

---

---

# **Customer Profiling Using a Service- Oriented Architecture**

**Felix Ntawanga**

Supervisors: Prof. A. P. Calitz and Dr. L. Barnard

April 2010

---

---

Submitted in partial fulfilment of the requirements for the degree  
Magister Scientiae in Computer Science and  
Information Systems at the  
[Nelson Mandela Metropolitan University](#)

# ACKNOWLEDGEMENTS

First and foremost would like thank the Almighty God for giving me everything I am, and the following:

- My supervisors: Professor André Calitz and Doctor Lynette Barnard for the long hours they spent reading the dissertation and for the support and guidance rendered during my studies;
- The Department of Computing Sciences at Nelson Mandela Metropolitan University (NMMU), especially the lecturers, the Systems and Networks Administrator, Mr. Jean Rademakers, and CoE Assistant, Ms. Desma van De Walt for the assistance rendered during my course and the research;
- Mr. Danie Venter, Senior Lecturer and Statistical Consultant at the NMMU, for providing valuable input for statistical data analyses for the research;
- Computer Science Masters students especially Mr. Emile Senga and Mr. Ryan Hill for their support and encouragement;
- Mr. Bradley van Tonder for technical programming assistance and for being a role model;
- Dr. Patrick Goldstone for proof reading and providing valuable input that improved the quality of this dissertation;
- Special thanks to all the participants who assisted during the evaluation;
- All my friends wherever they are for wishing me well and encouraging me to face the challenge; and
- My family for the continued love and support.

May God bless you all.

# ABSTRACT

Customer profiling has recently gained much recognition in the e-commerce domain because of the benefits it is capable of bringing to online business. Customer profiling has been implemented in various systems development approaches such as in a client-server environment. Recently there has been an increase in the number of organisations adopting and implementing e-commerce systems using service-oriented architecture (SOA) principles. This research set out to determine how a customer profile can be implemented using open source SOA implementation tools, and how SOA-based customer profiles can be utilised to provide appropriate personalisation in an SOA environment. The research further endeavoured to complete a comparative study on customer profile implementation in two different architectures, namely SOA and client-server.

An extensive literature review was conducted on SOA, customer profiling and e-commerce systems development. SOA enabling technologies, such as, web services, enterprise service bus (ESB) and open source Sun Java SOA implementation tools, for example, Open ESB, GlassFish application server and Netbeans IDE were analysed. A Java web services-based customer profiling system was prototyped following SOA design principles. An end-user evaluation survey was conducted using eye tracking with a sample of 30 participants. The evaluation was done on two e-commerce systems with the same interface but running on two different customer profile back-ends, SOA and client-server. The results show that participants did not experience significant difference between the two systems, however, eye tracking results showed a significant difference between the two systems.

The research concluded that customer profiling using SOA offers more benefits than implementations using other architectures such as client-server. SOA component-based development proved to be easier to manage, develop, integrate and improves interoperability between different technologies. The research brought together necessary techniques and technologies that organisations can use to implement SOA. Using SOA, organisations can integrate and utilise different technologies seamlessly to achieve business goals.

**Keywords:** Customer profiling, web services, service-oriented architecture (SOA), client-server, personalisation, e-commerce, open source SOA implementation tools.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	i
ABSTRACT .....	ii
TABLE OF CONTENTS .....	iii
LIST OF FIGURES .....	x
LIST OF TABLES .....	xii
LIST OF ABBREVIATIONS .....	xiii
CHAPTER 1: INTRODUCTION .....	1
1.1 BACKGROUND .....	1
1.2 RELEVANCE OF THE RESEARCH .....	3
1.3 PROBLEM STATEMENT .....	3
1.4 RESEARCH OBJECTIVES .....	4
1.5 RESEARCH QUESTIONS .....	5
1.6 RESEARCH METHODOLOGY .....	5
1.6.1 Literature study .....	6
1.6.2 Prototyping .....	6
1.6.3 Evaluation survey .....	6
1.6.4 Data collection and analysis .....	6
1.7 RESEARCH SCOPE .....	7
1.8 SUMMARY AND DISSERTATION OUTLINE .....	7
CHAPTER 2: CUSTOMER PROFILING IN E-COMMERCE .....	10
2.1 INTRODUCTION .....	10
2.2. E-COMMERCE SYSTEMS' ARCHITECTURE AND IMPLEMENTATION .....	11
2.2.1 CORBA .....	11
2.2.2 DCOM .....	12
2.2.3 JAVA RMI .....	12
2.2.4 Problems identified .....	12
2.3 CUSTOMER PROFILING .....	13
2.3.1 Definition .....	13

2.3.2 Customer profile implementation .....	14
2.3.3 Examples of customer profile data storage and maintenance .....	17
2.4 POTENTIAL USES OF CUSTOMER PROFILES IN E-COMMERCE .....	18
2.4.1 E-commerce personalisation .....	19
2.4.2 E-commerce customisation .....	21
2.4.3 E-commerce recommendation .....	22
2.5 SUMMARY .....	23
CHAPTER 3: SERVICE-ORIENTED ARCHITECTURE.....	24
3.1 INTRODUCTION .....	24
3.2 DEFINITION OF SOA.....	24
3.2.1 Service-orientation and services .....	25
3.2.2 Business Process Management (BPM) .....	25
3.2.3 Notable service-oriented applications .....	26
3.2.3.1 Software as a Service (SaaS) .....	26
3.2.3.2 Cloud computing.....	28
3.3 THE IMPLEMENTATION OF SYSTEMS USING SOA.....	30
3.4. SERVICES INTERACTION IN SOA.....	31
3.5 SOA ENABLING TECHNOLOGIES.....	31
3.5.1 Web services .....	32
3.5.2 Enterprise Service Bus (ESB).....	33
3.5.3 Standard technology and protocols used in SOA .....	34
3.5.3.1 SOAP .....	34
3.5.3.2 Web Service Description Language (WSDL).....	34
3.5.3.3 Universal Description, Discovery and Integration (UDDI).....	35
3.5.3.4 Extensible Mark-up Language (XML) .....	36
3.5.4 Java Business Integration (JBI) .....	36
3.5.5 Business Process Execution Language (BPEL).....	37
3.6 BENEFITS OF SOA.....	39

3.6.1 Improved integration (interoperability) .....	39
3.6.2 Inherent re-use .....	39
3.6.3 Leveraging of legacy investment .....	39
3.6.4 Best-of-breed alternatives .....	39
3.6.5 Organisational agility .....	40
3.7 SOA CHALLENGES .....	40
3.7.1 Autonomy and interoperability .....	40
3.7.2 Security .....	41
3.8 SOA GOVERNANCE .....	41
3.9 ORGANISATIONS WORKING TOWARDS STANDARDISATION IN SOA .....	42
3.9.1 W3C .....	42
3.9.2 OASIS .....	42
3.9.3 WS-I .....	43
3.10 SUMMARY .....	43
CHAPTER 4: THE ANALYSIS AND DESIGN OF A CUSTOMER PROFILING SYSTEM USING SOA .....	44
4.1 INTRODUCTION .....	44
4.2 SERVICE-ORIENTED ANALYSIS AND DESIGN .....	44
4.2.1 Service identification .....	46
4.2.2 Service specification .....	49
4.2.3 Service realisation .....	51
4.3 CUSTOMER PROFILE SERVICE-ORIENTED ANALYSIS AND DESIGN .....	53
4.3.1 Customer profile services identification .....	53
4.3.1.1 Register customer service .....	54
4.3.1.2 Add customer service .....	55
4.3.1.3 Get profile service .....	55
4.3.1.4 Update profile service .....	55
4.3.1.5 Identify customer service .....	56

