			<b>Date</b> 18-Jan-2010
<b>Entity Subject</b> CES	<b>Entity Name</b> Knowledgebase			<b>Page:</b> 1 Of 1
Field Name	Format			Validation Rules
	Type	Length	Null	
Key_ID	N	6	No	*primary key
Type_ID	N	6	No	
Key_Text	V	30	No	

\*key    \*\*Foreign Key    +Key of Father Entity

(N) Integer Number    (F) Floating Number    (V) Varchar2 String    (D) Date

The **Knowledgebase** table is a base table that contains the knowledgebase information; it is a part of the SWF knowledgebase.

## Chapter 5

### Results and Evaluation

#### 5.1 Results

The SWF makes the Semantic Web applicable through some solution to help the developers and the Semantic Web researchers:

- RDF standards models for each domain to use it when create RDF.
- RDF generator: to prepare websites description documents based on the RDF standards models.
- Dynamic RDF Generating to add special parameters to the RDF or to prepare undefined domain in the SWF and using SWF learning system.
- Using SQL to retrieve data from RDF documents, it will be as web query language.
- Adding a Rank parameter in each RDF Document to store hits counter. To help search engine for sorting the results.

This research is a contribution in the Semantic Web field it suggests:

- RDF Standards models for each domain.
- RDF Generator based on RDF Standards models.
- Adding dynamic parameters to RDF documents.
- RDF Ranking System based specials formula.
- Standard SQL for RDF with linguistic analysis.
- New Framework to include the all semantic web technologies.

After applying the SWF on the web there will be many benefits in the web:

- The search engines can search using query language from SWF, so it will be more advanced and specific results.
- The web programmers can use the internet as database.
- New Platform for web Research.

## 5.2 Evaluation

### - Evaluation of Creating RDF Manually and Automatically:

RDF is created through semantic matching and extracting based on knowledge base. In SWF we introduce a new platform to do all RDF creating steps, this platform reduces the complexity of modeling languages as seen by the researchers and developers, thus saves the time in learning new modeling languages, standardizes all RDF document in the internet based on domains, and this platform also reduces the creating RDF time.

To prove these features we looked at a sample of three RDF for universities websites that was automatically created using the RDF generator. RDF documents were created manually. The results were manually evaluated and by separating the matching to find accurate time as in the following table.

Table 5.1: The calculated time for creating RDF manually and automatically

Websites	Alquds.edu		Birzeit.edu		Najah.edu	
	Manually	Automatically	Manually	Automatically	Manually	Automatically
rdf:domain	5 S	1 S	5 S	1 S	5 S	1 S
rdf:title	10 S	1 S	10 S	1 S	10 S	1 S
rdf:keywords	30 S	2 S	40 S	1 S	30 S	1 S
rdf:about	1 M	10 S	50 S	10 S	1 M	10 S
rdf:contact	7 S	1 S	55 S	15 S	40 S	17 S
rdf:mainparts	20S	4 S	30 S	5 S	30 S	6 S
rdf:links	3 M	13 S	3 M	10 S	2 M	13 S
rdf:files	2 M	13 S	2 M	10 S	2 M	13 S
rdf:Faculties	7 M	30 S	5 M	40 S	5 M	35 S
rdf:persons	7 M	30 S	5 M	40 S	5 M	35 S
Sum:	21.12 M	1.45 M	18.1 M	2.13 M	16.55 M	2.12 M

S: Seconds, M: Minuets

The test results showed the RDF generator achieved the reducing of complexity in modeling languages, learning time, and reducing the RDF creating time.



- **Evaluation of Why SQL instead of SPARQL:**

SPARQL is the query language of the Semantic Web. Designed from the ground up to be capable of simultaneously querying and integrating the results from multiple data sources scattered across the web. SQL is a standard Query language for relational databases, it well designed to represent highly regular (structured) data such as that used by many business processes. Widely used examples include personnel and departments, students and classes, and manufacturing components. Such data usually includes a value for every column of every table. The syntax of the SQL language focuses on identifying data for which most or all components are available and combining data based on the values of those components.

In this thesis we use the SQL syntax because it is standardized, used by many business processes and widely used more than SPARQL in for example, the personnel and departments, students and classes, and manufacturing components. Thus, we have reduced the time to learn a new language for using our framework, reduce the query text size because we use standards syntax and functions, and save time in learning the RDF documents accurately because the language can understand what you write.

- **Evaluation of Preparing Websites Description Manually and Automatically:**

SWF introduces a new web service and it is the first of its kind, service for preparing summarizes for the websites. The idea in this service is to convert the RDF document to paragraph that includes the main information about the website.

To evaluate this feature we looked at a sample of 5 websites in universities domain, and we prepare the summarizing of websites using SWF and C# language it take 75 second for 5 websites, and in the same test case we try to prepare the summarizing manually it take 3-7 minutes for each website. So we provide an effective tool to summarizing the websites automatically and use it in the development systems.

- **Statistical Evaluation**

The aim of the study: The purpose of this survey is to discover the semantic web researchers' attitudes toward the attempt of creating Semantic Web Framework.

Society of the study: Researchers in the Semantic Web field.

The Statistics way: Use the descriptive statistician.

The study sample: The study was conducted on a sample of (26) Researcher in the Semantic Web field, whom have been randomly selected, tables (5.2), (5.3) showing the distribution of the study sample according to the independent variables:

Table 5.2: The distribution of the study sample according to the *Academic Rank*

Academic Rank	Frequencies	Percentage(%)
Assistant Prof	5	19.2
Associate Prof	8	30.8
Prof	4	15.4
Other	9	34.6
Total	26	100.0

Table 5.3: The distribution of the study sample according to the *Number of Publications*

Number of Publication	Frequencies	Percentage(%)
Less than 10	6	23.1
Greater than 10 and Less than 30	17	65.4
Greater than 30	3	11.5
Total	26	100.0

The Study tool: Based on the Semantic Web background research and earlier studies, we built a questionnaire to collect data from the Semantic Web researchers, the study tool included (12) questions distributed on four pivots as in Table (5.4)

Table 5.4: The study tool depending on the Pivots of the study.

	The Pivots	Number of Questions	The Questions
1	D1: The importance of the need for this framework.	3	3-1
2	D2: The extent of the problems that justifies the existence of such framework.	4	7-4
3	D3: The ability of the framework to solve these problems.	5	12-8
	Total	12	

The stability of the tool: To verify the stability of the study tool we used the Cronbach alpha formula to extract the stability, Ratio amounted to (0.84) which is stable confirm the possibility of using the tool.

Statistical treatment: After collecting the data we entered it into the Statistical Program for Social Sciences (SPSS), we used percentages and weighted averages and ANOVA test.

The Study results:

**First:** The results for the pivots of the study:

We have been given to the questions with positive content (5) degrees for each answer (Totally Agree), and (4) degrees for each answer (Partially Agree), and (3) degrees for each answer (Neither Agree or Disagree), and two degrees for each answer (Partially Disagree), degree and one for each answer (Totally Disagree), and for the interpretation of results we adopted the following balance of the percentages of responses:

Table 5.5: Balance of the percentages of responses

The degree of responses	Percentage
Very low	Less than 50%
Low	- %50.9-%59
Medium	%69.9 - %60
High	%79.9 - %70
Very high	Than 80% and above

The tables (5.6), (5.7), (5.8) show the results.

