



كلية الهندسة



جامعة القاهرة

اطار عمل مبتكر لتحقيق الاكتشاف و التركيب المبني على الأهداف لخدمات الويب الدلالي

إعداد

محمود ربيع السيد ربيع

رسالة مقدمة إلى كلية الهندسة، جامعة القاهرة
كجزء من متطلبات الحصول على درجة الماجستير
في هندسة الحاسبات

تحت إشراف

أستاذ مشارك دكتور

السيد عيسى حميد

جامعة القاهرة

أستاذ دكتور

ماجدة بهاء الدين فايق

جامعة القاهرة

كلية الهندسة ، جامعة القاهرة
الجيزة ، جمهورية مصر العربية

٢٠١٤

الملخص

تطبيقات الويب الممزوجة (Web Mashups) على شبكة الإنترنت هي تطبيقات تم تطويرها باستخدام المحتويات والخدمات المتاحة على الإنترنت (Web Services). وأصبحت (Mashups) واحدة من أهم المصطلحات الواعدة الرنانة في مجال تطبيقات الويب، فالعديد من الشركات والمؤسسات يسارعون لتقديم حلول تستخدم Mashups. وعلى الرغم من أن الإكتشاف الدلالي (Semantic Discovery) لم ينضج بعد، فإن أحدث ما أبدع فيه (State-of-the-art) ليس فقط يمكن أن يساهم في تسريع Mashup Discovery ولكن أيضا يمكن أن يتيح درجة أعلى من التشغيل الآلي في تطوير الممزوجات Mashup Development.

JOpera هي واحدة من أكفأ بيئات تطوير تطبيقات الويب الممزوجة (Mashup Tools) أو تعتبر من البيئات الفعالة لتركيب خدمات الويب (Web Service Composition Tool). يتم استخدام بيئة التطوير المتكاملة (JOpera IDE) لاعداد النماذج الأولية السريعة (Rapid Prototyping) و لتطوير عمليات سير العمل (Workflow Processes). تم تطوير JOpera IDE على منصة Eclipse باستخدام مجموعة من Plug-ins. تقوم JOpera باستدعاء مقدمي خدمات ويب محددين Certain Service Providers لتحقيق مهام محددة Certain Tasks قد تكون احادية مبسطة (atomic) أو تكون مركبة (composite). لكن JOpera بها قصور هو أنها لا توفر بدلاء من موفري الخدمة (Alternative Service Providers) لتلبية احتياجات المستخدم (User goals). إن (JOpera) تفتقر الى الاكتشاف التلقائي (Automatic Discovery) استنادا إلى أهداف المستخدم. بعد البحث و الدراسة في كيفية الإسهام في تطويرها تبين لنا أن الخيار الأفضل لتوفير خدمة الاكتشاف التلقائي Automatic Discovery هو استخدام خدمات الويب الدلالي (Semantic Web Services).

قمنا في هذه الأطروحة للحصول على الماجستير ببناء إطار عمل مبتكر هدفه توسيع قدرات JOpera بإضافة قدرات جديدة للاكتشاف الدلالي لخدمات الويب (Semantic Web Service Discovery) باستخدام النموذج المرجعي لتنفيذ خدمات الويب الدلالي (WSMX). حيث يقوم هذا الإطار بتوظيف WSMX Middleware كوسيط لدعم الإكتشاف التلقائي لخدمات الويب الدلالي و مع كون JOpera واجهة للمستخدم Front-end تستخدم اضافاتنا البرمجية framework adapters فإن قوة الإكتشاف التلقائي يمكن أن تساعد في بناء ال Mashups ذات مرونة عالية تتناغم و الجيل الجديد من الويب Semantic Web أو Web 3.0 .