4.3.1.6 Delete customer service .....	56
4.3.2 Customer profile services specification .....	57
4.4 CUSTOMER PROFILE DATA MODEL .....	59
4.5 CUSTOMER PROFILE SERVICES CLIENTS .....	60
4.6 OBJECT-ORIENTED CUSTOMER PROFILING SYSTEM .....	61
4.7 CUSTOMER PROFILE MODEL USING SOA .....	62
4.7.1 Model definition.....	62
4.7.2 UML.....	62
4.7.3 Customer profile UML composite and deployment model .....	63
4.8 SOA DESIGN PRINCIPLES .....	66
4.9 SECURITY DESIGN .....	67
4.10 SUMMARY .....	68
CHAPTER 5: CUSTOMER PROFILING SYSTEM IMPLEMENTATION USING SOA...	69
5.1 INTRODUCTION .....	69
5.2 IMPLEMENTATION TOOLS .....	69
5.2.1 SOAIFs .....	70
5.2.1.1 SOAIF evaluation procedure .....	72
5.2.1.2 SOAIF evaluation metrics.....	72
5.2.1.3 Evaluation results.....	73
5.2.1.4 Limitations of the evaluation .....	74
5.2.2 Back-end and front-end implementation tools.....	74
5.2.2.1 Netbeans IDE .....	74
5.2.2.2 Microsoft Visual Studio IDE .....	75
5.2.2.3 Microsoft SQL Server Express .....	76
5.3 IMPLEMENTATION .....	78
5.3.1 Customer profile web services implementation.....	78
5.3.2 Customer profile web services clients' implementation.....	81
5.3.3 Database implementation.....	83

5.4 E-COMMERCE WEB APPLICATION AND CUSTOMER PROFILE WEB SERVICES INTERACTION .....	83
5.4.1 Creating a link between service consumer and service producer in Microsoft Visual Studio.....	85
5.4.2 Utilising a web service in Microsoft Visual Studio .....	85
5.5 SUMMARY .....	86
CHAPTER 6: EVALUATION AND ANALYSIS OF RESULTS .....	87
6.1 INTRODUCTION .....	87
6.2 NMMU DEPARTMENT OF COMPUTING SCIENCES USABILITY LABORATORY .....	87
6.3 EVALUATION INSTRUMENTS.....	89
6.3.1 Questionnaires.....	89
6.3.1.1 Background questionnaire .....	89
6.3.1.2 Post-task evaluation questionnaire first system .....	90
6.3.1.3 Post-task evaluation questionnaire second system .....	91
6.3.2 Eye tracking .....	91
6.3.3 Statistical instruments .....	93
6.3.3.1 Descriptive statistics .....	93
6.3.3.2 Inferential statistics .....	94
6.3.4 Results representation .....	95
6.4 EVALUATION OBJECTIVES .....	95
6.5 EVALUATION.....	96
6.5.1 Pilot study evaluation.....	96
6.5.1.1 Sampling .....	97
6.5.1.2 Evaluation procedure .....	98
6.5.1.3 Pilot study results.....	98
6.5.1.4 Design and implementation changes .....	100
6.5.2 Main study evaluation.....	100
6.5.2.1 Sampling .....	100



6.5.2.2 Evaluation procedure .....	104
6.5.2.3 Eye tracking and questionnaire data collection .....	105
6.6 DATA ANALYSIS .....	106
6.7 ANALYSIS OF RESULTS .....	106
6.7.1 Descriptive statistical results.....	106
6.7.1.1 Completeness of customer profile components in SOA .....	107
6.7.1.2 Comparison between SOA and CS back-end systems.....	109
6.7.1.3 Eye tracking results.....	113
6.7.2 Inferential statistical results .....	121
6.7.3 Qualitative results .....	124
6.7.3.1 Positive comments .....	124
6.7.3.2 Negative comments.....	125
6.8 SOA IMPLEMENTATION EVALUATION .....	126
6.9 SUMMARY .....	127
CHAPTER 7: CONCLUSIONS .....	128
7.1 INTRODUCTION .....	128
7.2 RESEARCH OBJECTIVES REVISITED .....	129
7.2.1 Literature study .....	130
7.2.2 Prototyping.....	130
7.2.3 Evaluation .....	131
7.3 RESEARCH ACHIEVEMENTS.....	132
7.3.1 Theoretical achievements.....	132
7.3.1.1 Research conducted in customer profiling and SOA.....	132
7.3.1.2 Customer profile model using SOA.....	132
7.3.2 Practical achievements.....	132
7.3.3 Major research findings .....	133
7.4 RESEARCH CONTRIBUTIONS .....	133
7.4.1 Theoretical contributions .....	133

7.4.2 Practical contributions .....	134
7.5 RECOMMENDATIONS FOR FUTURE RESEARCH.....	135
7.6 PROBLEMS ENCOUNTERED .....	135
7.7 SUMMARY .....	135
REFERENCES .....	137
APPENDICES.....	148
Appendix A: Pilot Study Evaluation Tasks .....	148
Appendix B: Pilot Study Background and Post-Task Questionnaire.....	149
Appendix C: Main Evaluation Procedure Outline .....	151
Appendix D: Consent Form .....	152
Appendix E: Background Questionnaire .....	153
Appendix F: Main Study Evaluation Tasks .....	155
Appendix G: Post-Task Evaluation Questionnaire First System .....	156
Appendix H: Post-Task Evaluation Questionnaire Second System.....	158
Appendix I: Main Evaluation Results .....	161
Appendix J: Conference Proceedings Abstracts .....	165

# LIST OF FIGURES

Figure 1.1: Research methodology .....	7
Figure 2.1: Explicit feedback customer profiling method (kalahari.net, 2009).....	15
Figure 2.2: Product knowledge questionnaire .....	16
Figure 2.3: Personalised apartment information based on customer profile (Shearin and Lieberman, 2001) .....	20
Figure 2.4: Customisation of a bbc.co.uk homepage (BBC, 2008) .....	21
Figure 2.5: Recommendation page for a particular customer (Amazon.com, 2009).....	23
Figure 3.1: Salesforce CRM service page (Salesforce.com, 2009) .....	27
Figure 3.2: Zoho online invoicing service (Zoho.com, 2009) .....	28
Figure 3.3: Microsoft Azure Platform service (Microsoft.com, 2009).....	29
Figure 3.4: SOA reference architecture (IBM, 2007) .....	30
Figure 3.5: SOA's Find-Bind-Execute paradigm (Mahmoud, 2005; Erl, 2005).....	31
Figure 3.6: Meta model of web service architecture (W3C, 2004) .....	33
Figure 3.7: An example of a web service description language .....	35
Figure 3.8: JBI components .....	37
Figure 3.9: Expanded JBI components from Netbeans IDE 6.5 .....	37
Figure 3.10: BPEL Structure.....	38
Figure 4.1: Service specification framework (Terlouw and Maarse, 2009) .....	50
Figure 4.2: Summary of the service-oriented modeling architecture (Arsanjani, 2004) .....	52
Figure 4.3: Customer profile model using SOA .....	64
Figure 5.1: Apache ServiceMix ESB (Apache.org, 2009).....	71
Figure 5.2: Open ESB architecture (Java.sun.com, 2009).....	71
Figure 5.3: GlassFish application server console .....	73
Figure 5.4: Netbeans IDE 6.5 interface .....	75
Figure 5.5: Microsoft Visual Studio 2008 interface .....	76
Figure 5.6: A general outline of customer profile web services .....	79
Figure 5.7: Web service testing page .....	80
Figure 5.8: An e-commerce application red wine page .....	82
Figure 5.9: SOAP request and response .....	84
Figure 5.10: Accessing web services methods .....	85
Figure 6.1 NMMU Department of Computing Sciences Usability Laboratory observer and participant rooms .....	88

