

WEB SERVICE INTERFACES DESIGN FOR E-BUSINESS APPLICATIONS

by

Sia Minh Hong

PhD Thesis

Presented to the Faculty of the Graduate School of

The University of Technology, Sydney

in Partial Fulfilment

of the Requirements

for the Degree of

Doctor of Philosophy in the

School of Software,

Faculty of Engineering and Information Technology

Supervisor: Dr. George Feuerlicht

October 2015

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student:

Date:

ACKNOWLEDGEMENTS

I would like to express my thanks to my principal supervisor, Dr. George Feuerlicht, for his guidance and support during the course of my PhD studies. He gave me much encouragement and ideas, which allowed me to complete this thesis. We worked together and I found his comments and feedback to be very useful.

I also would like to thank my co-supervisor Richard White, who is the CEO of Eagle Datamation International (EDI). He provided me with the opportunity to undertake this research work at the company on several occasions. I gained access to and obtained much valuable commercial experience when designing these Global Trade e-business Web Service interfaces.

Thanks are also extended to the project's funding or partner organisations: Australian Research Council (ARC Linkage Projects) and Eagle Datamation International (UTS contribution to ARC projects). This research thesis may not have happened without their support.

Thanks also go to my family and especially my wife for fully supporting and understanding what this long journey entailed. Thanks also go to Mr. Phillip Thomas who proof-read my thesis and provided me with advice regarding English corrections and expression.

Finally, my thanks go to the examiners for providing their valuable time for reading this thesis.

Table of Content

| | |
|--|------------|
| WEB SERVICE INTERFACES DESIGN FOR E-BUSINESS APPLICATIONS | 1 |
| LIST OF ACRONYMS & ABBREVIATIONS..... | i |
| Abstract | iii |
| Chapter 1 | 1 |
| Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Web Services for e-business..... | 3 |
| 1.3 Web Service Interfaces | 4 |
| 1.4 Web Service Interface Design Issues | 6 |
| 1.5 Granularity Consideration..... | 7 |
| 1.6 Requirements for Web Service interface design..... | 9 |
| 1.7 Problem Identification and Definition..... | 11 |
| 1.8 Thesis Objective | 12 |
| 1.9 Summary of Contributions..... | 14 |
| 1.10 Research Methodology..... | 15 |
| Evaluation..... | 15 |
| 1.11 Structure of the Thesis | 16 |
| Chapter 2 | 17 |
| E-Business Document Standards..... | 17 |
| 2.1 Web Service Invocation Methods..... | 17 |
| 2.2 Web Service with Business Document | 19 |
| 2.3 Normalisation of XML Messages / Business Document | 21 |
| 2.4 Business documents and their design pattern | 22 |
| 2.4.1 Business Object Document (BOD) BY OAGIS..... | 22 |
| 2.4.2 Universal Business Language (UBL) BY OASIS | 25 |
| 2.4.3 Open Travel Alliance (OTA) Business document..... | 28 |
| 2.4.4 XML Common Business Language (xCBL) | 31 |
| 2.4.5 GS1 Messages by global language of business..... | 34 |
| 2.5 Command Pattern Interface | 37 |
| 2.6 Discussion..... | 40 |
| Chapter 3 | 43 |
| Review of existing Web Service Interface Design, Business Document Design and | |
| Web Service Composition Design Methodologies | 43 |
| 3.1 Existing Web Service interface design methodologies | 43 |
| 3.1.1 Web Service design based on Elementary Business Function..... | 43 |
| 3.1.2 Web Service design based on Requirement Analysis..... | 45 |
| 3.1.3 Web Service Design based on Transactional Service | 46 |
| 3.1.4 Web Service design based on Shareable Components..... | 47 |
| 3.1.5 Web Service design based on Data Centric Approach with Factual Dependency | 48 |
| 3.1.6 Generic Web Service Interface design methodology..... | 48 |
| 3.1.7 REST style Web Service design methodology | 49 |
| 3.1.8 Value-Based Service Modeling and Design..... | 51 |
| 3.1.9 Model Driven Design of Web Service Operations using Web Engineering | |
| Practice | 53 |
| 3.1.10 Pragmatic Web Service design approach..... | 54 |

