

AL- Azhar University

Faculty of Engineering

Systems and Computers Engineering Department

Semantic Web Services Framework Design

**This thesis is submitted to fulfill the partial requirements of Ph.D. degree
in Systems and Computers Engineering**

By Engineer

Islam Hany Mohamed Mohei Eldeen Mohamed Harb

Supervisors

Prof. Dr. Hassan Morsi Farahat

**Systems and Computers Engineering Department, Faculty of Engineering, AL- Azhar
University**

Dr. Mohamed Ezz

**Systems and Computers Engineering Department, Faculty of Engineering, AL- Azhar
University**

Cairo, 2014

Approval Sheet

On

Semantic Web Services Framework Design

By Engineer

Islam Hany Mohamed Mohei Eldeen Mohamed Harb

This thesis is for the Ph.D. degree in Systems and computers Engineering Department, Faculty of Engineering, Al-Azhar University, and has been

Approved by

Name	Position	Signature
Prof. Dr. Hassan Morsi Farahat	Professor of Computers and Systems Engineering, Faculty of Engineering, Al-Azhar University	
Prof. Mohamed Zaki Abdel-Majied	Professor of Computers and Systems Engineering, Faculty of Engineering, Al-Azhar University	
Prof. Fathy Ahmed Al-Sayed Amer	Professor of Information Technology, Computers and Information Faculty, Cairo University	

Cairo, 2014

Abstract

To conduct e-business, web services scattered over the Internet have to be discovered and integrated. **This research** introduces the key techniques and components of Web service architecture based on UDDI. The Web service architecture model and its implementation issues are also discussed. Intelligent registries as an ultimate goal of Semantic Web Services technology are introduced. It then discusses the semantic Web infrastructure dimensions. It also presents the semantic web services development environment showing all the required technologies. It addresses the semantic Web services activities such as automated service discovery, selection, invocation, and composition.

Semantic matching is based on the relationships between ontology-based concepts. This thesis designs a semantic matchmaker model. The model consists of Request client agent, Matchmaker Engine which implements a simple matching algorithm, Manager Module which coordinates the other modules interactions, SWS Repository and Crawler as focused search engine. Upon accepting a user request, the Request client agent invokes a given matching algorithm which returns a service or a candidate set of services satisfying the user request. Current Web services standards lack the means for expressing service's nonfunctional attributes — namely, its quality of service (QoS). The presented matching algorithm is a three-phase algorithm handling this lack. In its first phase, it selects the Web services that are relevant to a given service request by means of logic-based matching complemented with syntactic similarity measurement. These Web services should be written with Web Ontology language for Web Service (OWL-S). In the second phase, it chooses the services or groups of services with matching degree greater than a certain threshold value. The third phase considers QoS attributes as a key to dynamically selecting the services that best meet user needs. Experimental results of measuring performance of different variants of the algorithm are evaluated. The algorithm is evaluated using the second version of service retrieval OWL-S based Test Collection (OWLS-TC2).

A matchmaking application and case study is conducted and introduced to test and evaluate several and different matchmaker algorithms and techniques based on information retrieval model. It is applied on Islamic Fiqh Fatwa domain. A user query is semantically matched against the annotated Fatwas which are stored in Dar Al-Ifta Almasria database of Fatwas. A user query is also annotated base on the same domain ontology. The relevant Fatwas are retrieved. Handling large output set requires effective methods to rank the search results based on relevance to the query. Algorithms for determining such relevance are designed and evaluated. Such algorithms utilize WordNet and they consider essential features of domain Ontologies and RDFS languages to support determining this relevance.

In order to empower and improve the matchmaker, the problem of dynamically QoS-based selecting a web service and composing a set of web services to conduct a business task has been investigated in this thesis. It outlines the discovery of either atomic or composite services satisfying the request QoS requirements. There may exist more than one functionality matched services, and then the service with the best QoS (nonfunctional) attributes is chosen. The

optimal selection problem is formulated as an optimization problem, so an optimization solving techniques such as integer programming can be applied. A heuristic technique is designed and evaluated.

A cloud based 'Semantic Search Engine' architecture is designed to improve the traditional web search. It is used to search for and discover semantic web services on the internet. As the crawlers are deployed over the cloud, cloud services are discussed. Best practices for improving the reliability of the search engines using NSGA II are also introduced and proposed.