1) The results for the first pivot (The importance of the need for this framework)

Table 5.6: Averages and percentages of the first pivot

Number	Questions	Averages	Percentages	Degree of Responses
1	The researchers have assumed closed domains and manageable size of web to apply their semantic web researches.	3.92	78.46	High
2	The modeling languages that are used for describing the Web domains like the RDF needs time to be learned by the researchers and need more standardization.	4.15	83.08	Very high
3	The Semantic Web implementation aims to convert the web into a managed database.	3.62	72.31	High
	Total	3.90	77.95	High

Shown in the previous table (5.6), The responses were very high in question (2), where the percentage is above (80%) and high (1,3), where the percentage of ( 70% -79.9%) and the total responses are high in terms of the percentage (77.95%)

2) The results for the second pivot (The extent of the problems that justifies the existence of such framework)

Table 5.7: Averages and percentages of the second pivot

Number	Questions	Averages	Percentages	Degree of Responses
4	I think that the assuming of closed domains and manageable size of web when applying the semantic web researches leads to the lack of standardization and collaboration between researchers.	3.69	73.85	High
5	I think that the presence of specific standards to describe the websites types will achieve the primary goal of Semantic Web.	3.88	77.69	High
6	I think that the web is not perceived yet as a database and interrelated.	4.08	81.54	Very high
7	I think there is need for the semantic web framework to direct the researchers work to it.	3.96	79.23	High
	Total	3.90	78.08	High

Shown in the previous table (5.7), The responses were very high in question (6), where the percentage is above (80%), and high (4, 5, 7), where the percentage of ( 70% -79.9%) and the total responses are high in terms of the percentage (78.08%)

3) The results for the third pivot (The ability of the framework to solve these problems)

Table 5.8: Averages and percentages of the third pivot

Number	Questions	Averages	Percentages	Degree of Responses
8	I think the researchers in the semantic web field need to apply their works in a way to reach an integrated work.	4.08	81.54	Very high
9	I think that the existence of a framework that contains automatic RDF generator will facilitate the researches work and the rapid evolution of the semantic web	3.85	76.92	High
10	I think that the framework presence that integrates the researcher's works in the semantic web field achieves the perception of the database.	3.77	75.38	High
11	I believe that a framework like this will be a new start in the semantic web field and in its progress.	3.96	79.23	High
12	I believe that a framework like this will lead to move the semantic web from theories to technology.	3.96	79.23	High
	Total	3.92	78.46	High

Shown in the previous table (5.8), the responses were very high in question (8), where the percentage is above (80%), and high (9, 10, 11, 12), where the percentage of (70% -79.9%) and the total responses are high in terms of the percentage (78.46%)

**Second:** The results of the statistical hypothesis

1) The Results for the first hypothesis:

There is no statistically significant differences at the level of ( $\alpha= 0.05$ ) in the responses of the researchers in the Semantic Web field to a variable of Academic Rank.

To test the hypothesis we used (ANOVA) test, and table (5.9) shows the results:

Table 5.9: The results of (ANOVA) test to Academic Rank variable

The Pivots	Assistant Prof		Associate Prof		Prof		Other		(f)	Sig
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev		
D1	3.47	1.48	3.79	0.50	4.50	0.33	3.96	0.72	1.23	0.32
D2	3.65	1.08	3.53	0.75	4.25	0.54	4.22	0.65	1.60	0.22
D3	3.60	1.28	3.45	1.08	3.75	0.97	4.60	0.33	2.61	0.08
Total	3.57	1.28	3.59	0.78	4.17	0.61	4.26	0.57	1.81	0.21

Shown in the previous table No. (5.9) there is no statistically significant differences at level ( $\alpha = 0.05$ ) in the responses to the variables of Academic Rank on all study pivots where the level of significance of the values of (T) was greater than that (0.05) and thus accept the hypothesis.

2) The Results for the first hypothesis:

There is no statistically significant differences at the level of ( $\alpha= 0.05$ ) in the responses of the researchers in the Semantic Web field to a variable of Number of Publications.

To test the hypothesis we used (ANOVA) test, and table (5.10) shows the results:

Table 5.10: The results of (ANOVA) test to Number of Publications variable

The Pivots	Less than 10		Greater than 10 and Less than 30		Greater than 30		(f)	Sig
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev		
D1	3.94	1.56	3.90	0.40	3.78	1.17	0.05	0.96
D2	4.04	1.21	3.85	0.60	3.92	1.13	0.12	0.89
D3	4.27	1.18	3.86	0.93	3.60	1.25	0.53	0.60
Total	4.08	1.31	3.87	0.64	3.76	1.18	0.23	0.82

Shown in the previous table (5.10) there is no statistically significant differences at level ( $\alpha = 0.05$ ) in the responses to the variables of Number of Publications on all study pivots where the level of significance of the values of (T) was greater than that (0.05) and thus accept the hypothesis.



## Chapter 6

### Conclusion and Future Works

#### 6.1 Conclusion

In this research we have solved some outstanding research problems in the Semantic Web area, the most important closed domains which assumed by the researchers before embarking on their research, and the difficulty of use of the modeling languages which used by the Semantic Web techniques. In addition to make the Semantic Web vision applicable, all of the above by introducing the Semantic Web Framework (SWF) which is a system for creating, Querying, and sorting the RDF document.

SWF makes the semantic web more general and standard by introducing a new system for generating the RDF document automatically using prepared models for each domain. RDF generator presents a domain-oriented approach to automatic web data extraction. This approach is based on a highly efficient patterns structure analysis and allows the extraction of relevant text passages from the pages of a given Web site, fetching the entire Web site content, and the extraction of the relevant text passages discarding non-useful data. RDF generator used by users or other websites like search engines.

To make the SWF more general and familiar we have used modified SQL for querying the RDF documents, it uses a linguistic analyzer to enable the users to write what they want and the system analyzes and understands the written query. We propose three type of SQL for our system to request the data from RDF documents, to request RDF document, and to refresh RDF document.

To facilitate the search engines work; the SWF proposed a new system for ranking web pages based on the page content and the RDF document history. SWF ranking system introduce a new formula to calculate the page rank using parameters to keep the rank reasonable between old and new rank and to help in identifying the site rank in terms of contents. The Rank parameter has been added within the RDF document description part.

The SWF system has been designed carefully to test it in reality; we have implemented a web application using Microsoft C#.Net 2008 and Microsoft SQL Server 2008. We have chosen the educational domain as a case study to test the framework use cases.

We have tested the SWF parts manually to verify their usefulness and then verified the views of researchers in the world. Manually we take a sample of three RDF for universities websites that was automatically created using the RDF generator. And we create these RDF documents manually by accessing the websites and selecting the required data and then put it in the RDF file. The results were manually evaluated and by separating the matching to find accurate time, the result was that it provided a rate of 16 minutes in the process of creating each file. From the view of researchers after answering the paragraphs relating to the importance of the need for this framework; the responses average was 77.95% after answering the paragraphs relating to the extent of the problem that justifies the existence of such framework; the responses average was 78.08%, and after answering the paragraphs relating to the ability of the framework to solve these problems; the responses average was 78.46%.

## **6.2 Recommendations and Future Works**

We have developed this framework to accelerate the development of the Semantic Web. All researchers in the Semantic Web area are invited to direct their research to this framework to be as main Semantic Web Framework.

For future, we suggest the following ideas:

- Adding more semantics to the knowledge base and in different languages to be more general.
- Designing all RDF models in the implementation, we can take each model as research to find the best model and best semantic for it.
- Adding Draw graphs to view RDF relations.
- Extracting a description for each file (pdf, doc, ppt, and pictures...) in the website and adding it to file property in the RDF models.