**A FRAMEWORK FOR GOAL-BASED
SEMANTIC WEB SERVICE
DISCOVERY AND COMPOSITION**

By

Mahmoud Rabie El-Sayed Rabie

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Computer Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

A FRAMEWORK FOR GOAL-BASED SEMANTIC WEB SERVICE DISCOVERY AND COMPOSITION

By
Mahmoud Rabie El-Sayed Rabie

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Computer Engineering

Under the Supervision of

Prof. Dr. Magda Fayek

Assoc. Prof. Dr. Elsayed Hemayed

.....
Computer Engineering Department
Faculty of Engineering, Cairo University

.....
Computer Engineering Department
Faculty of Engineering, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

A FRAMEWORK FOR GOAL-BASED SEMANTIC WEB SERVICE DISCOVERY AND COMPOSITION

By
Mahmoud Rabie El-Sayed Rabie

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Computer Engineering

Approved by the
Examining Committee

Prof. Dr. Magda B. Fayek,	Main Advisor
Assoc. Prof. Dr. Elsayed E. Hemayed,	Advisor
Prof. Dr. Hoda M. Hosny,	External Examiner
Prof. Dr. Salwa K. AbdelHafiz,	Internal Examiner

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

Engineer's Name: Mahmoud Rabie El-Sayed Rabie
Date of Birth: 24/09/1978
Nationality: Egyptian
E-mail: mahmoudrabie2004@gmail.com
Phone: +966556246670
Address:
Registration Date: 1 / 10 / 2007
Awarding Date: / /
Degree: Master
Department: Computer Engineering
Supervisors:



Prof. Dr. Magda Fayek
Assoc. Prof. Dr. Elsayed Hemayed

Examiners:

Prof. Dr. Magda Fayek	(Main Advisor)
Assoc. Prof. Dr. Elsayed Hemayed	(Advisor)
Prof. Dr. Hoda Hosny	(External Examiner)
Prof. Dr. Salwa AbdelHafiz	(Internal Examiner)

Title of Thesis:

A Framework for goal-based semantic service discovery and composition

Key Words:

Semantic Web Services; Web Mashup; Discovery; Composition; JOpera

Summary:

Web mashups are Web applications developed using contents and services available online. "Mashup" has become one of the hottest buzzwords in the Web applications area, and many companies and institutions are rushing to provide mashup solutions. Even though semantic discovery has not yet matured, its state-of-the-art not only can contribute to streamlining mashup discovery but can also enable a higher degree of automation in mashup development.

JOpera is one of the efficient mashup development or service composition tools. JOpera's integrated development environment (IDE) is used for rapid prototyping and development of workflow processes. The IDE is implemented on the Eclipse platform as a set of plug-ins. JOpera calls specific service providers to achieve certain atomic or composite tasks. It does not provide choices for user needs (goals) from multiple alternative service providers. It lacks automatic discovery based on user goals. The best option to provide automatic service discovery is using semantic web services.

The thesis proposes a novel framework aiming to extend JOpera by adding new capabilities for semantic web service discovery using the discovery framework of the reference model of semantic web service execution environment (WSMX). The framework employs WSMX as a middleware to support semantic discovery of web services based on goals passed by JOpera as frontend. Therefore, the power of SWS discovery and composition was added to the mashups built by JOpera.

Acknowledgments

Prophet Mohammed PBUH said:

"Whoever does not give thanks to the people does not give thanks to Allah"

(a saheeh hadeeth narrated by Imam Ahmad, 2/258)

I would firstly like to express my sincere gratitude to my supervisors, Dr. Magda Fayek and Dr. Elsayed Hemayed for all the help, support, guidance and advice throughout the whole duration of my master thesis study. I have to thank them for being an endless source of motivation.

Furthermore, I should not forget to say thanks to the following people:

- 1- Dr. Cesare Pautasso, University of Lugano, Switzerland
- 2- Dr. Srdjan Komazec, STI Innsbruck, Austria
- 3- Dr. Leonardo Mariani, University of Milano-Bicocca, Italy
- 4- Dr. Fabrizio Pastore, University of Milano-Bicocca, Italy

Finally, I would like to give my special thanks to my mother, my wife, whole family and my friends for their continuous support, prayers and Doaa' for me. I say to all of them : *"May Allah reward you"*

Dedication

I would like to dedicate this dissertation to the memory of my father.