Finally, a generic personalized agent based Semantic Web Service Framework (PASWSF) as an agent-based Integrated Development Environment for Web Service activities is designed. This framework can facilitate automatic web service delivery, discovery, selection, invoking, composition and interoperation. Current solutions like OWL-S and WSMO do not satisfy all basic requirements of SWSF. Our proposed framework has more extensible and autonomous distributed architecture, and it has more complete and effective web service automation. The semantic facilities in our framework are designed at a conceptual level to guarantee correctness and avoid inconsistencies among its internal modules. This framework is based on a stack of ontologies to describe the different parts of a Semantic Web Service and it contains a set of logic rules to form more intended request from the submitted user request. The framework is based on two paradigms, agent- and service-oriented, in a way that capitalizes on their individual strengths.

ACKNOWLEDGEMENT

First, the whole gratitude is due to ALLAH. I would like to express my profound gratitude to Prof. Hassan Farahat, and Dr. Mohamed Ezz for their great effort in this research. Thanks to their encouragement, insightful remarks, limitless generosity with time and guidance, incessant follow-up. If it has any mistakes, they are all mine.

Special thanks must go to all my colleges for their guidance and continuous support.

Last but not Least, I must thank my family who has provided me with love and support, especially my Father Prof. Hany Harb who taught me a lot, my brother Mohei, and my sisters Alshimaa, and Hoda. My father has provided encouragement, sound advice, good company, and lots of good ideas. My deepest love and gratitude go to my mother for her love, encouragement and care in my life.

List of Publications

- 1- Islam Harb, Mohamed Ezz, Hassan Farahat, "Semantic Search Engine on Cloud", International Journal of Intelligent Computing and Information Science (IJICIS), December, 2013
- 2- Islam Harb, Mohamed Ezz, Hassan Farahat, "A Heuristic Algorithm For QoS (Non-Functional) Based Service Matching", International Journal of Computer Science Issues (IJCSI), Volume 10, Issue 6, November 2013.
- 3- Islam Harb, Mohamed Ezz, Hassan Farahat, "A THREE PHASE OWL-S MATCHMAKER MODEL", International Journal of Intelligent Computing and Information Science (IJICIS), Vol.13, No. 2, pages: 1-18, ISSN 1687-109X, April 2013.
- 4- Islam Harb, Mohamed Ezz, Hassan Farahat, "PASWSF: Personalized Agent-based Semantic Web Service Framework", Advances in Computer Science, proceedings in the 6th European Computing Conference (ECC'12), ISBN: 978-1-61804-126-5, Prague, Czech Republic, September 2012
- 5- Islam Hany Harb, AbdelRahaman Nasr, M. Ezz Salah Rammadan, Hassan Farahat and Hany Harb, "Design and Evaluation of Ontology Based Matching Algorithms for Arabic Information Retrieval System Applied on Islamic Fatwa", Al-Azhar Engineering Twelfth International Conference, December 25 - 27, 2012
- 6- Islam Harb, Mohamed Ezz, Hassan Farahat, "SEMANTIC WEB SERVICES DEVELOPMENT ENVIRONMENT", Al-Azhar Engineering Twelfth International Conference, December 25 - 27, 2012
- 7- Islam Harb, Mohamed Ezz, Hassan Farahat, "AUGMENTED UDDI BASED WEB SERVICES ARCHITECTURE DESIGN", International Journal of Intelligent Computing and Information Science (IJICIS), Vol.12, No. 2, July, 2012.

