

# **Enabling ad hoc interaction with electronic services**

by

**Phillipa Oaks, BInfoTech Hons.**

**Thesis**

Submitted by Phillipa Oaks

for fulfillment of the Requirements for the Degree of

**Doctor of Philosophy**

Supervisor: Associate Professor Arthur H.M. ter Hofstede

Associate Supervisor: Dr David Edmond

**Faculty of Information Technology  
Queensland University of Technology**

July, 2005

**Keywords:** Web services, service description, ad hoc interaction, capabilities, outsourced type descriptions.

## Abstract

Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web [154]

Web services are a promising technology for ad hoc machine to machine interaction across application, enterprise and web boundaries. Self describing web services is a catchy phrase but it should mean more than having an interface description written in XML syntax.

This research is motivated by the vision of web services in the future as loosely coupled applications operating on different platforms inter-operating without prior agreements in place and without direct human intervention at runtime.

The main obstacle to advancing the vision of ad hoc runtime interaction is complexity. The complexity of ad hoc interaction for web services is related to 1) the information the service requires and provides and the nuances of the domain or context the service operates on and in. 2) The specific nature of the operations the service provides and the constraints related to those operations and 3) the necessary ordering of operations to achieve the desired result.

There are three problems that must be addressed before the vision for web services can become a reality. These problems are aligned with the three aspects of service complexity identified above. The three inter-related elements of this research address each of these problems.

The first part of the research deals with *what* web services “talk” about and how the data required or provided by services can be described to enable mutual understanding. An extension to traditional conceptual models, called outsourced type descriptions, allows the description of shared data in terms of publicly available information, including standards, specifications, ontologies and definitions from dictionaries and thesauri.

The second part is concerned with describing *why* services interact and the capabilities (actions or information) services can provide. A structured format for the description, advertisement and discovery of services based on what they actually do is presented. The structured format is based on previous work in the description of actions and the context in which they are performed.

The last part of the research addresses *how* previously unknown services can talk to one another to supply and use the advertised capabilities. Interaction is based on providers having “plans” for the delivery of capabilities. The flow of interaction is directed by the service providers’ data requirements and is responsive to the resources of the client. A small language for information gathering based on well known interaction primitives is defined. An example implementation of a capability plan interpreter demonstrates how messages are generated, managed and interpreted at runtime in order to satisfy the client’s goals.

© Copyright

by

Phillipa Oaks

2005

In memory of my dad George Oaks, a life long learner.

# Contents

<b>Abstract</b> . . . . .	<b>iii</b>
<b>List of Figures</b> . . . . .	<b>ix</b>
<b>List of Tables</b> . . . . .	<b>xi</b>
<b>Acknowledgments</b> . . . . .	<b>xiii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Problem area . . . . .	1
1.2 Web services . . . . .	2
1.2.1 WSDL . . . . .	3
1.2.2 Semantic web services . . . . .	5
1.3 Ubiquitous services . . . . .	7
1.4 Problem statement . . . . .	7
1.4.1 The problem of shared data . . . . .	8
1.4.2 The problem of advertising and finding services that can satisfy goals . . . . .	11
1.4.3 The problem of interaction between web services . . . . .	13
1.5 Scope . . . . .	15
1.6 Criteria for solution . . . . .	16
1.6.1 Formal . . . . .	16
1.6.2 Conceptual . . . . .	17
1.6.3 Expressive . . . . .	17
1.6.4 Comprehensible . . . . .	18
1.6.5 Executable . . . . .	19
1.6.6 Suitable . . . . .	19
1.7 Case study scenario . . . . .	20
1.8 Outline . . . . .	21
<b>2 Web service interface description languages</b> . . . . .	<b>23</b>
2.1 Documenting interfaces . . . . .	24
2.2 WSDL evaluation . . . . .	26
2.3 OWL-S evaluation . . . . .	27
2.4 Summary . . . . .	29