*This dissertation is dedicated also to the endless sources of love:
my mother, my wife and my children: AbdulRahman and Ibrahim.*

Table of Contents

CHAPTER 1 : INTRODUCTION.....	1
1.1. MOTIVATION	1
1.1.1. <i>Internet Research</i>	1
1.1.2. <i>Semantic Web Research</i>	1
1.1.3. <i>Extending Integrated Development Environments</i>	2
1.1.4. <i>Extending JOpera</i>	2
1.2. RESEARCH GOALS	4
1.3. ORGANIZATION OF THE THESIS	4
CHAPTER 2 : LITERATURE REVIEW	5
2.1. BACKGROUND AND CONCEPTS	5
2.1.1. <i>Service-Oriented Architecture (SOA)</i>	5
2.1.2. <i>Service-Oriented Computing</i>	6
2.1.3. <i>Services and Web Services</i>	7
2.1.4. <i>WSDL</i>	8
2.1.5. <i>Web Service Types</i>	9
2.1.6. <i>REST</i>	10
2.1.7. <i>Web Service Repositories</i>	10
2.1.8. <i>Web Service Search Engines</i>	11
2.1.9. <i>Web Service Communities</i>	12
2.1.10. <i>Service Providers and Dynamism</i>	12
2.2. WEB SERVICE COMPOSITION	13
2.2.1. <i>Megaprogramming</i>	13
2.2.2. <i>Mashups</i>	13
2.2.3. <i>Mashup Development Tools</i>	14
2.3. JOPERA.....	15
2.3.1. <i>Functionalities and Architecture</i>	15
2.3.2. <i>Example Mashup composed by JOpera</i>	18
2.3.3. <i>Service Invocation Adapters</i>	20
2.3.4. <i>OML Component Type Definition</i>	21
2.4. SEMANTIC WEB (SW)	21
2.4.1. <i>Semantic Web Overview</i>	21
2.4.2. <i>Ontologies</i>	23
2.4.3. <i>Semaphopia</i>	25
2.5. SEMANTIC WEB SERVICE (SWS)	26
2.5.1. <i>Semantic Web Service Overview</i>	26
2.5.2. <i>SWS Lifecycle</i>	28
2.5.3. <i>SWS Languages</i>	29
2.5.4. <i>SWS Challenge</i>	29
2.5.5. <i>Web Service Modeling Ontology (WSMO)</i>	29
2.5.6. <i>Web Service Modeling Language</i>	31
2.5.7. <i>Web Service Modeling Toolkit (WSMT)</i>	33
2.6. WEB SERVICE DISCOVERY	34
2.6.1. <i>Web Service Discovery Overview</i>	34
2.6.2. <i>Semantic Web Service Discovery</i>	36
2.7. SEMANTIC EXECUTION ENVIRONMENT (SEE).....	37
2.7.1. <i>Semantically-Enabled Service-oriented Architecture</i>	37