Figure 6.2: NMMU Department of Computing Sciences Usability Laboratory Tobii T60 Eye tracker .....	92
Figure 6.3: Pilot study evaluation system usability results [n=8] .....	99
Figure 6.4: Participants' age range [n=29] .....	102
Figure 6.5: Participants' position [n=29] .....	102
Figure 6.6: Participants' ethnicity .....	102
Figure 6.7: Participants' gender .....	103
Figure 6.8: Customer profile components [n=29] .....	108
Figure 6.9: After scenario questionnaire results [n=29] .....	109
Figure 6.10: Systems' response and task completion comparison [n=29] .....	110
Figure 6.11: Difference between SOA and CS rating [n=29] .....	111
Figure 6.12: General comparison between SOA and CS systems [n=29] .....	112
Figure 6.13: Overall assessment difference between SOA and CS systems [n=29] .....	113
Figure 6.14: Eye tracking page loading times SOA vs. CS [n=29] .....	114
Figure 6.15: Eye tracking page loading times SOA vs. CS line graph [n=29] .....	115
Figure 6.16: Eye tracking page loading time SOA first [n=13] .....	115
Figure 6.17: Difference in eye tracking page loading time SOA first [n=13] .....	116
Figure 6.18: Eye tracking page loading times CS first [n=16] .....	116
Figure 6.19: Difference in eye tracking page loading times CS first [n=16] .....	117
Figure 6.20: Heat map for participants CS first [n=16] .....	118
Figure 6.21: Heat map for participants SOA second [n=16] .....	119
Figure 6.22: Heat map for participants SOA first [n=13] .....	120
Figure 6.23: Heat map for participants CS second [n=13] .....	120
Figure 7.1: Customer profiling in an SOA environment .....	134

## LIST OF TABLES

Table 1.1: Research questions, methods and deliverables .....	5
Table 2.1: Common parameters used to define factual information of a customer .....	17
Table 4.1: Customer profile services specification .....	58
Table 4.2: Table structure and data types for the customer profile database .....	60
Table 5.1: Web pages and customer profile service clients' link .....	84
Table 6.1: Computer and Internet experience in years .....	97
Table 6.2: Online shopping and registration experience .....	97
Table 6.3: Computer and Internet experience in years .....	103
Table 6.4: t-test results for participants' perceived differences between the SOA and CS systems .....	122
Table 6.5: t-test results for eye tracking page loading times of the two systems SOA and CS systems .....	123
Table 6.6: SOA implementation evaluation .....	126

# LIST OF ABBREVIATIONS

Amazon S3	Amazon Simple Storage Service
Amazon SQS	Amazon Simple Queue Service
API	Application Programming Interface
ASP	Active Server Pages
AWS	Amazon Web Services
BBC	British Broadcasting Corporation
BPEL	Business Process Execution Language
BPM	Business Process Management
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
DBMS	Database Management System
DCOM	Distributed Component Object Model
EC2	Elastic Computing Cloud
EJB	Enterprise Java Beans
ESB	Enterprise Service Bus
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transmission Protocol
IBM	International Business Machines
IDE	Integrated Development Environment
ISV	Independent Software Vendor
IT	Information Technology
JAX-WS	Java API for XML Web Services
JB1	Java Business Integration
JDBC	Java Database Connectivity
JDK	Java Development Kit
JSON	JavaScript Object Notation
JMS	Java Message Service
JWS	Java Web Service
NMMU	Nelson Mandela Metropolitan University
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
REST	Representational State Transfer

SaaS	Software as a Service
SDK	Software Development Kit
SOA	Service-Oriented Architecture
SOAIF	Service-Oriented Architecture Implementation Framework
SOAP	Simple Object Access Protocol
SOMA	Service Oriented Modeling Architecture
TCP/IP	Transmission Control Protocol / Internet Protocol
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language
USA	United States of America
W3C	World Wide Web Consortium
WSA	Web Services Architecture
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organisation
XML	Extensible Markup Language

# CHAPTER 1: INTRODUCTION

## 1.1 BACKGROUND

The Internet has changed the way people conduct business activities in more ways than any other technology in the history of business (Schneider, 2007). The ever-increasing number of innovations in the Internet, and other related technologies, has led to many businesses seeking new means of efficiently communicating with their customers, taking advantage of the latest Internet technologies. One method that is proving to be helpful is electronic commerce (e-commerce) or electronic business (e-business) (Schneider, 2007).

E-commerce can be defined as the use of Internet technologies in business processes (IBM, 2007; Schneider, 2007). Examples of business processes that can use e-commerce include sales and procurement (Schneider, 2007).

Businesses are adopting e-commerce; and indications are that online business is going to continue experiencing this remarkable growth. Research has shown that e-commerce is more competitive than conventional commerce, since e-commerce has low entrance barriers for new players thus, effectively levelling the playing field (Chang, Changchien and Huang, 2006). One-to-one marketing strategies, such as personalisation have proved effective in retaining and attracting online customers and giving e-commerce businesses a competitive advantage.

Personalisation can be defined as the use of relevant and available information to align business offerings towards customers' requirements. Online businesses take advantage of personalisation technologies to make their products and services unique and to tailor their products and services for customers' specific needs and preferences (Chang, *et al.*, 2006). Research has revealed that personalisation, among other benefits, can improve customers' satisfaction levels, loyalty and can consequently lead to increased sales.

Personalisation is based on the information known about the customer. This is contained in a customer profile (Adomavicius and Tuzhilin, 2005).

Recent studies have indicated that many e-commerce businesses have come to recognise the importance of building personalised profiles of individual online customers (Adomavicius and Tuzhilin, 2005). Customer profiles contain information about a customer's demographic details, preferences, characteristics and activities (Eirinaki and Vazirgiannis, 2003). Customer



profiles can help online businesses perform effective personalisation; and can further enable personal recommendations to be made (Liu, Lin, Chen and Huang, 2001).

E-commerce systems are normally implemented as distribution systems, using a number of architectures, for example, Client-Server, Distributed Component Object Model (DCOM) and Common Object Request Broker (CORBA) (Rajput, 2000). Online businesses are now adopting an SOA to build e-commerce systems and streamline e-commerce activities between online businesses and customers (Ordanini and Pasini, 2008; Amazon.com, 2009; IBM, 2007). SOA can be regarded as the natural evolution of previous architectural styles such as DCOM and CORBA (IBM, 2007).

SOA is defined as a set of patterns and guidelines for creating loosely coupled, standards-based and business-aligned services that provide a new level of flexibility in responsiveness to business threats and opportunities (Canfora, Fasolino, Frattolillo and Tramontana, 2008; Maurizio, Sager, Jones, Corbitt and Girolami, 2008; IBM, 2007). An SOA is made up of a collection of services; and these can be implemented using several different techniques. Moreover, web services are the most commonly used implementation techniques for SOA in the e-commerce domain (Pastore, 2008).

Current SOA models and systems implemented using SOA do not emphasise customer profiling and personalisation (Amazon.com, 2009; IBM, 2007; Erl, 2005; W3C, 2004; Balke and Wagner, 2003; Kuno and Sahai, 2002). Furthermore, no evidence could be found of online customers' preferences for elicitation techniques incorporated in existing SOA models (IBM, 2007; W3C, 2004). Customer profiles provide a basis from which online customers' preferences can be determined. Research has revealed many benefits from the use of customer profiles by online businesses (Adomavicius and Tuzhilin, 2005).

This research study aims to investigate how a customer profile model can be implemented in an SOA. SOA research projects are currently on the increase; however, little attention is being given to the role customer profiles can play in an SOA environment (IBM, 2007; W3C, 2004). This research will analyse and evaluate the available open source SOA implementation frameworks and tools needed to aid in identifying a suitable framework for this study. Furthermore, the research will complete a comparative study in customer profile implementation in SOA and client-server architecture.

## 1.2 RELEVANCE OF THE RESEARCH

Businesses are now adopting SOA to design and develop their mission-critical systems. A recent survey conducted by Computer Economics indicated that SOA adoption rates in the United States of America (USA) increased in 2008 to 58%, compared with 17% in 2007 (Computer Economics, 2009). This suggests that SOA is the cutting-edge technology that organisations are implementing to align business goals with IT. SOA adoption, however, comes along with numerous challenges (Kulkarni and Dwivedi, 2008).

According to Ingesby, 90% of problems at Telkom South Africa are SOA-related (2009). Incorporating customer profiling into SOA e-commerce systems is vital for survival and success in the competitive web environment.

One important factor that has received relatively little attention in e-commerce personalisation is the impact of the preference-elicitation process. This comprises the procedures used to capture customers' likes and dislikes (Gretzel and Fesenmaier, 2006). Effective and accurate customer-preference elicitation techniques can support effective communication between an online business and customers and can improve personalisation.

