2006

Constraint-based & agent-based requirements engineering

Y. Guan
University of Wollongong, yguan@uow.edu.au

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author.

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Recommended Citation


Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Constraint-Based & Agent-Based Requirements Engineering

A thesis submitted in fulfillment of the requirements for the award of the degree

Master of Computer Science by Research

from

UNIVERSITY OF WOLLONGONG

by

Ying Guan

School of IT & Computer Science

June 2006
© Copyright 2006

by

Ying Guan

All Rights Reserved
Dedicated to

my grandparents,

parents,

and husband
Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Ying Guan
June 5, 2006
Non-functional requirements represent a critical and difficult problem in requirement engineering, but are often ignored. Usually, these are articulated as statements of objectives, as opposed to propositional assertions. A key challenge in dealing with objectives is that there is no obvious means of deciding when they are satisfied. In effect, these objectives are never fully satisfied, but satisficed to varying degrees. In evaluating alternative design decisions, we need to trade-off varying degrees of satisfaction of potentially mutually contradictory non-functional requirements.

One key contribution of this work is the use of the hierarchical constraint logic programming framework in dealing with non-functional requirements. We show how NFRs can be formulated as soft constraints and how the machinery associated with constraint hierarchies can be used to evaluate the alternative trade-offs involved in seeking to satisfy a set of non-functional requirements that might pull in different directions. We apply also this approach to the problem of reasoning about web service selection and composition, and establish that significant value can be derived from such an exercise.

Our second contribution is to develop an approach to executing high-level requirements models represented in the $i^*$ agent-oriented conceptual modeling language. We achieve this by translating these into sets of interacting agents implemented in the 3APL language. This approach enables us to analyze early phase system models by performing rule-/consistency-checking at higher-levels of abstraction. We show how this approach finds special application in the analysis of high-level models of service-oriented architectures.

The overarching project within which this research has been conducted seeks to bring to bear the full power of hierarchical constraint logic programming to requirements engineering problems. That this is a feasible exercise is clear from the close connections between logic programming and the style of agent programming explored here, as well as the connections between non-functional requirements and hierarchical
representations of soft constraints that we have established in this thesis. The full project is beyond the scope of this Masters dissertation, but the research presented here may be viewed as laying the groundwork for this exercise.
I would like to express my gratitude to my supervisor Prof. Aditya Ghose for his many insightful comments and thoughts that guided me to finish this research. I am also thankful to my other colleagues in Decision Systems Laboratory (DSL) for their valuable comments, supports, helps and encouragement during the process of completing this thesis as well as during the period of my master study.
List of Publications

This is a list of referred papers that is related to this research work.


# Table of Contents

Abstract v

Acknowledgements vii

List of Publications viii

Table of Contents ix

## 1 Introduction 1

1.1 Motivations 2

1.2 Main Contributions 4

1.3 Organization of the Thesis 4

## 2 Background 6

2.1 Agent-Oriented Methodologies 6

2.1.1 $i^*$ framework 7

2.1.2 Tropos 8

2.1.3 AOR 9

2.1.4 Prometheus 11

2.1.5 Gaia 12

2.1.6 Comparison 14

2.2 Non-functional requirements 15

2.3 Constraint Hierarchies 18

2.4 Web Service Composition 20

2.5 Executable Specification 21

2.6 3APL (An Abstract Agent Programming Language) 21

2.7 Service-Oriented Architecture (SOA) 24

2.8 Summary 26

## 3 Executing Agent-Oriented Conceptual Models 27

3.1 Introduction 27

3.2 Executable Specification of $i^*$ framework 28

3.3 Hybrid Modeling 32

3.4 Agent-based prototyping of Service-oriented Architectures 38
3.4.1 Early Requirements Analysis ........................................... 40
3.4.2 Executable Specification .............................................. 45
3.5 Limitations ........................................................................ 48
3.6 Summary ............................................................................ 48

4 Dealing with Non-Functional Requirements using Constraint Hierarchies ................................. 50
4.1 Non-Functional Requirements and Service QoS factors ......................................................... 51
4.2 Dealing with non-functional requirements using constraint hierarchies ................................. 51
4.3 Dealing with Web Service QoS factors using Constraint Hierarchies: The QoSCH model ......................... 56
4.4 Web Services Selection ........................................................ 59
4.5 Web Services Composition .................................................... 61
4.5.1 Related Work .................................................................. 66
4.6 Hierarchical Constraint Logic Programming for Requirements Engineering ....................... 67
4.7 Limitations ........................................................................... 68
4.8 Summary ............................................................................ 68

5 The Case Study ...................................................................... 69
5.1 The Scenario ........................................................................ 69
5.2 Application of QoSCH Model ................................. ................................. 71
5.2.1 Web Service Selection ........................................................... 71
5.2.2 Web Service Composition .................................................... 73
5.3 Agent-based prototyping of Service-Oriented Architectures ..................................................... 83
5.3.1 i* models of Guardian Life Insurance System ...................................................... 83
5.3.2 Executable Specification of Guardian Life Insurance System ........................................... 87
5.4 Summary ............................................................................ 106

6 Conclusion and Future Work ............................................................ 107

Bibliography .......................................................................... 109