## References

- AbuZir, Y. and Nada, A. (2006): "Researchers Search Engine On The WEB -RSEWEB and an Application". Proceedings of the 2006 International Arab Conference on Information Technology (ACIT'2006).
- Aron S. (2002): "The Semantic Web in Breadth". ([http://logicerror.com/semantic Web-long](http://logicerror.com/semantic%20Web-long), 21.9.2009)
- Adams T., et al. (2005): "Kowari: A Platform for Semantic Web Storage and Analysis". (<http://www.itee.uq.edu.au/~dwood/docs/www2005-kowari.pdf>, 21.9.2009)
- Aduna (2007): "Sesame". (<http://www.openrdf.org/>, 20-9-2009).
- Ankolekar A., et al. (2008): "The Two Cultures: Mashing up web 2.0 and the semantic web". Journal of Web Semantics 6, 70-75.
- Antoniou, et al. (2008): "A Semantic Web Primer", 2<sup>nd</sup> Edition. Cambridge, Massachusetts: The MIT Press.
- Aurona A., et al. (2008): "A functional Semantic Web Architecture". 5th European Semantic Web Conference (ESWC2008).
- Beckett D. (2005): "Redland". (<http://librdf.org/>, 20.9.2009).
- Beckett D. (2001): "The design and implementation of the Redland RDF application framework". Proceedings of the tenth international conference on World Wide Web, Hong Kong.
- Bojars U., et al. (2008): "Using the Semantic Web for Linking and reusing data across Web 2.0 communities". Journal of Web Semantics 6, 21-28.
- Carroll J., Dickinson I., et al. (2004): "Jena: implementing the semantic web recommendations". Proceedings of the 13th international World Wide Web conference.

Chidlovskii B. (2002): Automatic repairing of Web wrappers by combining redundant views. In Proceedings of IEEE Conf. Tools with AI.

Delbru R., et al. (2008): "A Document-oriented Lookup Index for Open Linked Data". Proceedings of the 5th European Semantic Web Conference (ESWC2008).

Dillon A., Gushrowski B, (2000): "Genres and the Web: Is the personal home page the first uniquely digital genre? ". Journal of the American Society for Information Science.

Ding , et al. (2004): "Swoogle: A Search and Metadata Engine for the Semantic Web". Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management.

Elliott B. (2008): "Hierarchical And Semantic Data Management And Querying For Patient Records And Personal Photos". Case Western Reserve University.

Embley D., et al. (1998): Ontology-based extraction and structuring of information from data-rich unstructured documents. In Proceedings of the Conf. on Info. and Knowledge Management.

Feigenbaum L. , et al. (2007): "Boca: an open-source RDF store for building Semantic Web applications". Briefings in Bioinformatics.

Gregg, D. and Walczak, S. (2006): Exploiting the Information Web. IEEE Trans. on System, Man and Cybernetics Part C.

Guha R.V. (2007): "rdfDB : An RDF Database". ( <http://www.guha.com/rdfdb/>, 3.10.2009)

Harman K., Koohang A. (2005): "Discussion Board: A Learning Object", Interdisciplinary Journal of Knowledge and Learning Objects Volume 1.

Hewlett-Packard (2009): "Jena – A Semantic Web Framework for Java ". (<http://jena.sourceforge.net/> , 20.9.2009).

Hogan A., et al. (2006): "ReConRank: A Scalable Ranking Method for Semantic Web Data with Context". In 2nd Workshop on Scalable Semantic Web Knowledge Base Systems.

Immaneni T. (2007): "A Hybrid Approach to Retrieving Web Documents and Semantic Web Data". Wright State University, Wright.

Jain k. (2007): "D-RDF: Dynamic Resource Description Framework". Iowa State University, Ames, Iowa.

KAON (2005): "KAON Tool Suite Homepage". (<http://kaon.semanticweb.org/>, 5.10.2009)

Kendall C. (2005): "RDF Data Access Use Cases and Requirements". W3C Working Draft, World Wide Web Consortium.

Kittur A., et al. (2009): "What's in Wikipedia? Mapping Topics and Conflict Using Socially Annotated Category Structure", CHI 2009.

Klyne G., et al. (2004): "Resource Description Framework (RDF): Concepts and Abstract Syntax". W3C Recommendation, 10 February 2004. Latest version: <http://www.w3.org/TR/rdf-concepts/>

Knoblock, et al. (2000): Accurately and reliably extracting data from the Web: A machine learning approach. Bulletin IEEE Computer Society Technical Committee on Data Engineering.

Laender A., et al. (2002): Surveys: A brief survey of web data extraction tools. ACM SIGMOD Record 31.

LECKNER S., APPELGREN E. (2007): "E-paper News Publishing Strategies for Product and Production", Nordicom Review 28.

Llopis J., et al. (2009): "Identifying Corporate Managerial Topics with Web Pages". International Journal of Social Sciences 4:3.

Lopez X., et al. (2007): "Semantic Data Integration for the Enterprise". Oracle Corporation.

Lorio A.D., et al. (2008): "Wiki Content Templating". Proceedings of the International World Wide Web Conference- 2008.

Mäkelä E., et al. (2004): "ONTOVIEWS A Tool for Creating Semantic Web Portals". The Semantic Web – ISWC 2004.

Marut B. (1994): "A Framework for The Organization and Discovery of Information Resources in A WWW Environment Using Association, Classification and Decuction". King Mongkut's Institute of Technology LadKrabang, Thailand.

Muthaiyah S. (2008): "A Framework and Methodology for Ontology Mediation through Semantic and Syntactic Mapping". The Volgenau School of Information Technology and Engineering, Volgenau.

Oracle (2009): "Oracle Semantic Technologies Center". ([http://www.oracle.com/technology/tech/semantic\\_technologies/](http://www.oracle.com/technology/tech/semantic_technologies/), 20.9.2009).

Park S. (1995): "A Methodological Framework for Automated Support to Text Based Identification of System Model Elements". George Mason University, Virginia.

Punin j. (2003): "WWWPal Suite For Analysis And Organization for Web Sites". Rensselaer Polttechnic Institute, New York.

Ramakrishnan C. (2008): "Extracting, Representing and Mining Semantic Metadata from Text: Facilitating Knowledge Discovery in Biomedicine". Doctoral dissertation, Wright State University.

RDFStore (2006): "Perl/C RDF storage and API". (<http://rdfstore.sourceforge.net/>, 1.11.2009)

Sesame (2009): "Sesame Homepage". (<http://www.openrdf.org>, 5.10.2009)

Thij G., et al. (2007): "A framework for exploring relationships between online community characteristics and regulation principles". Proceedings of the BIS 2007 Workshop on Social Aspects of the Web Poznan, Poland.

Thomas R. (2008): "The Semantic Web, Service Oriented Computing, And The Standards That Bind Them". The Graduate School of The College of Charleston and Citadel.

Tribunella T. (2001): "Linking Organizational Strategy to E-Commerce". International Conference Of the Academy of Business and Administrative Sciences (ABAS) Quebec City, Canada.

Volz R., Oberle D., et al. (2003): "KAON SERVER – A Semantic Web Management System". Proceedings of the Twelfth International World Wide Web Conference WWW2003, Budapest.

W3C (1998): "Extensible Markup Language (XML) 1.0". (<http://www.w3.org/TR/1998/REC-xml-19980210>, 20.9.2009)

W3C (2004): "OWL Web Ontology Language". (<http://www.w3.org/TR/owl-features/>, 22-11-2009)

W3C (2004): "Resource Description Framework (RDF) ". (<http://www.w3.org/RDF/> 1.7.2009).

W3C (2007): "Semantic Web Layers cake". <http://www.w3.org/2007/03/layerCake.png>, 20.9.2009)

W3C (2008): "SPARQL Query Language for RDF". (<http://www.w3.org/TR/rdf-sparql-query/>, 21.9.2009).

W3C (2009): "OWL 2 Web Ontology Language ". (<http://www.w3.org/TR/owl2-overview/>, 30-11-2009)

W3C (2009): "Semantic Web" . (<http://www.w3.org/standards/semanticweb/>, 3.6.2009).