Customer profiles are customer preference repositories that can be used by online businesses to pre-determine online customers' needs and preferences.

The main aim of this research is to gain both a theoretical and a technical understanding in customer profiling and SOA. Studies on how online businesses implement and utilise customer profiles will be done. An investigation into the recent paradigm of SOA, its enabling technologies, such as ESB and web services, and available implementation frameworks will be conducted.

The research will propose a usable and extendable customer profile model, using SOA as a confirmation of this concept. A further goal is to determine key customer profile implementation aspects in the SOA environment, as compared with other existing environments, such as client-server.

## 1.3 PROBLEM STATEMENT

Researchers in e-commerce acknowledge the need to understand customers' desires, prior to providing them with products and services (Martin-Guerrero, Palomares, Balaguer-Ballester, Soria-Olivas, Gomez-Sanchis and Soriano-Asensi, 2006; Srivihok and Sukonmanee, 2005).

Customer profiles provide a basis for understanding customers (Adomavicius and Tuzhilin, 2005).

Recent studies on SOA conducted by major IT research firms, such as Computer Economics and Forrester Research, indicate that SOA is here to stay (Computer Economics, 2009). However, there has been little effort in customer profiling and personalisation in an SOA environment as compared with the efforts in an ordinary e-commerce environment (Kim, Lee, Shaw, Chang, Nelson and Easley, 2006; Kim and Lee, 2005; Erl, 2005; W3C, 2004; Balke and Wagner, 2003; Kuno and Sahai, 2002).

Furthermore, many organisations that have adopted SOA have been experiencing a number of problems; and some have failed to adopt and succeed with SOA (Veeraragadoo, 2008). Irreconcilable SOA literature and differing approaches to SOA have both been noted at vendors such as Microsoft and IBM (Zapthink, 2009).

This research study will fill these gaps by investigating how a generic customer profile can be implemented using an SOA. The research will endeavour to establish essential business and IT practices for a successful SOA implementation with a customer profile as a confirmation of this concept. Furthermore, the research aims to complete a comparative analysis of customer profiling implementation using SOA and other architectures, such as client-server.

## 1.4 RESEARCH OBJECTIVES

The research aims to find an effective technique for developing and implementing a customer profile model using SOA. Such a model would need to be suitable for use by e-commerce businesses. Specific research objectives include the following:

- To understand customer profiling techniques and models used in businesses;
- To gain understanding of current SOA models, systems and related technology, for example, web services and open source SOA implementation tools;
- To design and implement a suitable customer profile model in an SOA; and
- To complete a comparative study between customer profile implementation, using an SOA and client-server.

## 1.5 RESEARCH QUESTIONS

The primary research question for this research was:

*How can one implement a customer profile using an SOA?*

Several secondary goals have been structured from the research objectives discussed in Section 1.4. Achieving secondary goals will enable a successful resolution of the primary goal. Table 1.1 gives an indication of the secondary research questions, the research methodology to be used to answer a particular question and the deliverables expected from each research question.

No.	Research Question	Research Method	Deliverable
1	What is customer profiling and how do online businesses implement and utilise customer profiles?	Literature study	Chapter 2
2	What is SOA, which SOA models, implementation frameworks, tools and SOA- enabling technologies exist? What are SOA benefits and challenges?	Literature study	Chapter 3
3	What customer profile services and appropriate SOA technology can be used to define and implement a customer profile in an SOA environment?	Literature study and evaluation survey	Chapter 4
4	How can an effective customer profile model be implemented using a selected open source SOA implementation framework?	Prototyping	Chapter 5
5	How can a customer profile be implemented using an SOA, and then be utilised to benefit an online business?	Literature study and prototyping	Chapter 6
6	What are the key aspects of customer profile implementation in SOA as compared with other distributed environments?	Evaluation survey, literature study and arguments	
7	What conclusions, recommendations and future research could follow this research?	Arguments	Chapter 7

Table 1.1: Research questions, methods and deliverables

## 1.6 RESEARCH METHODOLOGY

The primary research methodology will be prototyping, and that will be supported by a literature study and evaluation surveys. Table 1.1 is a summary of the research design showing the different research methods that will be employed in this research and the expected outcomes. The research will source information from different fields of study, for example, Business Management. Consequently, several research methods need to be

employed. This section discusses the research design to be followed in this project. Subsequent chapters of the dissertation will follow the research design discussed in this section.

### **1.6.1 Literature study**

A literature study on e-commerce customer profiling and SOA will be conducted. The literature study will provide a significant understanding of these two aspects of the research.

### **1.6.2 Prototyping**

A customer profile that the research proposed will be prototyped using open source SOA implementation tools and Microsoft .NET framework tools. The implementation tools to be used include Netbeans integrated development environment (IDE) 6.5, GlassFish application server V2, Microsoft Visual Studio 2008 and Microsoft SQL Server 2008 Express. Open ESB will be used as the SOA implementation framework for the research.

### **1.6.3 Evaluation survey**

An end-user evaluation survey will be conducted on the completed prototype to collect data to assess whether the research has met its targets. Tasks will be set in accordance with the different goals of evaluation; and questionnaires will be used to collect both qualitative and quantitative data.

### **1.6.4 Data collection and analysis**

Qualitative and quantitative data will be gathered and will need to be analysed to help attain the goals of the research (Leedy and Ormrod 2001). Relevant eye tracking data will also be collected to provide additional evaluation data. Microsoft Excel 2007 will be used to analyse these data. Figure 1.1 summarises the steps that will be followed during the course of the research.

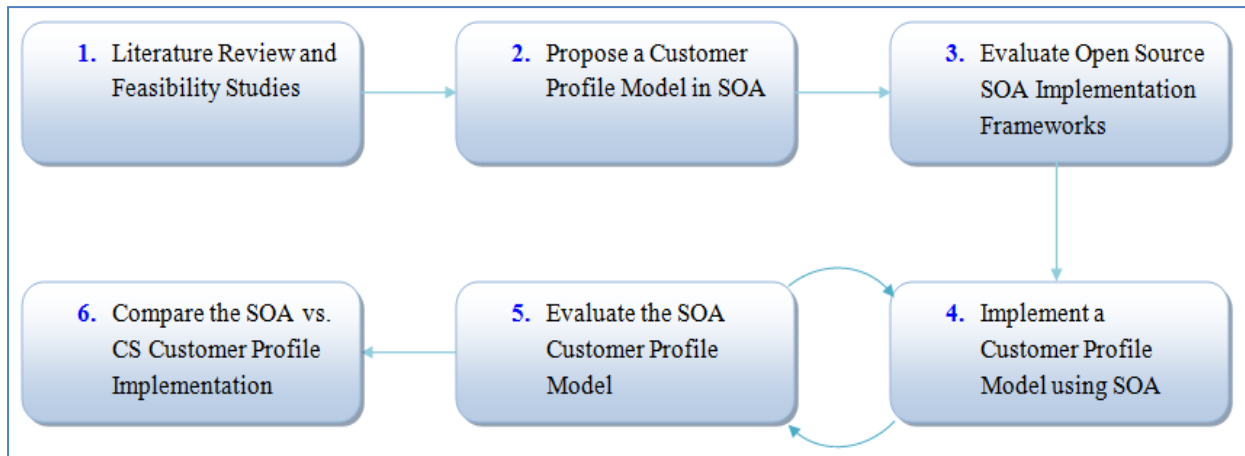


Figure 1.1: Research methodology

## 1.7 RESEARCH SCOPE

The scope of this research is limited to investigating how SOA design principles can be used to implement a customer profile for e-commerce applications using open source SOA implementation tools. Part of the research will proceed to do a comparative study on customer profile implementation in SOA and a client-server environment, from Usability and Software Engineering perspectives. The main focus will be customer profiles for online customers and how e-commerce businesses can implement and utilise online customers' profiles in an SOA environment.