2.5	Epilogue . . . . .	30
2.6	Discussion . . . . .	32
<b>3</b>	<b>Outsourced types . . . . .</b>	<b>35</b>
3.1	Introducing Object Role Modeling . . . . .	35
3.2	Case Study . . . . .	36
3.3	Outsourced Types . . . . .	38
3.3.1	Information Sources . . . . .	38
3.3.2	Service Descriptions and Capability Oriented Modeling . . . . .	39
3.3.3	Identification Schemes . . . . .	39
3.4	Meta Model . . . . .	41
3.5	Identifying Outsourced Types . . . . .	45
3.6	Realisation . . . . .	46
3.7	Discussion . . . . .	48
<b>4</b>	<b>Capabilities: describing what services do . . . . .</b>	<b>51</b>
4.1	Existing work in capability description . . . . .	52
4.1.1	Capability descriptions for software agents . . . . .	54
4.1.2	Capability description for reuse . . . . .	56
4.1.3	Semantic service description . . . . .	56
4.2	A conceptual model of capability . . . . .	58
4.2.1	Evaluation of the model . . . . .	64
4.2.2	Querying the model . . . . .	65
4.3	Realisation . . . . .	66
4.4	Discussion . . . . .	67
<b>5</b>	<b>Interaction . . . . .</b>	<b>69</b>
5.1	Criteria . . . . .	70
5.2	Foundations . . . . .	71
5.2.1	Computer interaction mechanisms . . . . .	71
5.2.2	Linguistics . . . . .	73
5.2.3	Intelligent Agents . . . . .	73
5.2.4	Protocols . . . . .	74
5.2.5	Wizards . . . . .	76
5.2.6	Human Computer Voice Dialogue systems . . . . .	76
5.2.7	XML, XML schema . . . . .	77
5.2.8	Voice XML . . . . .	77
5.3	The guide: a dialogue mechanism for services . . . . .	78
5.3.1	A conversation language . . . . .	79
5.3.2	A plan language . . . . .	86
5.4	Realisation: Tracing the delivery of a capability . . . . .	95
5.4.1	Initiate a dialogue . . . . .	99
5.4.2	Start a capability . . . . .	100
5.4.3	Select an option . . . . .	107

5.4.4	Run a subdialogue . . . . .	110
5.4.5	Collect input . . . . .	111
5.4.6	Return from a subdialogue . . . . .	113
5.4.7	Provide help . . . . .	115
5.4.8	Process the input . . . . .	118
5.4.9	Summary . . . . .	120
5.5	Related work . . . . .	123
5.6	Discussion . . . . .	124
<b>6</b>	<b>Conclusion . . . . .</b>	<b>129</b>
6.1	Evaluation . . . . .	129
6.2	Discussion . . . . .	131
	<b>Bibliography . . . . .</b>	<b>133</b>
	<b>Appendix A Information source in OWL . . . . .</b>	<b>151</b>
	<b>Appendix B Example data for a currency conversion service . . . . .</b>	<b>155</b>
	<b>Appendix C Model with Restart, Cancel, Pause and Resume . . . . .</b>	<b>157</b>
	<b>Vita . . . . .</b>	<b>159</b>