2.7.2.	<i>Web Service Modeling Execution Environment (WSMX)</i>	41
2.7.2.1.	SWS Discovery approaches and WSMX.....	43
2.7.2.2.	WSMX discovery framework.....	44
2.7.3.	<i>COIN GSP SEE and WSMX</i>	44
CHAPTER 3 : RELATED WORK		46
3.1.	RESEARCHES EMPLOYING WSMX	46
3.2.	WSMX DISCOVERY USED BY FRAMEWORK	51
3.3.	JOPERA RELATED RESEARCH.....	51
CHAPTER 4 : PROPOSED FRAMEWORK.....		54
4.1.	THE FRAMEWORK: BEFORE AND AFTER	54
4.2.	MAIN CONTRIBUTION CONCEPT MAP	55
4.3.	WSMXENTRYPOINTS WEB SERVICE	58
4.4.	MAINTAINING WSMX-RESHAPE BRANCH	59
4.5.	ADDING WSMX DISCOVERY ADAPTERS TO JOPERA	61
4.5.1.	<i>First Adapter (WSMX)</i>	65
4.5.2.	<i>Second Adapter (WSMXDiscovery)</i>	68
4.5.2.1.	Adopting the Integrated Discovery Framework	68
4.5.2.2.	Building the adapter.....	68
4.5.2.3.	Sequence Diagrams.....	71
CHAPTER 5 : TESTS AND RESULTS.....		74
5.1.	TESTING WSMX ENTRY POINTS ADAPTER.....	74
5.2.	TESTING WSMXDISCOVERY ADAPTER	76
5.2.1.	<i>Discovery test on Vehicle Ontology</i>	76
5.2.2.	<i>Discovery test on COIN Enterprise Interoperability Ontology</i>	79
CHAPTER 6 : CONCLUSION AND FUTURE WORK.....		86
6.1.	CONCLUSION	86
6.2.	FUTURE WORK	86
APPENDIX A: SOME IMPORTANT CONCEPT MAPS		87
	WSML	87
	SWS DISCOVERY APPROACHES AND WSMX	88
	SESA.....	89
APPENDIX B: LISTINGS.....		90
	LISTING 1: THE FIRST ADAPTER SUBSYSTEM'S CODE	90
	LISTING 2: VEHICLE ONTOLOGY	91
	LISTING 3: FIND AN AUTOMOBILE GOAL	92
	LISTING 4: AUTOMOBILE VENDOR WEB SERVICE	92
	LISTING 5: CAR VENDOR WEB SERVICE.....	93
	LISTING 6: BICYCLE VENDOR WEB SERVICE	93
	LISTING 7: CARBICYCLE_ONTO.XML	93
	LISTING 8: CARBICYCLE_SWS.XML	94
	LISTING 9: THE SECOND ADAPTER SUBSYSTEM'S CODE	94
	LISTING 10: MODELTRANSFORMATIONGOAL.WSML	97
	LISTING 11: BUSINESSMODELTRANSFORMATIONGOAL.WSML.....	97
APPENDIX C: WSML SYNTAX OVERVIEW		98

<i>Identifiers in WSML</i>	<i>98</i>
<i>Conceptual Syntax.....</i>	<i>98</i>
<i>Logical Expression Syntax</i>	<i>100</i>
REFERENCES.....	101

List of Tables

Table 1.1 Evaluation of Mashup Tools [6].....	3
Table 2.1: Overview of retrieval methods in Web service repositories [15].....	11
Table 2.2: Evolution from Web 1.0 to Semantic Web [43].....	22
Table 2.3: Comparison between web 1.0, web 2.0, and web 3.0 [43]	22
Table 2.4: Phases of SWS Lifecycle [49].....	28
Table 2.5: Feature Matrix of WSML variants [57].....	32

List of Figures

Figure 2.1: Web Services Programming Model [9]	6
Figure 2.2: SOC Research Roadmap [7]	7
Figure 2.3: Web Services usage process [1]	8
Figure 2.4: WSDL elements [1]	9
Figure 2.5: Ingredients of WSDL [8]	9
Figure 2.6: JOpera Service Meta-Model [29]	16
Figure 2.7: JOpera Plugin-based Architecture [29]	16
Figure 2.8: Extensible JOpera Component Model [37]	17
Figure 2.9: Common service adapters used in JOpera compositions [5]	17
Figure 2.10: Anatomy of JOpera platform [30]	18
Figure 2.11: DoodleMap: Control Flow Dependency Graph [38]	19
Figure 2.12: Screenshot of the DoodleMap example [38]	20
Figure 2.13: Semantic Web Layer Cake [42]	23
Figure 2.14: An ontology example [41]	24
Figure 2.15: Types of ontologies [41]	24
Figure 2.16: From www to Semantic Web Service [2]	27
Figure 2.17: WSMO Meta-Model [55]	30
Figure 2.18: WSMML variants and layering [58]	32
Figure 2.19: WSMML Document Structure [57]	33
Figure 2.20: Position of discovery in the web services lifecycle [62]	35
Figure 2.21: SESA Components [9]	40
Figure 2.22: High-level diagram of WSMX [77]	42
Figure 3.1: GGODO Main Components [81]	46
Figure 3.2: High-level architecture of the SCAIMO framework [82]	47
Figure 3.3: COIN BSP adopts WSMX middleware [72]	48
Figure 3.4: Dynamic Discovery in B2B Integration using WSMX [83]	49
Figure 3.5: The framework proposed is using WSMX [84]	50
Figure 3.6: Elderly Caring Solution employs JOpera [85]	51
Figure 3.7: JOpera storage architecture extended with buffers [86]	52
Figure 3.8: Virtual EZ Grid Architecture [88]	53
Figure 3.9: XtremWeb-CH (XWebCH) Adapter for JOpera [89]	53
Figure 4.1: The Framework adding SWS adapters to JOpera	55
Figure 4.2: Main Contribution Concept Map	56
Figure 4.3: Concept Map to understand adapters in JOpera	57
Figure 4.4: WSMXEntryPoints web service	58
Figure 4.5: The result of calling getGoalIdentifiers	59
Figure 4.6: Comments in workspace for fixing and workarounds	60
Figure 4.7: Some completed TODOs	61
Figure 4.8: List of Component Types declared by predefined JOpera adapters	62
Figure 4.9: WSMX adapter used to call WSMXEntryPoints	63
Figure 4.10: Structure of the framework's plugin when building the first adapter .	64
Figure 4.11: wsmx.oml opened with text editor	65
Figure 4.12: wsmx.oml opened with JOpera editor	65
Figure 4.13: extensions of plugin.xml	66
Figure 4.14: wsmx_example.oml used to test the framework	67
Figure 4.15: Classpath of WSMXDiscovery adapter's plugin	68
Figure 4.16: WSMXDiscovery Component Type	69
Figure 4.17: WSMXDiscovery in JOpera's Component Type Browser	70
Figure 4.18: Sequence Diagram of Execute() method	71