Table of Contents

Chapter 1: Introduction	1
1.1. Motivation	1
1.2. The Problem Definition.....	4
1.3. The Research Objectives	4
1.4. Research Domain and Scope.....	5
1.5. The Research Approach and Difficulties	5
1.6. The Thesis Organization	6
Chapter 2: Augmented UDDI Based Web Services Architecture Design	9
2.1. Introduction	9
2.2. Web Services.....	9
2.3. Web Services Architecture (WSA)	10
2.4. The WSA Implementation	11
2.4.1. The Web Services Hosting.....	12
2.4.2. The Web Services Invocation	13
2.5. Web Service Technologies Standards.....	14
2.5.1. Web Service Description Language: WSDL	14
2.5.2. Simple Object Access Protocol: SOAP.....	15
2.5.3. Universal Description Discovery and Integration : UDDI.....	15
2.5.4. The UDDI APIs	19
2.6. Using UDDI to Discover Web Services.....	20
2.6.1. Adding Categorization Information to the Service Type	21
2.7. The WSA Activity Flow	22
2.8. The Augmented UDDI Model.....	23
2.9. The UDDI Distribution Issues.....	26
2.10. The Model Performance	26
Chapter 3: Semantic Web Services Development Environment	28
3.1. Introduction	28
3.2. The Service Ontology.....	30
3.2.1. The OWL-S Concepts.....	30
3.2.2. The WSMO Concepts	40
3.3. The Matchmaking Activity	41
3.3.1. The Matchmaking Search Engine	41
3.3.2. The Search Engine Architecture	41
3.4. The Service Composition	42
3.4.1. Composition Modeling	43
3.4.2. Interaction with Composite Services	44
3.5. A Generic Semantic Web Services Framework (GSWSF).....	45
3.5.1. IRS II, IRS III: Internet Reasoning Service	45
3.5.2. ODE SWS: A Framework for Designing and Composing Semantic Web Services.....	45
3.5.3. Agent-Based Web Services Framework and Development Environment	46
3.5.4. ISCF: A Semantic Web Service Composition Framework Based on OAA.....	46
3.5.5. Semantic Web Services with the SADI Framework: my notebook.....	46
Chapter 4: A Three-Phase OWL-S Matchmaker Model	48
4.1. Introduction	48
4.2. The MatchMaker Model (MMM)	52

4.2.1.	The Model Architecture	52
4.2.2.	The Semantic Matching Metrics	52
4.2.3.	The proposed Semantic Matching Algorithm	54
4.2.4.	Phase II: K-means Clustering	55
4.2.5.	Phase III: Quality of Service Weighting	57
4.3.	The Implementation Details	58
4.3.1.	The Web Crawler	58
4.3.2.	The Repository	60
4.4.	The Matching Algorithm Analysis	61
4.5.	Test Collection	63
4.5.1.	SWS-TC	64
4.5.2.	OWLS-TC	64
4.5.3.	SAWSDL-TC	65
4.5.4.	Test Collection Discussion	65
Chapter 5:	Matchmaking Application	66
5.1.	Introduction	66
5.2.	Ontology matching	67
5.3.	The Model architecture	68
5.4.	The Matching Algorithms	72
5.4.1.	Vector space algorithm 1	72
5.4.2.	TF-IDF Modified Algorithm 2	73
5.4.3.	Weight-based Algorithm 3	74
5.4.4.	Distribution-based Algorithm 4	74
5.4.5.	Multi-level Algorithm 5	75
5.5.	The Implementations	78
Chapter 6:	A Heuristic Algorithm For QoS (Non-Functional) Based Service Matching...81	
6.1.	Introduction	81
6.2.	The Related Works	84
6.3.	The QoS Model	86
6.3.1.	The Composition model	87
6.3.2.	The QoS Aggregation	89
6.4.	The Non-functional Matching Problem Formulation	90
6.5.	The Genetic Heuristic (G_HEU) algorithm	93
6.6.	The Algorithm Analysis and Evaluation	95
Chapter 7:	Reliable Cloud Based Semantic Search Engine..... 97	
7.1.	Introduction	97
7.2.	The Semantic Search	98
7.3.	The Semantic Search Engine Architecture	101
7.3.1.	The Query User Interface	102
7.3.2.	The Discovery and Crawling Strategy	102
7.4.	Cloud Based Reliability Enhancement On Semantic Search Engine	104
7.4.1.	Cloud Models	105
7.4.2.	The Cloud clients	106
7.4.3.	The Deployment models	106
7.5.	The Problem Definition	107
7.6.	Multi-Objective RAP on crawler parallel system	108
7.7.	NSGA II Numerical Example Of The Parallel Crawlers System	110
Chapter 8:	PASWSF: Personalized Agent-based Semantic Web Service Framework...113	

8.1.	Introduction	113
8.2.	Related Works	115
8.2.1.	Need-Aware Multi-agent System.....	115
8.2.2.	A Multi-agent Based Intelligent WWW Interfacer (MAII)	115
8.2.3.	Personalized semantic search engine (PSSE)	116
8.2.4.	Case-based reasoning technique for Arabic QA (CBRA).....	116
8.2.5.	Cross language IR Based on an Arabic Ontology in the Legal Domain (CLIR)	117
8.2.6.	Malik	118
8.3.	The PASWSF Architecture	118
8.3.1.	The User Services Tier (The User Agency).....	119
8.3.2.	The Broker Services Tier (The Middleware Agency).....	123
8.3.3.	The Semantic Web Service Tier (The Services Agency).....	124
8.3.4.	The Goals	125
8.3.5.	The Mediators	126
8.4.	The Implementation and results	128
Chapter 9: Conclusions and Future Works		130
9.1.	Conclusions	130
9.2.	Future Work	133
References		134