A Source Code ......................................................................... 117
A.1 Programs of Guardian Life Insurance System ........................................................................... 117
A.1.1 Clientsystem.3apl ................................................. ................................. 117
A.1.2 Enterpriseservicemanager.3apl ................................................... 118
A.1.3 Claimsystem.3apl .......................................................... 120
A.1.4 Datawarehouse.3apl ......................................................... 121
A.1.5 prolog.pl ........................................................................ 122
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Characteristics of some Methodologies</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Possible Measures for Quality attributes</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Constraint hierarchy for FBTS</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Valuations for constraints variables</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>QoSCH model for Online Payment Service</td>
<td>60</td>
</tr>
<tr>
<td>4.5</td>
<td>Valuations for constraints variables of OPS</td>
<td>60</td>
</tr>
<tr>
<td>4.6</td>
<td>QoSCH Model of Online Shopping Services</td>
<td>64</td>
</tr>
<tr>
<td>4.7</td>
<td>Valuations for constraints variables of RLS&lt;sub&gt;1&lt;/sub&gt; and RLS&lt;sub&gt;2&lt;/sub&gt;</td>
<td>64</td>
</tr>
<tr>
<td>4.8</td>
<td>Valuations for constraints variables of SES&lt;sub&gt;1&lt;/sub&gt; and SES&lt;sub&gt;2&lt;/sub&gt;</td>
<td>64</td>
</tr>
<tr>
<td>4.9</td>
<td>Valuations for constraints variables of SES&lt;sub&gt;3&lt;/sub&gt; and PDS</td>
<td>65</td>
</tr>
<tr>
<td>4.10</td>
<td>QoSCH Model of View Product Services</td>
<td>65</td>
</tr>
<tr>
<td>4.11</td>
<td>Valuations for constraints variables of composite web services RLS&lt;sub&gt;1&lt;/sub&gt; ⊕ VPS ⊕ OPS&lt;sub&gt;1&lt;/sub&gt;, RLS&lt;sub&gt;1&lt;/sub&gt; ⊕ VPS ⊕ OPS&lt;sub&gt;2&lt;/sub&gt;</td>
<td>65</td>
</tr>
<tr>
<td>5.1</td>
<td>QoSCH model for Policyholder Systems</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Valuations for constraints variables of PS</td>
<td>73</td>
</tr>
<tr>
<td>5.3</td>
<td>QoSCH model for Composed Claim System</td>
<td>75</td>
</tr>
<tr>
<td>5.4</td>
<td>Valuations for constraints variables of Claim systems</td>
<td>76</td>
</tr>
<tr>
<td>5.5</td>
<td>Valuations for constraints variables of Benefits plans systems</td>
<td>76</td>
</tr>
<tr>
<td>5.6</td>
<td>Aggregation functions</td>
<td>77</td>
</tr>
<tr>
<td>5.7</td>
<td>Valuations for constraints variables of Claim systems</td>
<td>78</td>
</tr>
<tr>
<td>5.8</td>
<td>QoSCH model for Client system - Web portal</td>
<td>79</td>
</tr>
<tr>
<td>5.9</td>
<td>QoSCH model for Enterprise service manager</td>
<td>80</td>
</tr>
<tr>
<td>5.10</td>
<td>QoSCH model for Data warehouse</td>
<td>80</td>
</tr>
<tr>
<td>5.11</td>
<td>QoSCH model for Guardian Life Insurance System</td>
<td>81</td>
</tr>
<tr>
<td>5.12</td>
<td>Valuations for constraints variables of Web portal</td>
<td>81</td>
</tr>
<tr>
<td>5.13</td>
<td>Valuations for constraints variables of Enterprise service manager</td>
<td>81</td>
</tr>
</tbody>
</table>
5.14 Valuations for constraints variables of Data warehouse . . . . . . . . . 82
5.15 Valuations for constraints variables of Guardian Life Insurance systems 82
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Co-evolution of Hybrid Modeling</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Strategic Dependency Model of online shopping service</td>
<td>43</td>
</tr>
<tr>
<td>3.3</td>
<td>Strategic Rationale Model of online shopping service</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>Web services selection</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Requirements of Online Shopping Service</td>
<td>63</td>
</tr>
<tr>
<td>5.1</td>
<td>Guardian Insurance System</td>
<td>70</td>
</tr>
<tr>
<td>5.2</td>
<td>Elementary Process of Guardian Life Insurance System</td>
<td>74</td>
</tr>
<tr>
<td>5.3</td>
<td>Tree View of Claim System Web Service Composition</td>
<td>77</td>
</tr>
<tr>
<td>5.4</td>
<td>Tree View of Guardian Life Insurance System Composition</td>
<td>82</td>
</tr>
<tr>
<td>5.5</td>
<td>Strategic Dependency Model of Guardian Life Insurance System</td>
<td>84</td>
</tr>
<tr>
<td>5.6</td>
<td>Strategic Rationale Model of Guardian Life Insurance System</td>
<td>86</td>
</tr>
<tr>
<td>5.7</td>
<td>3APL program for Guardian Life Insurance System</td>
<td>104</td>
</tr>
<tr>
<td>5.8</td>
<td>Communication messages</td>
<td>104</td>
</tr>
<tr>
<td>5.9</td>
<td>Communication flow</td>
<td>105</td>
</tr>
</tbody>
</table>