## 1.8 SUMMARY AND DISSERTATION OUTLINE

This chapter has set the scene by presenting an introduction to the research by explaining the trends in e-commerce customer profiling and the existing architectures used to implement e-commerce systems. An introduction to the recent paradigm of SOA has also been presented. The problem that the research aims to solve has been discussed. Specifically this involves the need to investigate how a customer profile can be implemented in an SOA environment. The necessity for this research and its contribution to the body of knowledge has also been identified.

The primary research objective and specific research questions have also been discussed. The specific research questions will assist in the step-by-step realisation of the research objectives. The chapter has also discussed the research design that will be employed in this research in order to achieve its goals. This will include a literature study, prototyping and an evaluation survey. Subsequent chapters of the dissertation have been structured as follows:

- *Chapter 2:* Discusses the review of related work on e-commerce customer profiling by focusing on architectures used to implement e-commerce systems and how customer profiles are implemented and utilised using various well-known architectures. The chapter will also investigate how online businesses maintain and update customer profiles to reflect changing customers' needs and preferences over the course of time;
- *Chapter 3:* Discusses SOA by looking at its definition, trends in SOA, SOA benefits and challenges, and SOA enabling technologies; for example, web services and ESB. Notable SOA applications, such as, software as a service (SaaS) and cloud computing will also be discussed. In addition, the chapter will discuss the organisations that are working towards SOA standardisation, such as, W3C, WS-I and OASIS. The chapter will further discuss SOA governance;
- *Chapter 4:* Discusses the process of service-oriented analysis and design (SOAD) of the proposed customer profile system using an SOA. The chapter discusses the three important phases in SOAD using IBM's service-oriented modeling architecture (SOMA) namely: service identification, service specification and service realisation. Customer profile data model and the proposed customer profile UML deployment and component model to aid implementation will also be discussed. Furthermore, SOA design principles and security design will also be discussed in this chapter;
- *Chapter 5:* Discusses the implementation of the customer profile model using SOA. Service implementation, testing and deployment will also be discussed. The chapter will discuss the evaluation of open source SOA implementation tools that were utilised to identify a suitable framework to be used for implementation. Furthermore, the chapter will discuss the selected supporting implementation tools to be used in this research. For example, Netbeans IDE as the back-end customer profile web services implementation tool, Microsoft Visual Studio as the e-commerce web application implementation tool and Microsoft SQL Server as the database management system (DBMS);
- *Chapter 6:* Discusses research evaluation and the analysis of the results. The chapter will discuss how the evaluation instruments, such as questionnaires, eye tracking and statistical instruments were used in this research. In addition, the chapter will discuss

the sampling and evaluation procedures, and data collection and analysis. Results of the evaluation will also be discussed; and

- *Chapter 7:* Concludes the dissertation with research achievements, contributions and recommendations for possible future research projects.



# CHAPTER 2: CUSTOMER PROFILING IN E-COMMERCE

## 2.1 INTRODUCTION

E-commerce has enabled online businesses to extend their activities to a global market, thereby increasing the customer base, sales and consequently profits. Recent research studies indicate that e-commerce is competitive; and opportunities for e-commerce success definitely exist; however, failures also exist (Kim, *et al.*, 2006). Businesses are able to accomplish critical business processes, for example, sales, using e-commerce systems that are managed in-house commonly on client-server architecture (Schneider, 2007). The advent of the latest web technologies, such as Web 2.0 has made interactive web development and e-commerce easier to adopt.

This has increased competition on the web as new online businesses can now be established much more easily.

Online businesses face a number of challenges, one major challenge being determining customers' preferences. Several techniques have been developed as solutions, for example one-to-one marketing strategies. These techniques assist e-commerce businesses to survive on the Internet by increasing customer loyalty and turning e-commerce website visitors into customers (Kim, *et al.*, 2006; Adomavicius and Tuzhilin, 2005).

E-commerce systems are implemented by using various distributed system architectures. The challenge with distributed architecture is in the management of the complexity and heterogeneity inherent in distributed systems (Bakken, 2003). Several attempts have been made by software vendors to develop specific software as a remedy for this challenge; however, few have successfully achieved their goal (Henning, 2006). Customer profiles have been implemented as part of many e-commerce systems, for example, [www.amazon.com](http://www.amazon.com), to identify online customers' needs and preferences (Amazon.com, 2009).

Recently there has been an increase in one-to-one online marketing strategies, for example, personalisation (Kim, *et al.*, 2006). The goal is to have online websites more responsive to the individual needs of each customer or specific group of customers (Adomavicius and Tuzhilin, 2005). Personalisation and other online marketing strategies are conducted based on

information known about a particular customer. Such information is contained in a customer profile.

This chapter gives a literature overview of e-commerce customer profiling. The chapter investigates how online businesses implement e-commerce systems and how customer profiles can be incorporated and utilised. A definition of customer profiles and what parameters constitute a customer profile will be discussed next. The chapter discusses how customer profiles can be maintained by online businesses to incorporate changes in customer's needs and preferences. This chapter's discussion represents step 1 of the research methodology shown in Figure 1.1.

## **2.2. E-COMMERCE SYSTEMS' ARCHITECTURE AND IMPLEMENTATION**

E-commerce systems are normally implemented as distributed systems on the basis of a client-server architecture (Schneider, 2007). The challenge in distributed system architecture is to manage the complexity and heterogeneity inherent in them. Large e-commerce businesses implement middleware software; which is a class of software technologies designed to address the complexity and heterogeneity challenges in distributed systems (Bakken, 2003).

The most commonly used and distributed middleware architectures are Object Management Group's (OMG's) CORBA, Microsoft's DCOM and Sun's Java Remote Method Invocation (RMI) (Schneider, 2007; Bakken, 2003; Raj, 1998). The following sections summarise each of these architectures by discussing how they operate and their heterogeneity across different languages and operating systems.

### **2.2.1 CORBA**

CORBA 2.0 was introduced by OMG in 1997 and it provided a standardised protocol and C++ language mapping, with Java mapping following in 1998 (Henning, 2006). Everything in CORBA depends on the Object Request Broker (ORB). This acts as a central bus over which CORBA objects interact transparently with other objects (Raj, 1998). The ORB in CORBA is responsible for finding an object's implementation, for preparing the object to receive a request, to communicate requests and to carry the reply back to the clients (Raj, 1998). CORBA offers heterogeneity across languages and operating systems.

### 2.2.2 DCOM

DCOM is a distributed object technology from Microsoft that evolved from its Object Link Embedding (OLE) and Component Object Model (COM) (Bakken, 2003). DCOM supports remote objects communications by running on a protocol called Object Remote Procedure Call (ORPC) and is mostly used on a Windows platform (Henning, 2006; Bakken, 2003; Davis and Zhang, 2002; Raj, 1998). A DCOM client calls into exposed methods of a DCOM server by acquiring a pointer to one of the server's object interfaces and then starts calling the server object's exposed methods through the initially acquired interfaces pointer, as if the server object is based in the client's address space (Raj, 1998).

### 2.2.3 JAVA RMI

Java RMI is a distributed object abstraction similar to both DCOM and CORBA; and it provides heterogeneity across operating systems, but not across languages (Bakken, 2003). Java RMI relies on Java Remote Method Protocol and Java Object Serialisation; this, in turn, allows objects to be transmitted as streams (Raj, 1998). A Java RMI server object defines an interface that exposes a set of methods or services offered by the server (Raj, 1998). RMI registries hold information about server objects and the clients then perform a look-up to enquire about the services they require.

### 2.2.4 Problems identified

A number of problems associated with the use of these middleware architectures in e-commerce systems were discovered. For example, there are operating system and programming language interoperability problems in DCOM and RMI respectively (Henning, 2006). These problems have made CORBA the more preferred and successful technology. However, some problems were also found with CORBA, for example (Henning, 2006):

- The implementation of CORBA is costly;
- Difficulty in finding excellent CORBA programmers has resulted in longer development times and high defect rates;
- Unencrypted traffic in CORBA is subject to security threats, especially over the Internet; and
- Versioning is not backward-compatible.