List Of Figures

Figure 2.1: The web service technology stack (W3C web service architecture document).....	11
Figure 2.2: The server side in a Web Services Application	12
Figure 2.3: A Typical Web Service Invocation	13
Figure 2.4: The Basic UDDI Data Types	17
Figure 2.5: The WSA Activity Flow	23
Figure 2.6: Architecture of OWL-S / UDDI Matchmaker of a Single Node	24
Figure 2.7: Mapping between OWL-S Profile and UDDI	24
Figure 3.1: The OWL-S Process Types	31
Figure 3.2: The OWL-S upper ontology for describing Web services.	31
Figure 3.3: Describing IOPE in profile document using terms defined in process.owl.	34
Figure 3.4: The Structure of a search engine for Semantic Web Services.	42
Figure 4.1: The Matchmaker Model (MMM) architecture.....	53
Figure 4.2: The Crawler Algorithm Steps	59
Figure 4.3: percentage of altering services in the Test Collections	65
Figure 5.1: The Fatwa Database Schema	66
Figure 5.2: The Islamic Fatwa Arabic Information Retrieval System Architecture	69
Figure 5.3: The Islamic Fatwa Arabic Information Retrieval System Architecture Details	70
Figure 5.4: A Sample Of Fatwa WordNet Ontology.....	71
Figure 5.5: A sample Of Prayer Fatwa ontology	71
Figure 5.6: The Multi-Level Algorithm main Steps.....	76
Figure 5.7: The Query Expansion and Augmentation	77
Figure 5.8: Precision and recall curve for a query (حكم التلاوة من القرآن في الصلاة).....	78
Figure 6.1: The Functionl and Nonfunctional service discovery.	82
Figure 6.2: The QoS taxonomy.....	86
Figure 6.3: Example on Task, activities and candidate services for each activity	88
Figure 6.4: Experiment the algorithm's time consumption VS. Atomic Services Number.....	95
Figure 6.5: Experiment the algorithm's Optimality VS. Atomic Services Number	96
Figure 7.1: Semantic Search Engine Architecture	102
Figure 7.2: The More Focused Crawler	103
Figure 7. 3: The Focused Crawler Procedure.....	104
Figure 7.4: crawlers connected in parallel	107
Figure 7.5: The fast non-dominated sorting pseudo-code.....	109
Figure 7.6: The Cost and Reliability for all Six Solutions	111
Figure 8.1: The PASWSF Architecture	121
Figure 8.2: The IDB Schema	121

List Of Tables

Table 2.1: Attributes and Elements for businessService	17
Table 2.2: Attributes and Elements for bindingTemplate	18
Table 2.3: UDDI Set of APIs	20
Table 3.1: OWL-S and Web Services.....	32
Table 3.2: Properties and Classes Defined in Grounding.owl	37
Table 3.3: The Structure of the main repository Table	42
Table 3.4: The Service Details Table.....	42
Table 4.1: The Semantic Matching Degrees	53
Table 4.2: The Semantic Matching Degrees	54
Table 4.3: Notations for Our Simple Matchmaking Algorithm	54
Table 4.4: Structure of the main repository Table	60
Table 4.5: The Service Details Table.....	60
Table 4.6: Classified Services and Queries in different domains supported by TC	61
Table 4.7: The OWLS-MX relevance Set.....	62
Table 4.8: The MM Clustered Relevance Set.....	63
Table 5.1: Precision and recall curve for a query (حكم التلاوة من القرآن في الصلاة).....	79
Table 5.2: Precision and Recall values for 10 different queries.....	80
Table 6.1: The QoS Aggregation Under Different Service Compositions	89
Table 6.2: The Criterion Aggregation Formula for Sequence Composition	90
Table 7.1: The initial parent vector of six solutions	110
Table 8.1: The User Ontology Concepts.....	122
Table 8.2: Comparison between different systems.....	127
Table 8.3: Average IR Response Time and Relevancy Evaluation	129