Yoo Jung An (2008): "Ontology Learning for the Semantic Deep Web". Faculty of New Jersey Institute of Technology, New Jersey.

## Appendices

### Appendix 1: An Example of RDF Document for Alquds.edu

The following scheme shows Al-Quds University website which discussed in the case study (see chapter 5). This case study has been submitted to evaluate our work and might become a W3C Working Group.

To get the following RDF document you can open this link:

<http://www.nadasys.com/swf/rdfg.aspx?sql=select RDF from web where site=www.alquds.edu>

```
<?xml version="1.0" encoding="utf-8"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">

    <rdf:Description rdf:about="http://www.alquds.edu">
      <rdf:domain>Universities and Educational</rdf:domain>
      <rdf:rank>0.0</rdf:rank>
      <rdf:RDFdate>21-12-2009</rdf:RDFdate>
      <rdf:title>:: Al-Quds University :: The Arab University in Jerusalem ::</rdf:title>
      <rdf:preperationcomments>non</rdf: preperationcomments >
      <rdf:programerscomments>non</rdf: programerscomments >
      <rdf:keywords>quds, alquds, al-quds, university, jerusalem, palestine, academic, international,
program, conference, student, institute, faculty, medicine, west bank, middle east, arab, community,
department, science, math, mathematics, physics, chemical, technology, art, ramallah, beth hanina, abu
dies, calendar, degrees, time line, centers, lectureship, summer, admission, planning, strategy,
initiation, society, graduate, course, public, health, law, school, young, profession, development,
primary, care, research, library, learning, leadership, area, studies, modern media, semester, academic
network , planet, speech, language, business, economics, refugee, peace, traumatic, stress, old city,
president, admin, administration, teaching, arabic, international, credits, exchange, learn, teach, ÌÇãÚÉ
ÇãĐĬÓ, ÇÈæĬÓ</rdf:keywords>
```



<rdf:about>Al-Quds University is The Only Arab University in Jerusalem, it was established in 1984. This Page Cover All Informations About Academic, International, Conferences, Public Relations, and News.</rdf:about>

<rdf:contact>P.O. Box 51000 Telephones: +(972)-(2)-5838652, 5838654, Fax: +(972)-(2)-</rdf:contact>

<rdf:mainparts>

<rdf:part rdf:resource="http://student.alquds.edu/loginsp.pro">Students</rdf:part>

<rdf:part rdf:resource="http://employee.alquds.edu/login.pro">Staff</rdf:part>

<rdf:part rdf:resource="http://eclass.alquds.edu/">E-Class</rdf:part>

<rdf:part rdf:resource="http://egroup.alquds.edu/">E-Group</rdf:part>

<rdf:part rdf:resource="http://mail.alquds.edu/cgi-bin/openwebmail/openwebmail.pl">Mail</rdf:part>

<rdf:part

rdf:resource="http://www.alquds.edu/events&activities/index\_ar.php">News</rdf:part>

</rdf:mainparts>

<rdf:links>

<rdf:link rdf:resource="http://student.alquds.edu/loginsp.pro">Students</rdf:link>

<rdf:link rdf:resource="http://employee.alquds.edu/login.pro">Staff</rdf:link>

.....

<rdf:link rdf:resource="http://www.itce.alquds.edu/">SKITCE</rdf:link>

</rdf:links>

<rdf:files>

<rdf:file rdf:resouce="/images/main/university-logo.gif">Al-Quds University</rdf:file>

<rdf:file rdf:resouce="//images/main/new/sprofile.gif">Student Profile</rdf:file>

.....

```

    <rdf:file rdf:resouce="/images/more.gif">More News</rdf:file>
  </rdf:files>

</rdf:Description>

<rdf:Faculties rdf:about="">

  <rdf:Faculty rdf:ID="Faculty1" rdf:about=
"http://www.alquds.edu/faculties/economic/index.php">
    <rdf:title>Admin. & Economic Science</rdf:title>
    <rdf:contact>Tel: 2799360 / 2799502 Fax: 00972-2-2796960 </rdf:contact>
    <rdf:text>The Dean's Word: Our mission in the college of economics and Business is to
prepare individuals to function effectively in a dynamic and complex global business and economic
environment. The college provides academic and professional education to enable individuals,
successfully to pursue their career paths and to make effective contribution to their communities and
country at large.The faculty is making a sustained commitment to providing students a demanding and
relevant economic and business curriculum, designed to meet the diverse and changing needs of the
economy.The College offers majors in Accounting, Banking and Finance, Business Administration,
Economics and Marketing. Minors are also offered in Said majors.Our main objective is to produce a
policy maker and a decision taker individual.</rdf:text>
  </rdf:Faculty>

.....

  <rdf:Faculty rdf:ID="Faculty12"
rdf:about="http://www.alquds.edu/faculties/science/index.php">
    <rdf:title>Science & Technology</rdf:title>
    <rdf:contact></rdf:contact>
    <rdf:text>The idea of establishing an institution of higher learning in the outskirts of
Jerusalem was conceived early in 1957, when a Board of Trustees was formed in Kuwait. The Board
elected an Executive Committee entrusted with bringing this grand project into being. </rdf:text>
  </rdf:Faculty>

</rdf:Faculties>

```

```

<rdf:persons rdf:about="">
  <foaf:Person rdf:ID="person1">
    <foaf:name>Mohammed Khaled I. Bader </foaf:name>
    <foaf:title>PhD</foaf:title>
    <foaf:mbox rdf:resource="mailto:mbader@econ.alquds.edu " />
    <foaf:homepage rdf:resource="http://alquds.edu/~f3102"/>
    <foaf:workmajor>Admin. and Economics Sciences</foaf:workmajor>
    <foaf:workingin rdf:ID="Faculty1"/>
  </foaf:Person>

.....

<foaf:Person rdf:ID=" person123">
  <foaf:name>Badie Sartawi</foaf:name>
  <foaf:title>Dr.</foaf:title>
  <foaf:mbox rdf:resource="" />
  <foaf:homepage rdf:resource=""/>
  <foaf:workmajor>Engineering Science (System Theory & Engineering)</foaf:workmajor>
  <foaf:workingin rdf:ID="Faculty12"/>
</foaf:Person>
</rdf:persons>

```

The following is the [www.alquds.edu](http://www.alquds.edu) RDF Graph



## **Appendix 2: W3C Document License**

Public documents on the W3C site are provided by the copyright holders under the following license.

### License

By using and/or copying this document or the W3C document from which this statement is linked, you (the licensee) agrees that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

- A link or URL to the original W3C document.
- The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright © [\$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>"
- If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