The above problems have prompted the recent focus and increasing adoption of service-oriented architecture (SOA) by businesses. SOA and its related technologies aim at solving some of the problems faced by the use of the previous architectures. SOA will be discussed in Chapter 3.

## 2.3 CUSTOMER PROFILING

Online businesses implement and utilise customer profiles to assist with determining customers' needs and preferences. Online businesses respond accordingly by offering products and services that meet specific customers' needs and preferences. This section discusses the concept of customer profiling by looking at its definition and how customer profiles may be implemented and utilised by online businesses.

### 2.3.1 Definition

A customer profile is a snapshot of who your customers are, how to reach them and why they buy from you. In short, a customer profile is a collection of information that describes the customer (Adomavicius and Tuzhilin, 1999). Customer profiling is the process of developing a profile using relevant and available information to describe the characteristics of an individual customer and to identify discriminators from other customers and drivers for their purchasing decisions (Manifold Data mining inc., 2008).

In other instances online businesses create customer profiles for a group of customers to identify the group's collective characteristics.

Customer profiles can be classified into two sets of information: *factual or static* and *behavioural or dynamic* (Eirinaki and Vazirgiannis, 2003; Adomavicius and Tuzhilin, 1999). Factual profile information contains specific facts about the customer, including demographics, for example, age, gender and name. Behavioural profile information models the behaviour of the customer and this is done using conjunctive rules, such as association and classification rules (Adomavicius and Tuzhilin, 1999).

An example of behaviour is when shopping at weekends, customer X usually spends R100 on groceries (Adomavicius and Tuzhilin, 1999).

By establishing customer profiles, businesses can filter and analyse information to help understand and endeavour to meet the needs and preferences of each specific customer or a group of customers based on their profiles (Liu, *et al.*, 2001). Some notable online businesses

that utilise customer profiles include kalahari.net in South Africa and Amazon.com in the United States of America (kalahari.net, 2009; Amazon.com, 2009).

### 2.3.2 Customer profile implementation

Explicit feedback and implicit feedback are two common methods used by online businesses to establish a customer profile (Jokela, Turpeinen, Kurki, Savia and Sulonen, 2001). Explicit feedback is the simplest method to establish a customer profile in which customers are openly asked to register their details on the website by using an online questionnaire. Information that is captured during this process normally includes *factual* information, such as name, gender, age and other demographic details.

In some instances the registration process proceeds by asking the customer to provide preferences, answer a specific questionnaire, or rate items or products on the website (kalahari.net, 2009; Ntawanga, Calitz and Barnard, 2008a). These sections include *behavioural* information that is used to model the customers' online behaviour. Examples of online businesses that use explicit feedback to establish customer profiles include Amazon.com, kalahari.net and Ebay.com (Amazon.com, 2009; kalahari.net, 2009; Ebay.com, 2008).

Figure 2.1 shows a screenshot of customer profile creation from kalahari.net (2009). The figure shows an explicit feedback page for capturing customers' demographic or factual information. Kalahari.net's website has further pages where a customer explicitly rates products during or after registration and the ratings are then used to model a customer's behaviour to assist kalahari with providing relevant and accurate recommendations (kalahari.net, 2009).

kalahari.net

Welcome to kalahari.net  
Sign in or Register as a new customer

Search for  All Products

Home Books Music DVD Games Electronics Cameras Jewellery Flowers Gifts Toys Wine

### My details

\* Required fields

Create a company profile. [Click here.](#)

Title  \*

First name  \*

Last name  \*

Email address  \*

Communication

Daytime phone no. and code  \*

Cellphone

☒ Tick for SMS Communication. For SA deliveries only.

Date of birth  /  /    
 DD MM YYYY

ID Number  [Click here](#) for more info

Required for SA addresses

Password  \* Minimum of 5 characters.

Confirm Password  \* Minimum of 5 characters.

### My delivery address

Address line  \*   
 Maximum 30 characters

Address line  \*   
 Maximum 30 characters

City/Suburb  \*   
 Maximum 30 characters

Postcode/ZIP  \* Unsure of your postal code? [Click here.](#)   
 Maximum 15 characters

Country

Figure 2.1: Explicit feedback customer profiling method (kalahari.net, 2009)

Figure 2.2 is another example of how customer product information rating can be obtained from the customer by the use of a questionnaire (Ntawanga, *et al.*, 2008a). Research shows that questionnaires are the most commonly used method to establish an initial customer profile (Chang, *et al.*, 2006).

**Create your profile:**

Please select what applies to you in the questionnaire below.

Wines						
1	I purchase wine on regular basis.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
2	I know most of South African wines.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
3	I purchased the Platter wine guide.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
4	I attend wine tasting.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
5	I understand the wine analysis (Alc., RS., pH, TA) on a bottle of wine.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>

Groceries						
1	I purchase groceries on regular basis.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
2	I know most groceries in the local grocery store.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
3	I purchase groceries in order to prepare food.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
4	I know how groceries are grouped in aisles.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>
5	I know the layout of local grocery store.	1 Not at all <input type="radio"/>	2 To some extent <input type="radio"/>	3 To a moderate extent <input type="radio"/>	4 To a large extent <input type="radio"/>	5 Always <input type="radio"/>

Figure 2.2: Product knowledge questionnaire

Implicit feedback is a method that creates customer profiles with very little, if any, customer effort. Different techniques have been developed to assist with the collection of customer details without the customers' effort, for example cookie analysis and web-log mining (Eirinaki and Vazirgiannis, 2003; Millet, Friedman and Felton, 2001). Cookies are web-log files that are created at the server side, but are kept on the client machine (Nelte and Saul, 2000). Cookies keep a record of information, such as passwords, IP addresses, usernames, and other information that customers or users enter on a site with active cookies (Eirinaki and Vazirgiannis, 2003).

This information is used to prevent the need for entering the same type of information every time the user accesses the website. Examples of systems that utilise cookies to create a customer profile are client-side Microsoft .NET passport and Google's auto-fill application.

Web-log mining involves the extraction of valuable information from web-log files that are normally created and stored on the web server. By applying statistical and data mining methods to the web-log data, interesting patterns about the online users' behaviour can be identified, for example, possible correlations between web pages and user groups (Eirinaki

and Vazirgiannis, 2003). The common type of information contained in the web-log file includes (Woon, Ng, Li and Lu, 2003):

- Remote or client machine IP address;
- User's log in name;
- Time and date of request;
- Requested pages and results of requests; and
- Size of data transferred.

In both the explicit and implicit feedback methods, flaws have been identified. For example, in explicit feedback, users assign arbitrary ratings or responses that do not reflect their true opinions (Lee, Park and Park, 2008). In implicit feedback, large volumes of data are collected. These large volumes of data need to go through a data cleaning process and research has shown that data mining is a costly activity.

Table 2.1 displays the most commonly used parameters by online businesses to define a customer profile. The table indicates a summary of an analysis done on three well-known e-commerce websites in South Africa and internationally, namely: kalahari.net, Amazon.com and ebay.com. Explicit feedback is the most common method used by these online businesses to establish their customers' profiles (Figure 2.1).

	Field name	kalahari.net	Amazon.com	ebay.com
Required Fields	Name	✓	✓	✓
	e-mail address	✓	✓	✓
	Phone number	✓	✓	✓
	Date of Birth	✓	✓	✓
	Country	✓	✗	✓
	Password	✓	✓	✓
Optional Fields	Cellphone	✓	✗	✓
	Initial preferences	✓	✓	✓
	Mobile alerts	✓	✗	✗

Table 2.1: Common parameters used to define factual information of a customer

### 2.3.3 Examples of customer profile data storage and maintenance

Researchers in e-commerce have developed a number of ways to model online customers' behaviour by using customer profiling. Customer behaviour information changes with time and therefore there is a need to capture and incorporate these changes.



In a study conducted by Ntawanga, Calitz and Barnard (2008b) a customer profiling system was developed that focused on modeling customer product knowledge and tracking customer's online and purchasing behaviour to update the customer profile (Ntawanga, *et al.*, 2008b). Online customer's product knowledge was initially determined by a questionnaire that asked specific questions on groups of products included in the study namely: Groceries, Wine and Electrical goods (Figure 2.2) (Ntawanga, *et al.*, 2008a).