List of Symbols

AI	Artificial Intelligence
API	Application Program Interface
DAML+OIL	DARPA Agent Markup Language + Ontology Inference Layer
Ai	Abstract activity number i in the user task
EJ	Extended Jaccard Similarity Metrics
FTP	File Transfer Protocol
GSWSF	Generic Semantic Web Services Framework
HTTP	Hyper Text Transfer Protocol
IaaS	Infrastructure as a service
IEEE	Institute of Electrical and Electronics Engineers
IOPEs	Inputs, Outputs, Preconditions, Effects
ISO/OSI RM	International Standards Organization/Open Standards Interconnection
IRIs	Internationalized Resource Identifier
IRS	Internet Reasoning System
JS	Jensen Shannon metrics similarity value
MILP	Mixed Integer Linear Program
MMM	The Matchmaker Model
MMKP	Multi-dimension Multi-choice Knapsack Problem
OME	Ontology Management Engine
OIL	Ontology Inference Layer
OO	Object Orientation
OWL	Web Ontology Language
OWL-S	Web Ontology Language-Services
OWLS-TC	Web Ontology Language-Services Test Collection
PaaS	Platform as a Service
PASWSF	Personalized Agent-based Semantic Web Service Framework
QoC	Quality of Content
QoD	Quality of Device
QoS	Quality of Service
QSD	Quality-based service description
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
OIL	Ontology Inference Layer

S	Set of service candidates associated with an abstract activity activity
SaaS	Software as a Service
S_{ik}	Concrete service k fulfilling the abstract activity i
SMTP	Simple mail Transfer Protocol
SOAP	Simple Object Access Protocol
S-SGMA	Simple Semantic Greedy Matchmaking Algorithm
SWS	Semantic Web Service
SWSF	Semantic Web Service Framework
UDDI	Universal Description Discovery and Integration
U	Set of global QoS constraints required by the user
UI	User Interface Layer
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WI	Web Intelligence
WSA	Web Services Architecture
WSDL	Web Service Description Language
WSDL-S	Web Service Description Language-Services
WSMO	Web Service Modeling Ontology
WSD	Word Sense Disambiguation
W	Set of weights defining user preferences for QoS properties
W_i	User preference for QoS property p _i
TF-IDF	Term Frequency-Inverse Document Frequency
XML	eXtensible Markup Language

CHAPTER ONE

Introduction

Chapter 1: Introduction

1.1.Motivation

The web services are now one of the main resources on the Internet. This field significantly affects both academic research and everyday life, revolutionizing how we gather, store, process, present, share, and use information and make actions. It offers great opportunities in many areas, including business, commerce, marketing, finance, publishing, education, and research and development. Since the last decade, the number of web services and its number of applied fields (such as e-learning and e-commerce) have been dramatically increased.

A service is a function or some processing logic or business processing that is well-defined, self-contained, and does not depend on the context or state of other services. Web Services are services that can be published, located, and invoked across the Internet. These Web Services may use other Web Services in order to perform their task. Web Services, as formally defined by “WebServices.org”, are encapsulated, loosely coupled contracted functions offered via standard protocols”. **“Encapsulated”** means the implementation of the function is never seen from the outside. **“Loosely coupled”** means changing the implementation of one function does not require change of the invoking function. **“Contracted”** means there are publicly available descriptions of the function’s behavior and how to bind to the function as well as its input and output parameters. Web Services are platform-independent and language-independent, since they use standard XML languages and most Web Services use HTTP for transmitting messages. This is a major advantage if we want to build an Internet-scale application, since most of the Internet’s proxies and firewalls will not mess with HTTP traffic but transmitting data in XML is obviously not as efficient as using a proprietary binary code. Anyway, this overhead is usually acceptable for most applications, but a critical real-time application probably never uses Web Services.

With the emergence of computer networks, the paradigm of distributed computing was born, where applications were split first into two parts with one part, the client, initiating a distributed activity, and the other part, the server, carrying out that activity (two-tier architecture). This decentralization minimizes bottlenecks by distributing the workload across multiple systems providing application design flexibility. This two-tier architecture still has its drawbacks concerning scalability, so a middle tier was added containing the business logic and a third tier became a database handler. This three tier model of distribution has become the most popular way of splitting applications. The foundation for the inter-tier communication is the remote procedure call (RPC). This middleware may mask the differences between various kinds of hosts to hide the low-level tasks from developers. This type of software sits on top of the operating system and networking services. Middleware types like CORBA, DCOM, or RMI, which are the most popular middleware at present, result in a tight coupling of the client and the server and the connection is established on a point-to-point basis. Because of the different protocols, a DCOM server for example cannot be called from a RMI client, so these middleware types are typically