Figure 4.19: Sequence Diagram of callDiscoveryFW() method	72
Figure 4.20: Sequence Diagram of method initialize()	73
Figure 5.1: run configuration to test the first adapter.....	74
Figure 5.2: Test values for example project of WSMX adapter.....	75
Figure 5.3: The result of calling the first WSMX adapter.....	75
Figure 5.4: Vehicle Ontology in WSMT Visualizer	76
Figure 5.5: wsmxdiscovery_prog.....	77
Figure 5.6: Data Flow of WSMXDiscovery test process	77
Figure 5.7: discoveredXML output by calling the test process shown in JOpera Monitor.....	78
Figure 5.8: COIN Enterprise Interoperability Ontology in WSMT Visualizer	80
Figure 5.9: JOpera Design View of mashup TwoGoalsDiscovery	81
Figure 5.10: JOpera Control Flow of TwoGoalsDiscovery mashup	82
Figure 5.11: JOpera Data Flow of TwoGoalsDiscovery mashup	82
Figure 5.12: JOpera Monitor shows Mashup tasks in ready and running states	83
Figure 5.13: JOpera Monitor shows Mashup tasks in finished state	84

Nomenclature

API	Application Programming Interface
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Model
cufe.cd	Cairo University, Faculty of Engineering . Computer Department
DAML-S	DARPA Agent Markup Language for Services
ebXML	electronic business using XML
IDE	Integrated Development Environment
JDT	Java Development Tools
MOF	Meta-Object Facility
OASIS	Organization for the Advancement of Structured Information Standards
QoS	Quality of Service
OSGi	Open Service Gateway initiative
OWL	Web Ontology Language
OWL-S	Web Ontology Language for Web Services
PDE	Plug-in Development Environment
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SAWSDL	Semantic Annotations for WSDL and XML Schema
SEE	Semantic Execution Environment
SESA	Semantically-Enabled SOA
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SPARQL	Simple Protocol And RDF Query Language
SQL	Structured Query Language
SSOA	Semantic Service Oriented Architecture
SWING	Semantic Web services INteroperability for Geospatial decision making
SWS	Semantic Web Service
SWT	Standard Widget Toolkit
UDDI	Universal Description, Discovery and Integration
W3C	WorldWide Web Consortium
WS-Discovery	Web Services Dynamic Discovery
WSDL	Web Services Description Language
WSMF	Web Service Modeling Framework
WSML	Web Service Modeling Language
WSMO	Web Service Modeling Ontology
WSMX	Web Service Modeling Execution
XML	Extensible Markup Language