A product knowledge rating was determined for each online customer in each product category and based on customers' online purchasing behaviour, the rating was adjusted by a decision model implemented in the system. Results indicated that the system was able to establish customers' product knowledge and establish an initial customer profile with which customers were satisfied (Ntawanga, *et al.*, 2008a).

The decision model that was designed, implemented and incorporated in the customer profile was found to be effective in analysing online customers' activities, such as time spent on a product page, number of products clicked and the number of times a customer requested more product information. Subsequently, if necessary, a particular customer's profile was updated (Ntawanga, *et al.*, 2008b). The customer profile in this research was used to provide the appropriate level of product information to a customer during purchasing or browsing of products on the online website created (Ntawanga, *et al.*, 2008a; 2008b).

Customer profile data are normally stored, either in a separate table within the system's database, or in a specific customer profile database. Different parameter values can be used to model customer behaviour, for example customer product knowledge levels (Ntawanga, *et al.*, 2008a). Various coding methods can be used to define a customer's profile, for example, 1 meaning expert, 2 meaning intermediate and 3 meaning novice (Ntawanga, *et al.*, 2008a).

## 2.4 POTENTIAL USES OF CUSTOMER PROFILES IN E-COMMERCE

Customer profiles provide a number of benefits; and online businesses create and maintain customer profiles for various uses. The increase in the amount of information available on the Internet for online users has resulted in an increase in the use of personalisation, recommendations and customisation techniques on the web (Hofgesang, 2007; Chung, Sundaram and Srinivasan, 2007; Kim and Lee, 2005).

The objective for employing these techniques on websites is to reduce information overload on the part of the user and to display only items that are relevant to the user (Chung, *et al.*, 2007; Chang, *et al.*, 2006).

Successful and effective implementation of these online marketing strategies is done based on the information known about a particular customer. This information is contained in a customer profile. Recent studies have shown that many online stores have come to recognise the importance of building personalised profiles of individual web customers (Adomavicius and Tuzhilin, 2005). The need for online businesses to understand their customers is very important, because unlike other traditional forms of business conduct, there is a virtual connection between customers and businesses in e-commerce.

This section discusses some of the main reasons why online businesses utilise customer profiling, namely: personalisation, customisation and recommendation.

#### 2.4.1 E-commerce personalisation

Personalisation refers to the use of technology and available customer information to tailor e-commerce interactions between a business and each individual customer (Adomavicius and Tuzhilin, 2005; Kim and Lee, 2005). Personalisation helps to make a website more responsive to the unique and individual needs of each user (Adomavicius and Tuzhilin, 2005). Successful personalisation applications depend on knowledge about customers' needs and preferences that is contained in a customer profile.

In a study conducted by Ntawanga, *et al.* (2008a; 2008b), a customer profile was used to provide online customers with personalised levels of product information based on each individual customer's profile. In another study conducted by Shearin and Lieberman (2001), a customer profile was used to provide personalised renting apartment information based on a particular customer's profile.

Figure 2.3 shows an example of personalised apartment information provided to a customer based on the customer's profile (Shearin and Lieberman, 2001).

Decide which apartment you prefer, then press the button underneath the apartment.

Apt. Feature	Apartment A	Apartment B
Bedrooms	:Two	:Two
Price	1100	1050
City	:Somerville	:Somerville
Neighborhood	:Beacon	:Inman
Parking?	Yes	No
Mass transit	(Red_Line)	(Bus)
Walk time	10	
Pets allowed	:Cats	:Any
Handicap?	No	Yes
EIK?	Yes	No
DW?	Yes	Yes
Closets	Yes	No
Bathrooms	1	1
Extra rooms	None	None
Laundry fac.	:In_Unit	:In_Building
AC?	Yes	No
Modern?	Yes	Yes
Quiet?	No	No
Sunny?	Yes	Yes
Outside space	None	(Yard)
Luxury features	None	None
Comments		

Buttons: Prefer A, Prefer B

Figure 2.3: Personalised apartment information based on customer profile (Shearin and Lieberman, 2001)

There are three main phases that are followed in personalisation, namely (Eirinaki and Vazirgiannis, 2003):

- *The collection of web data:* This process involves the collection of relevant customer details that can be used for personalisation. The main sources from which customer data is collected are cookies and explicit questionnaires;
- *Web data pre-processing and analysis:* This process prepares the customer data collected from sources such as questionnaires and cookies for web usage mining. Web usage mining is used to reveal hidden knowledge from the data collected. Pre-processing sub-phases include data cleaning and data filtering; and
- *Determining which actions to be taken based on the analysis results:* This phase involves the decision-making process to provide personalisation to customers depending on any relevant information on preferences and needs observed from the web usage mining.

The next section discusses e-commerce customisation as another potential use of customer profiles.

## 2.4.2 E-commerce customisation

Customisation is another widely used method to remove a burden of information overload on the part of the online user (Kim and Lee, 2005). Customisation refers to a process whereby end-users are given an opportunity to select their requirements from a set of comprehensive options and the online businesses provide products and services based on users' explicitly selected requirements (Zhang and Jiao, 2007; Weightman and McDonagh, 2003).

In customisation end-users have a direct input in the product or service specification and creation process (Weightman and McDonagh, 2003).

Customisation is commonly used to outline the appearance and content of an online website (Choi and Han, 2008). For example, the BBC website (2008) asks visitors to select what type of information should be provided, the preferred website interface colour and the region they come from.

Figure 2.4 shows customisation of the bbc.co.uk home page by a particular user (BBC, 2008). Online users select what type of news, for example, sports news, and colour they need on their customised homepage, as displayed in Figure 2.4. After the selections have been made the user clicks the save button and this information is kept in a customer profile and every time the customer accesses the page, the selected options are displayed.

Customisation is further used in application software, such as Microsoft Office in which end-users are asked which Microsoft Office packages they would like to install. However, in this process no customer profile is created for the user as opposed to online interaction.

Display Options | Accessibility Help | Mobiles | Make this my Homepage | Take the tour

the BBC the Web Search

Thursday 23 October 2008

+ Customise your homepage ? Set your location Reset homepage

Add and remove your preferred topics

<input type="checkbox"/> Blogs	<input type="checkbox"/> History	<input checked="" type="checkbox"/> Sport	<input type="checkbox"/> Weird & Wonderful <b>NEW</b>
<input type="checkbox"/> Business & Money	<input type="checkbox"/> Music	<input checked="" type="checkbox"/> TV	<input checked="" type="checkbox"/> World Service
<input checked="" type="checkbox"/> Children	<input checked="" type="checkbox"/> News	<input type="checkbox"/> Teens	
<input type="checkbox"/> Entertainment	<input checked="" type="checkbox"/> Radio	<input checked="" type="checkbox"/> US Elections 2008 <b>NEW</b>	
<input type="checkbox"/> Food	<input type="checkbox"/> Science & Nature	<input checked="" type="checkbox"/> Weather	

Choose a page colour

☒ Rotate Colours

Save Cancel

Figure 2.4: Customisation of a bbc.co.uk homepage (BBC, 2008)

### 2.4.3 E-commerce recommendation

Recommendation refers to the use of available customer information to propose what users might be interested in on the website. A recommender system is a system that attempts to assist users to make decisions in various domains by offering suggestions about items by employing various methods (Chung, *et al.*, 2007). A major difference between personalisation and customisation, discussed earlier, and recommender systems is that in recommender systems suggestions are made to customers in the form of recommendations and the customers have a choice of whether to view or ignore such recommendations.