# List of Figures

3.1	Securities lending conceptual model . . . . .	37
3.2	Meta-model of outsourced object types . . . . .	42
3.3	Information sources for outsourced object types . . . . .	43
3.4	One to Many and Many to Many relationships . . . . .	47
4.1	A conceptual model to describe capability . . . . .	60
4.2	A conceptual model of capability parameters . . . . .	61
4.3	Information sources relating to the capability models . . . . .	62
4.4	Conceptual queries on the capability model. . . . .	66
5.1	Protocol building blocks . . . . .	75
5.2	Message schema in ORM . . . . .	80
5.3	XML Schema for Messages . . . . .	81
5.4	Message example in XML . . . . .	82
5.5	Two performatives and the allowed responses . . . . .	85
5.6	Instruction schema . . . . .	87
5.7	A tree representation of the “ConvertCurrency” capability plan . . . . .	88
5.8	A tree representation of the “PickFrom” plan . . . . .	89
5.9	The convert currency capability plan . . . . .	90
5.10	Obligation and expectation schema . . . . .	92
5.11	Item and parameter schema . . . . .	94
5.12	The Guide representing the service provider . . . . .	97
5.13	The service client . . . . .	98
5.14	The client has a plan to get information from the service . . . . .	100
5.15	The client creates a message requesting a menu . . . . .	100
5.16	Mapping the conceptual models to the implementation in CPN ML. . . . .	101
5.17	The client sends the request message . . . . .	102
5.18	The service receives an Ask Result message requesting a “Menu” . . . . .	102
5.19	The service starts processing the “Menu” request . . . . .	104
5.20	The service sets up the “Menu” capability . . . . .	105
5.21	The first instruction in the “Menu” capability plan is loaded . . . . .	106
5.22	The first item in the “Menu” capability plan is instantiated. . . . .	106
5.23	The guide prepares the first request for input. . . . .	107
5.24	The guide sends the first request for input. . . . .	108
5.25	The client matches the offered capabilities with its goals. . . . .	109

5.26	The client prepares a response to the Ask Select message. . . . .	109
5.27	The guide receives the clients selection. . . . .	110
5.28	The guide evaluates the clients selection. . . . .	110
5.29	The guide initiates a sub-dialogue. . . . .	111
5.30	The guide sets up the “Login” sub-dialogue. . . . .	112
5.31	The guide asks for input. . . . .	112
5.32	The client responds to the input request. . . . .	113
5.33	The guide finishes collecting input. . . . .	114
5.34	The “Login” subdialogue returns to its caller. . . . .	115
5.35	The guide returns to the “Menu” capability plan. . . . .	116
5.36	The guide offers a list of its capabilities. . . . .	116
5.37	The client needs to supply a currency code. . . . .	117
5.38	The client needs help. . . . .	117
5.39	The guide responds to the help request. . . . .	118
5.40	The client receives help. . . . .	118
5.41	The guide receives the final input value. . . . .	119
5.42	The guide is finished and can process the input. . . . .	120
5.43	The input is sent for processing. . . . .	120
5.44	The client receives the response to its request. . . . .	121
B.1	Alternative sources of information for the conversion capability. . . .	156
C.1	Service Provider . . . . .	157

# List of Tables

2.1	Summary of evaluation results . . . . .	30
2.2	Updated summary of evaluation results . . . . .	31

# Enabling ad hoc interaction with electronic services

## Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

---

Phillipa Oaks  
July 15, 2005

# Acknowledgments

This work would not have been possible without the unflagging support of my partner Craig Barr. I cannot thank him enough for all he has done to provide a good life for me and our children while I have been preoccupied with my research. Thank you also to Matthew and Emily who have put up with me not being there and kept me in touch with the real world.

I want to thank my supervisors Arthur ter Hofstede and David Edmond. They are the best kind of supervisors, involved, available, supportive and pushy! Each, in his own way, has shown me how to make this work better than I could have hoped for.

I undertook my research in the context of the REDCONE project funded by the Australian Research Council's SPIRT Grant "Self-describing transactions operating in a large, heterogeneous, distributed environment". This was a collaborative research project involving GBST Holdings Pty Ltd (GBST), the Queensland University of Technology (QUT) and the University of New South Wales (UNSW). I am grateful for the financial assistance and travel opportunities provided by GBST and the SPIRT Grant.

The advantage of being part of a large research project is having the support of the other people involved. These people have given me their friendship as well as listened, asked questions and made suggestions which have made my work better. In addition to my supervisors Arthur and David I want to thank: Mitra Heravizadeh, Marlon Dumas, Murray Spork, Lachlan Aldred, Lawrence Si, Justin O'Sullivan, Nick Russell and Moe Thandar Wynn (especially for the Tim Tams!)

Phillipa Oaks

*Queensland University of Technology*  
*July 2005*