| | |
|--|------------|
| 3.1.11 Evolving Web Service interface design..... | 55 |
| 3.2 <i>Business Document Design</i> | 56 |
| 3.2.1 Web Service design based on Document Engineering..... | 56 |
| 3.3 <i>Existing Web Service Composition design methodology</i> | 60 |
| 3.3.1 Web Service composition..... | 60 |
| 3.3.2 Composition design based on UML | 62 |
| 3.3.3 Composition Design based on Formal Language | 63 |
| 3.3.4 Composition Design based on Case-Based Reasoning | 64 |
| 3.3.5 Composition Design based on RosettaNet PIPs | 64 |
| 3.4 <i>Discussion</i> | 65 |
| Chapter 4 | 70 |
| Proposed Methodology | 70 |
| 4.1 <i>Objective</i> | 70 |
| 4.1.1 Minimalist Interface Design | 71 |
| 4.1.2 Methodology | 71 |
| 4.1.3 Case study | 72 |
| 4.2 <i>Design with proposed solution</i> | 75 |
| 4.3 <i>Advantages and Disadvantages</i> | 82 |
| 4.3.1 Advantages | 82 |
| 4.3.2 Disadvantages | 83 |
| 4.4 <i>Discussion</i> | 84 |
| Chapter 5 | 87 |
| Implementation | 87 |
| 5.1 <i>Implementing Case Study</i> | 87 |
| 5.1.1 Individual Business Process and Requirement | 90 |
| 5.1.2 Proposed Ordering Process..... | 94 |
| 5.2 <i>Implementation Detail</i> | 95 |
| 5.2.1 System Architecture | 95 |
| 5.2.2 Database System | 96 |
| 5.2.3 Application server system | 96 |
| 5.2.4 Application Prototype..... | 97 |
| Chapter 6 | 100 |
| Evaluation | 100 |
| 6.1 <i>Service-Oriented Design Principles</i> | 100 |
| 6.2 <i>Quantitative Evaluation</i> | 103 |
| 6.3 <i>Qualitative Evaluation</i> | 107 |
| 6.3.1 Reusability Principle..... | 107 |
| 6.3.2 Business Suitability | 110 |
| 6.3.3 Abstraction Principle..... | 111 |
| 6.3.4 Extensibility | 113 |
| 6.3.5 Flexibility..... | 116 |
| 6.3.6 Maintainability..... | 117 |
| 6.3.7 Comparison of the outcomes..... | 119 |
| 6.4 <i>Discussion</i> | 119 |
| Chapter 7 | 121 |
| Conclusion | 121 |
| 7.1 <i>Research Summary</i> | 121 |
| 7.2 <i>Research Contributions</i> | 124 |
| 7.3 <i>Future Work</i> | 126 |

| | |
|---|------------|
| References | 127 |
| Appendices | 137 |
| <i>Appendix 1: Script to generate Database Table for case study</i> | <i>137</i> |
| <i>Appendix 2: Available Web Method as Web Service.....</i> | <i>139</i> |
| <i>Appendix 3: Persistence Unit for the Prototype Application.....</i> | <i>140</i> |
| <i>Appendix 4: Business Entities of the Prototype application.....</i> | <i>141</i> |
| <i>Appendix 5: Data Access Object of the Prototype application</i> | <i>157</i> |
| <i>Appendix 6: Business Value Object of the Prototype application.....</i> | <i>161</i> |
| <i>Appendix 7: Purchase Order Schema of the Prototype application</i> | <i>183</i> |
| <i>Appendix 8: Object Serializer of the Prototype application.....</i> | <i>185</i> |
| <i>Appendix 9: Prototype application Web Services</i> | <i>192</i> |
| <i>Appendix 10: Prototype application Web Service Interface.....</i> | <i>193</i> |
| <i>Appendix 11: Client's GUI of the Prototype application.....</i> | <i>195</i> |
| <i>Appendix 12: Overview of the Project Prototype application.....</i> | <i>201</i> |
| <i>Appendix 13: Web Service interface showing available ports and operations.....</i> | <i>202</i> |
| <i>Appendix 14: Sample UBL 2.0 Purchase Order example (Source: OAGIS)</i> | <i>203</i> |
| <i>Appendix 15: Sample UBL 2.0 Purchase Order Response example (Source: OAGIS).....</i> | <i>207</i> |
| <i>Appendix 16: Sample UBL 2.0 Change Purchase Order Response example (Source: OAGIS)</i> <i>.....</i> | <i>210</i> |
| <i>Appendix 17: Sample UBL 2.0 Purchase Order Cancellation example (Source: OAGIS) ..</i> | <i>229</i> |