### Disclaimers

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR

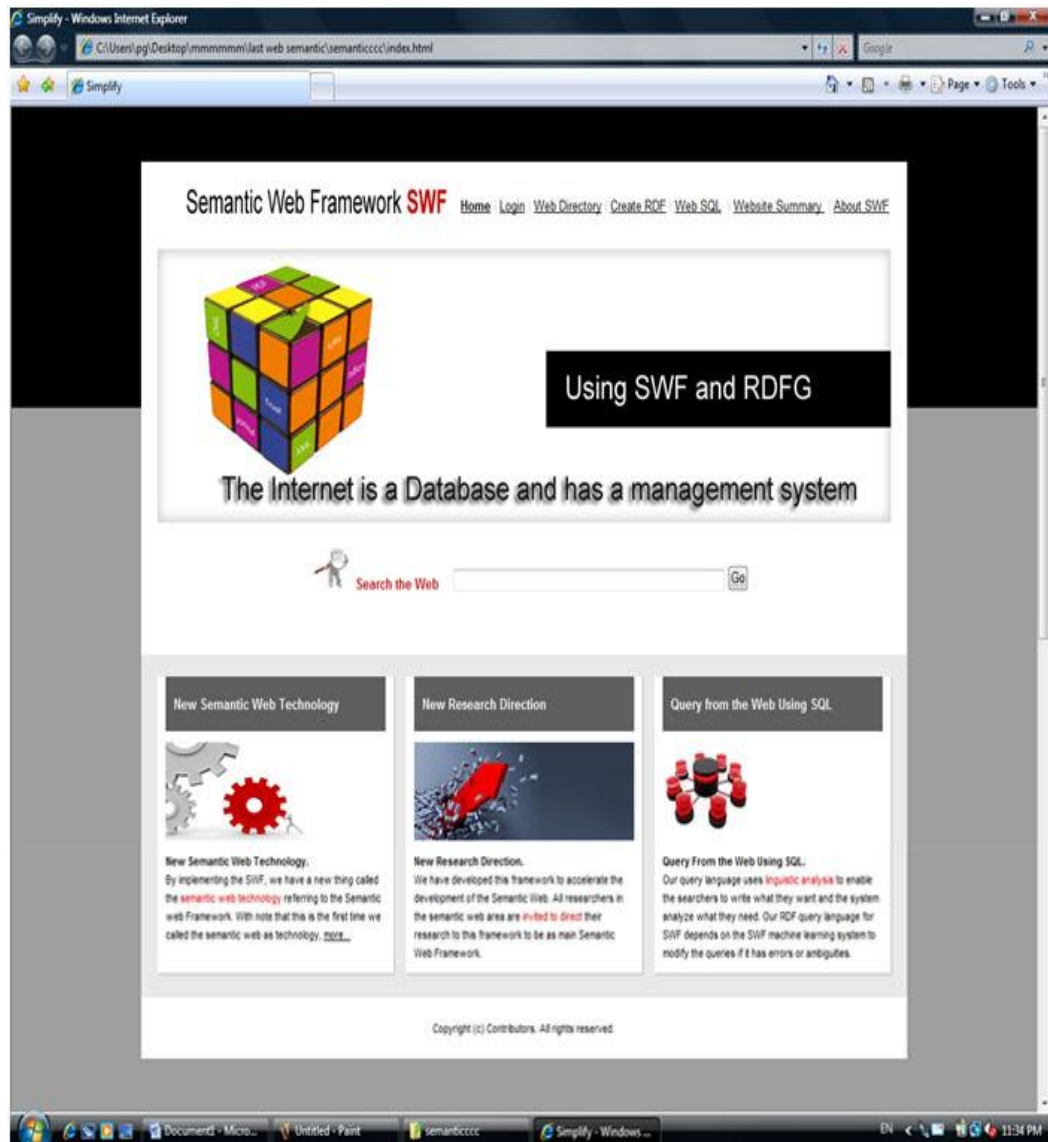
PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

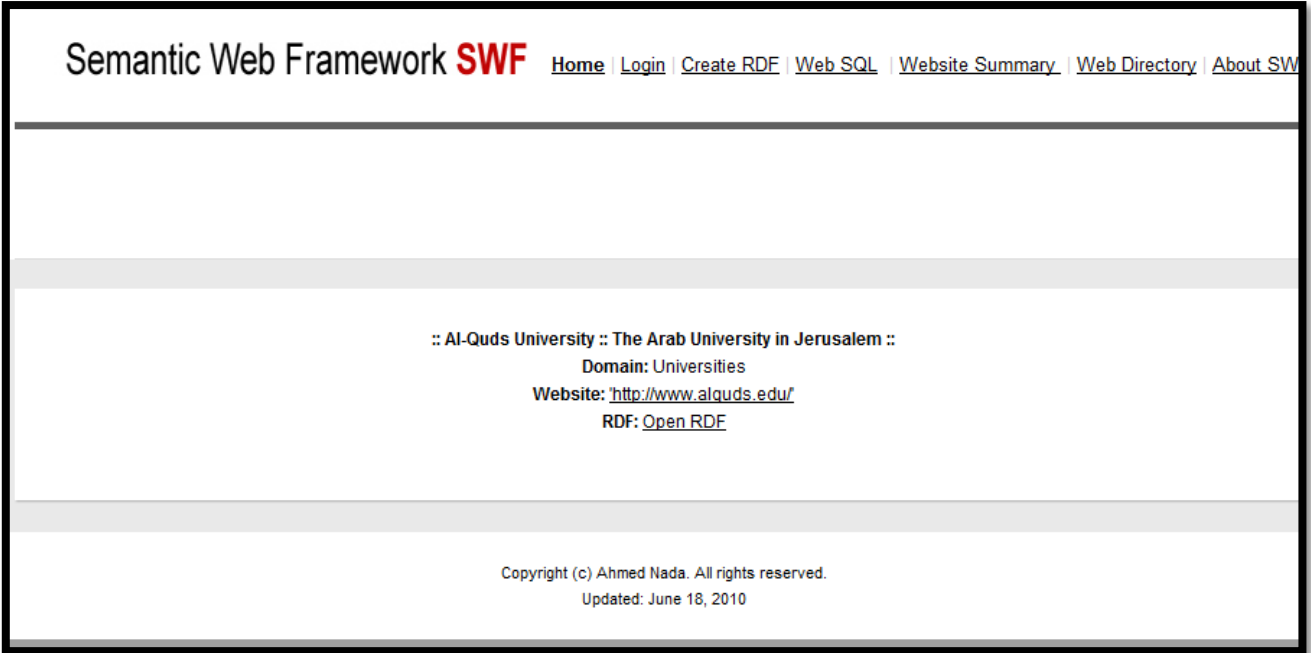
## Appendix 3: SWF User Manual

### SWF Homepage

The next page is the SWF online application homepage, it include search functionality, that provide brief websites description



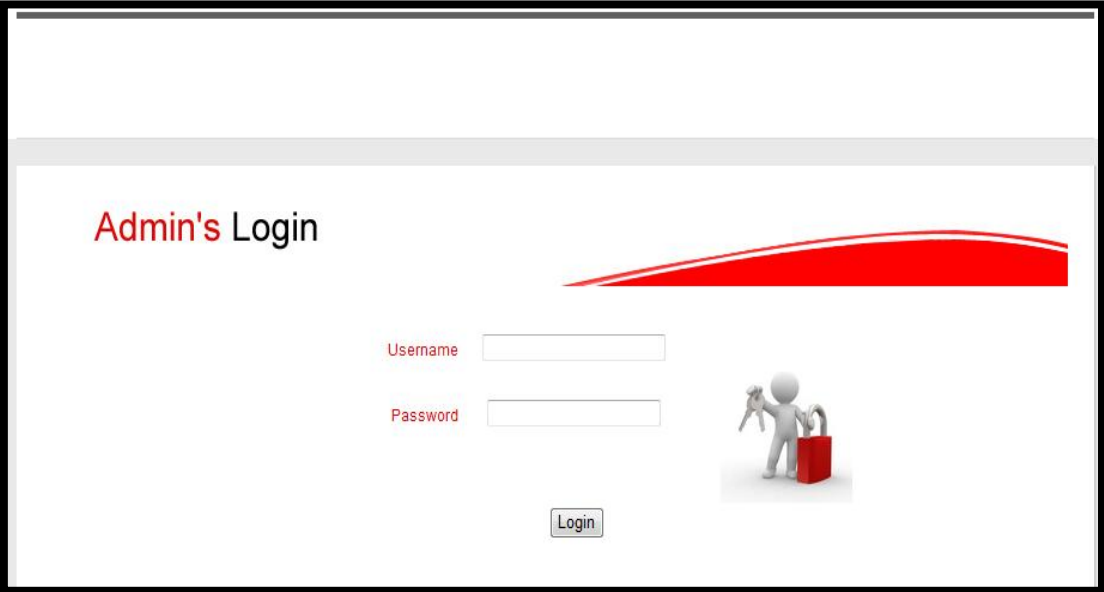
When writing the website URL such as “http://www.alquds.edu/” in the search box the next page appears:



**Admin Control Panel Web Page**

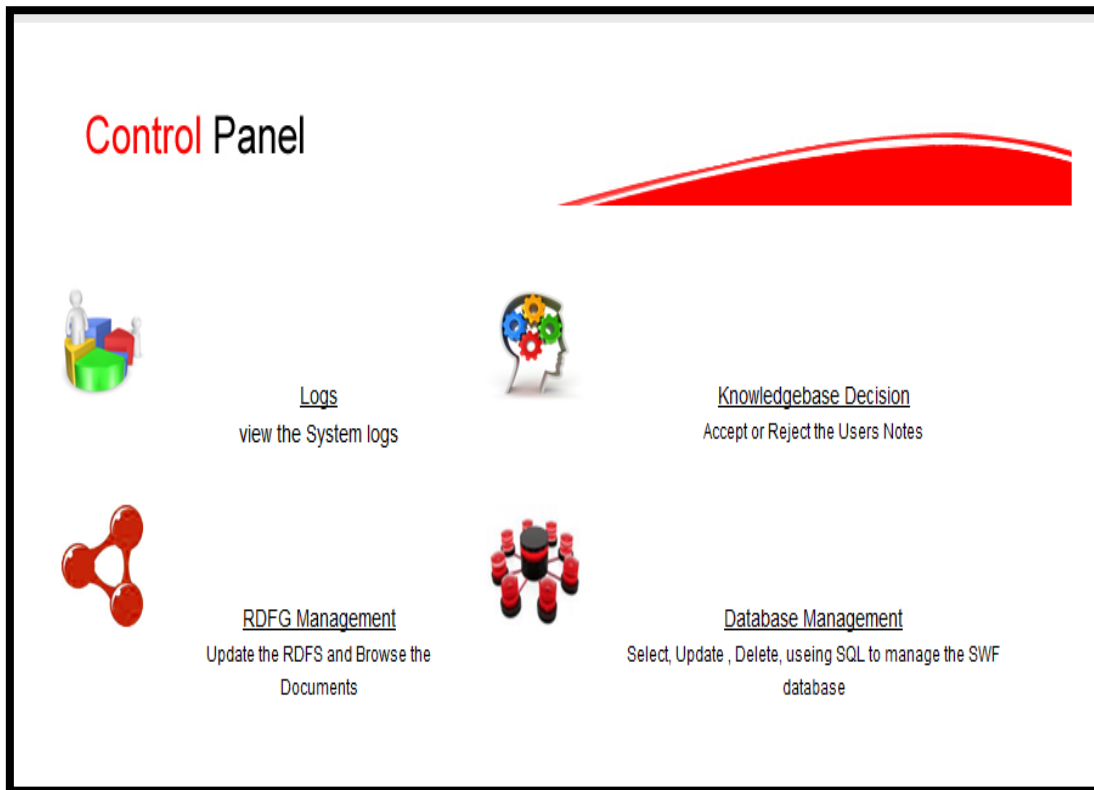
The admin can access his page through the login page that appears next, the admin must enter:

- His user name in the user name box
- His password in the password box





The login process leads to the Control Panel Web Page as appears next:

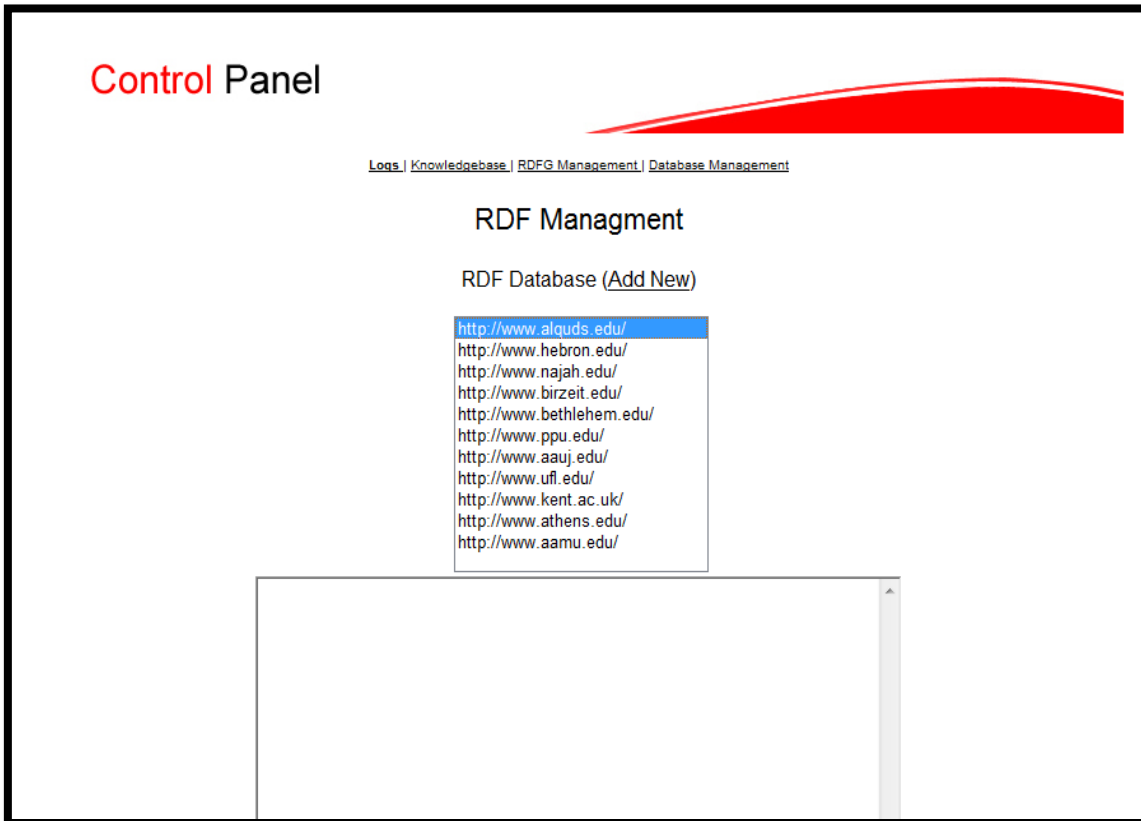


When choosing the Logs link the next page is loaded, this page includes all the users actions that are don on the SWF website:

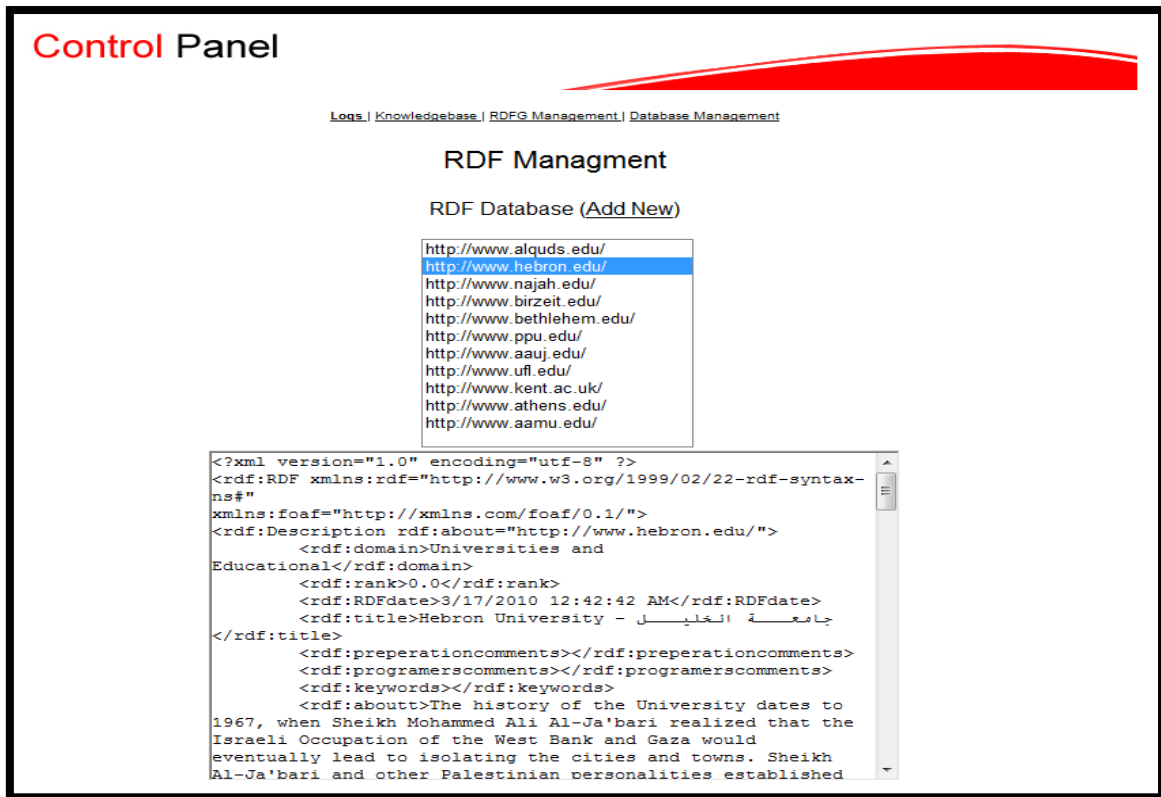
The screenshot shows the 'The System Logs' page. At the top, the 'Control Panel' title is visible. Below it, a navigation bar contains links for [Logs](#), [Knowledgebase](#), [RDFG Management](#), and [Database Management](#). The main heading is 'The System Logs'. Below the heading is a table with the following data:

ID	Event	User_ip	At_time
773	Open Logs Page	195.114.29.1	6/19/2010 5:05:20 AM
772	Open Control Panel	195.114.29.1	6/19/2010 5:04:23 AM
771	Login to Control Panel using admin	195.114.29.1	6/19/2010 5:04:22 AM
770	Open Control Panel Login	195.114.29.1	6/19/2010 5:04:22 AM
769	Open Control Panel Login	195.114.29.1	6/19/2010 5:04:16 AM
768	Open Logs Page	213.6.238.85	6/17/2010 5:57:31 PM
767	Open Control Panel	213.6.238.85	6/17/2010 5:57:29 PM
766	Login to Control Panel using admin	213.6.238.85	6/17/2010 5:57:28 PM
765	Open Control Panel Login	213.6.238.85	6/17/2010 5:57:28 PM
764	Open Control Panel Login	213.6.238.85	6/17/2010 5:57:21 PM
763	Open RDF SQL Page	213.6.238.85	6/17/2010 5:54:56 PM
762	Open RDF SQL Page	213.6.238.85	6/17/2010 5:53:51 PM
761	Error While execute SQL: The SQL syntax error	213.6.238.85	6/17/2010 5:53:38 PM

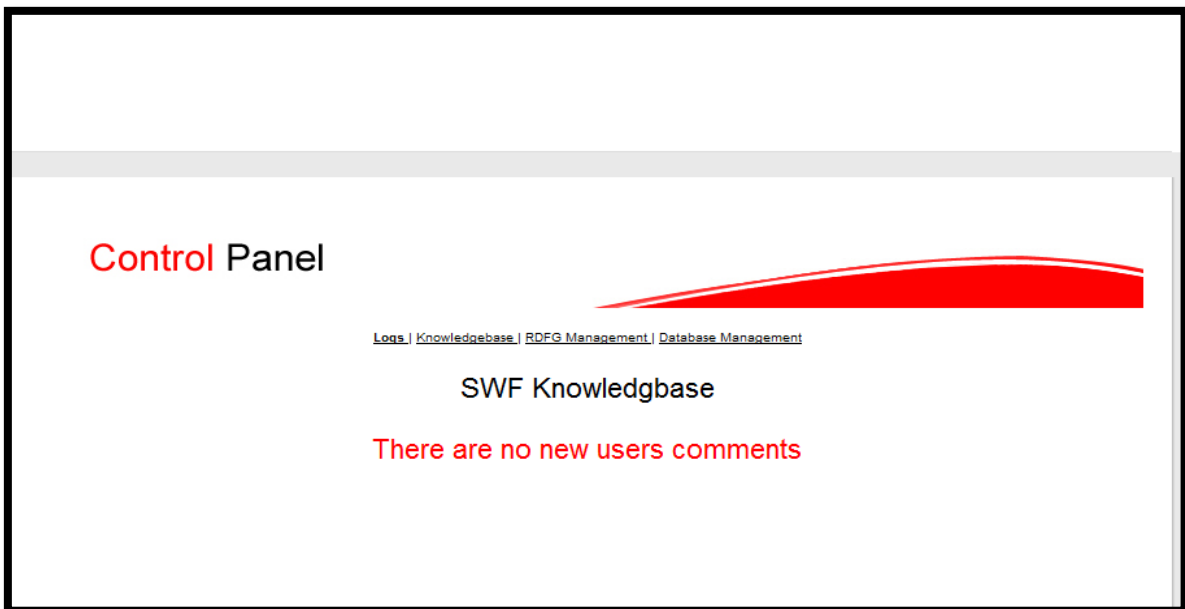
When choosing the RDFG Management link the next page is loaded, this page includes the generated and saved RDF documents:



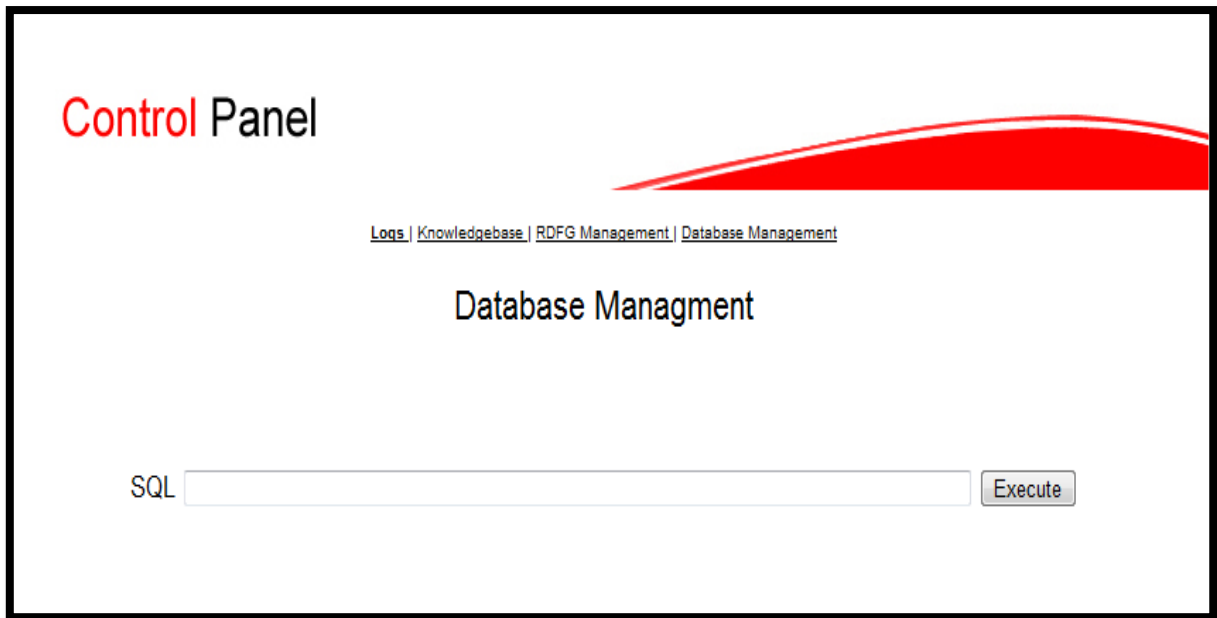
When choosing any item in the list the RDF document appears in the text area as follows:



When choosing the Knowledgebase Decision link the next page is loaded, this includes the added user notes:



When choosing the Database Management link the next page is loaded, this pages enables the admin to execute the SQL statements:



### **RDF Generation Web Page**

The next screen shows the RDF creation page, where the users write the website URL and get the automatic RDF document:

# Create RDF

Website URL

```


<rdf:RDF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="http://www.alquds.edu/">
    <rdf:domain>Universities and Educational</rdf:domain>
    <rdf:rank>0.0002</rdf:rank>
    <rdf:RDFdate>6/17/2010 5:56:35 PM</rdf:RDFdate>
    <rdf:title:: Al-Quds University :: The Arab University in Jerusalem ::</rdf:title>
    <rdf:preparationcomments />
    <rdf:programerscomments />
    <rdf:keywords>quds, alquds, al-quds, university, jerusalem, palestine, academic,
international, program, conference, student, institute, faculty, medicine, west bank, middle
east, arab, community, department, science, math, mathematics, physics, chemical, technology,
art, ramallah, beth hanina, abu dies, calendar, degrees, time line, centers, lectureship,
summer, admission, planning, strategy, initiation, society, graduate, course, public, health,
law, school, young, profession, development, primary, care, research, library, learning,
leadership, area, studies, modern media, semester, academic network , planet, speech,
language, business, economics, refugee, peace, traumatic, stress, old city, president, admin,
administration, teaching, arabic, international, credits, exchange, learn, teach</rdf:keywords>
    <rdf:aboutt>Al-Quds University is The Only Arab University in Jerusalem, it was
established in 1984. This Page Cover All Informations About Academic, International,
Conferences, Public Relations, and News.</rdf:aboutt>
    <rdf:contact> Administrative Offices: Abdel Hamaid Shoman Street, Beit Hanina - Jerusalem,
P.O. Box 51000 Telephones: +(972)-(2)-5838652, 5838654, Fax: +(972)-(2)-5838653.</rdf:contact>
    <rdf:mainparts>
      <rdf:part rdf:resource="http://www.alquds.edu/server-message.php">Student</rdf:part>
      <rdf:part rdf:resource="http://student.alquds.edu/loginsp.pro">Student</rdf:part>
      <rdf:part rdf:resource="http://www.alquds.edu/server-message.php">Staff</rdf:part>
      <rdf:part rdf:resource="http://employee.alquds.edu/login.pro">Staff</rdf:part>

```

## Web SQL Web Page

The next screen shows the Web SQL page, where the users write the SQL query (including the select RDF and refresh RDF statement):

# Create RDF



**Enter SQL**


```
select about from web where
site=http://www.alquds.edu/
```

```
Al-Quds University is The Only Arab University in Jerusalem, it was established in 1984. This
Page Cover All Informations About Academic, International, Conferences, Public Relations, and
News.
```

**Web SQL Web Page**

The next screen shows the Web Summary page, where the users write the website URL and the application summarizes it:

# Website Summary



**Website URL**

**:: Al-Quds University :: The Arab University in Jerusalem ::**

Al-Quds University is The Only Arab University in Jerusalem, it was established in 1984. This Page Cover All Informations About Academic, International, Conferences, Public Relations, and News.

quds, alquds, al-quds, university, jerusalem, palestine, academic, international, program, conference, student, institute, faculty, medicine, west bank, middle east, arab, community, department, science, math, mathematics, physics, chemical, technology, art, ramallah, beth hanina, abu dies, calendar, degrees, time line, centers, lectureship, summer, admission, planning, strategy, initiation, society, graduate, course, public, health, law, school, young, profession, development, primary, care, research, library, learning, leadership, area, studies, modern media, semester, academic network, planet, speech, language, business, economics, refugee, peace, traumatic, stress, old city, president, admin, administration, teaching, arabic, international, credits, exchange, learn, teach

Administrative Offices: Abdel Hamaid Shoman Street, Beit Hanina - Jerusalem, P.O. Box 51000 Telephones: +(972)-(2)-5838652, 5838654, Fax: +(972)-(2)-5838653.

## Web Directory Web Page

This page includes websites directories, when choosing any directory the lists of websites classified in chosen directory are displayed, as shown below:



## About SWF Web Page

This page includes information about the online SWF application, the webpage is displayed next:

## About SWF

For many years, the Semantic Web has been little more than a dream. It began as a vision put by World-Wide-Web creator Tim Berners-Lee and was built as new and untested technology standards. The Semantic Web has promised solutions to the challenges of huge data integration structured and directed search and knowledge management.

This thesis is an attempt to make the Berners-Lee vision applicable by gathering researchers' participations in the Semantic Web field and introduce it in one framework called The Semantic Web Framework (SWF). The Semantic Web Framework is a system for creating, Querying, and sorting the RDF document. SWF introduces a new system for generating the RDF document using a web service to build the RDF documents automatically using prepared models for each domain, this service used by users or other websites like search engines. SWF uses modified SQL for querying the RDF documents and also it is a web service used by users or search engines; this web service sorts the results using ranking parameter in each RDF document.

This thesis aims to solve some problems in the Semantic Web research area, the most important of these problems are to avoid the closed domains problem, the complexity of modeling to use the Semantic Web by programmer without the need to be trained in modeling languages, and realize the entity creating, sharing and reusing Metadata on the Web. These issues are solved in this thesis by introducing the Semantic Web Framework to manage the web and put general standards to use the Semantic Web; we implement our framework and it is available on (<http://www.nadasys.com/swf/>).



## Developers Codes

- Generate RDF documents for websites.

The developers can create RDF for websites automatically using RDFS. Web developers can use the RDFG by adding the website URL after the RDFG webpage URL. For example if a programmer wants to create or retrieve RDF for [www.alquds.edu](http://www.alquds.edu) he can request the following link to prepare it

“<http://www.nadasys.com/swf/RDFG.aspx?URL=www.alquds.edu>” the developers can request

- Execute SQL statement on the Web.

As we mentioned in the RDF generator service, the users and developers can execute the SQL statements using two ways; fill the SQL in the form at (<http://www.nadasys.com/swf/SQL.aspx>) or use the following request to retrieve the data in the SQL:

“[http://www.nadasys.com/swf/SQL.aspx?SQL=select faculties from web where university=Al-Quds university](http://www.nadasys.com/swf/SQL.aspx?SQL=select%20faculties%20from%20web%20where%20university=Al-Quds%20university)“.

The following figure shows the SQL statement results:



#### **Appendix 4: Publications**

- Ahmed Nada and Badie Sartawi (2009): “The RDF Generator (RDFG) - First Unit in the Semantic Web Framework (SWF)”. Proceedings of the International Conference on Systems, Computing Sciences and Software Engineering (SCSS 09). Bridgeport, USA.