LIST OF ACRONYMS & ABBREVIATIONS

| Abbreviation | |
|--------------|--|
| ABIE | Aggregate Business Information Entity |
| ACC | Aggregate Core Component |
| B2B | Business-to-business |
| BBC | Basic Core Component |
| BBIE | Basic Business Information Entity |
| BIE | Business Information Entity |
| BOD | Business Object Document |
| BPEL | Business Process Execution Language |
| CBL | Common Business Library |
| CCL | Core Component Library |
| CCT | Core Component Type |
| CCTS | Core Components Technical Specification |
| CEFACT | Centre for Trade Facilitation and Electronic Business |
| ebXML | Electronic Business eXtensible Markup Language |
| EDI | Electronic Data Interchange |
| EDIFACT | Electronic Data Interchange For Administration, Commerce and Transport |
| EPC | Electronic Product Code |
| GDD | Global Data Dictionary |
| GDT | Global data type |
| GS1 | Global Standards One |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Secured HyperText Transfer Protocol |
| IATA | International Air Transport Association |
| ISO | International Organization for Standardization |
| MIME | Multipurpose Internet Mail Extension |
| NDR | Naming and Design Rules |

Web Service interfaces design for e-business applications

| | |
|--------|--|
| OAGI | Open Applications Group, Inc. |
| OAGIS | Open Applications Group Integration Specification |
| OASIS | Organization for the Advancement of Structured Information Standards |
| OTA | Open Travel Alliance |
| OWL | Web Ontology Language |
| UBL | Universal Business Language |
| UDDI | Universal Description, Discovery and Integration |
| UMM | UN/CEFACT Modelling Methodology |
| WS-CAF | Web Services Composite Application Framework |
| WSDL | Web Services Description Language |
| xCBL | XML Common Business Library |
| XML | Extensible Markup Language |
| XSD | XML Schema |
| XSL | Extensible Stylesheet Language |
| XSLT | Extensible Stylesheet Language Transformations |

ABSTRACT

As a result of the rapid developments in Web Service standards and technologies during the last decade, many organisations are implementing applications using Web Services. Some organisations are making significant commitments to Web Service standards and technology platforms. Successful projects using Web Services will to a large extent depend on the effective design and development methodologies used in the construction of an e-business application. While the importance of application design in general is recognised, so far only limited attention has been paid to design issues for service-oriented e-business applications. Currently there are no comprehensive methodologies for designing service interfaces.

The traditional e-business interoperability approach is for business partners to interchange the industry standard business documents or XML messages (i.e. UBL, OTA). This approach is complex and inefficient because the business document is large and results in many optional and repeated elements that are redundant. Developing Web Services for e-business is time-consuming due to the considerable effort required to define the interfaces and maintaining a large volume of standard business documents.

This work proposes to use the minimalist design approach to optimise a set of standard business documents and interfaces. The proposed interface is exposed as an abstract layer to the external parties and is able to process multiple actions corresponding to a business document. The method is based on analysing existing business documents, identifying the key elements responsible for an operation, and then exposing the operation interface that corresponds to the business document rather than the business event. This can be achieved by inserting business event action elements into the XML schema. Doing so will not only reduce the number of operation interfaces but also

increase the Web Service interface's flexibility and extendibility. Web Service implementation projects conducted in the absence of a design framework are likely to suffer from poor reuse and extensibility. In order to achieve reusability, this method enables the operation to be invoked based on individual action or multiple actions for the same interface. This is because a single action operation typically represents a fine-grained business task. Consequently, the interface is always extendable due to using multiple actions in the operation.

Finally, this thesis will detail the above mentioned methodology to optimise e-business Web Service interface to become more flexible, reusable and extendable. We will illustrate the design methodology using a purchase order business process example based on the Universal Business Language specification in order to demonstrate its effectiveness.