The following methods are commonly used to derive recommendations in online websites:

- *Collaborative filtering*: Collaborative filtering determines the contents for a specific customer profile based on the customer's profile and reference to other customers with similar profiles. The system does the recommendations solely based on analysing similarities in other customers (Kim and Lee, 2005), and tries to offer recommendations on items the customer has not yet rated.
- *Content-based filtering*: The content-based filtering technique is done based on the customer's profile created from what the customer has done in the past (Kim and Lee, 2005). Some of the customer actions that can be used to create the profile are purchasing behaviour and items a customer has rated in the past.
- *Knowledge-based or Hybrid filtering*: The knowledge-based filtering technique is a combination of content-based filtering and collaborative filtering techniques (Adomavicius and Tuzhilin, 2005). One way to implement the knowledge-based filtering technique is to implement the content-based filtering and collaborative filtering techniques separately and then to combine the results.

Figure 2.5 shows a recommendation page made to a customer from Amazon.com (2009). The figure clearly shows that the contents of the page are recommendations, as indicated at the top of the page.

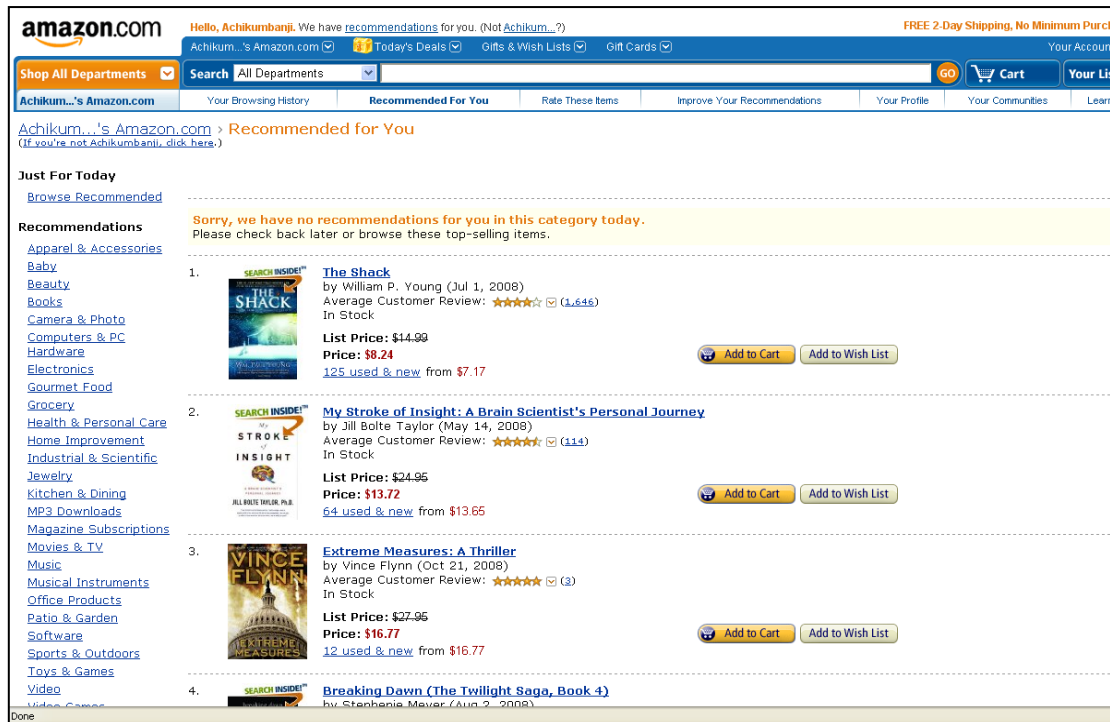


Figure 2.5: Recommendation page for a particular customer (Amazon.com, 2009)

Customer profiling has recently gained popularity in the e-commerce domain because of the benefits it is capable of bringing to online businesses (Kim, *et al.*, 2006; Kim and Lee, 2005). E-commerce system implementation using SOA principles is on the increase and to survive the competition online businesses have to find more effective methods of incorporating customer profiles into these systems. Currently, research does not emphasise customer profiling in an SOA environment.

## 2.5 SUMMARY

This chapter has discussed the concept of e-commerce customer profiling. The chapter has looked at how e-commerce systems are implemented and how customer profiles are incorporated and utilised by online businesses. A customer profile has been defined as a snapshot of who your customers are. Parameters that make up a customer profile have been discussed. The chapter has also discussed some typical uses of customer profiles, such as personalisation, customisation and recommendation. Chapter 3 will discuss SOA.

# CHAPTER 3: SERVICE-ORIENTED ARCHITECTURE

## 3.1 INTRODUCTION

The business and technology focus is now shifting from static business processes towards more dynamic and agile service-oriented processes. The concept of SOA has recently gained significant traction and is proving to have a major impact on many branches of business technology (Sommerville, 2007; Jammes, Mensch and Smit, 2005; Barry, 2003). SOA helps businesses to match the dynamic nature of today's business environment; and additionally, it promotes organisations' agility.

SOAs are a way of developing distributed systems where components of the system are loosely coupled, stand-alone services (Sommerville, 2007). This means that SOA is made up of services that interact and communicate with each other to achieve a common goal.

This chapter covers SOA by discussing what it is and how businesses implement systems using SOA. Attention will also be given to enabling technologies, such as web services and ESB, existing SOA models and examples. Different standards and technologies that are useful in SOA are introduced, for example, SOAP, WSDL and UDDI. Business process management (BPM), which illustrates how business goals and objectives are aligned with IT, will also be also explored.

SOA success depends largely on standards; hence this chapter introduces some notable organisations that are working towards standardisation in SOA, such as W3C, OASIS and WS-I (Erl, 2005). In addition, the chapter also discusses SOA governance. The chapter further represents step 1 of the research methodology discussed in Figure 1.1.

## 3.2 DEFINITION OF SOA

A number of SOA definitions exist; however, for the purposes of this research SOA has been defined as the set of architectural patterns and guidelines needed for creating loosely coupled, standard-based, business-aligned services that provide a new level of flexibility in response to business threats and opportunities (Canfora, *et al.*, 2008; Maurizio, *et al.*, 2008; IBM, 2007).

In short, SOA can be explained as a set of architectural tenets for building autonomous, yet interoperable systems (Jammes, *et al.*, 2005). The fundamental idea of SOA is that systems

are designed and implemented using loosely coupled components called services (Canfora, *et al.*, 2008; Cugola and Di Nitto, 2008; Bell, 2006). Services form the basis for SOA.

### 3.2.1 Service-orientation and services

Recently, service-orientation has become a multi-discipline concept that covers science and technology to bridge the gap between Business Services and IT Services. Service-orientation focuses on automating business processes using IT services (Buyya, Yeo, Venegopal, Broberg and Brandic, 2009). Businesses are now becoming more service-oriented and IT plays a crucial role in their service delivery (Qui, 2007).

Today's business environment is dynamic; and in order to remain competitive, enterprises have to be agile and be able to respond to customers' needs (Oracle, 2008).

Service-orientation enables businesses to respond quickly to the ever-changing business environment and thereby meet customers' dynamic needs. Through this process, service orientation provides a basis for the link between business processes and IT. A number of technologies promoting service-orientation and service-computing exist, for example, web services and SOA (Buyya, *et al.*, 2009).

A service is the implementation of a specific well-defined business function (Josuttis, 2007; Mahmoud, 2005). The complexity of the function can vary from being simple, such as retrieving a customer's address, to complex such as processing a customer's order (Josuttis, 2007). Fully implemented services in SOA must possess the following characteristics (Sommerville, 2007; Barry, 2003):

- A well-defined function;
- Self-contained; and
- Should not depend on the context or state of other services in the architecture.

These properties enable services in SOA to work independently of each other, while at the same time being able to communicate and interact to achieve a common goal.

### 3.2.2 Business Process Management (BPM)

BPM is a management discipline that deals with the analysis, design and continuous improvement of business processes (Josuttis, 2007; Oracle, 2008). BPM provides IT and business experts with a common language to achieve their separate goals that aim, in turn, to



achieve the organisation's goal (Oracle, 2008). Services are components of business processes and form the basis for SOA (Josuttis, 2007). BPM helps to analyse and decompose business processes into their lowest level of activities, services that can be implemented using SOA.

### 3.2.3 Notable service-oriented applications

This section explores the concept of software as a service and, in addition, cloud computing; it sees these as two emerging fundamentals in the current business environment, and as the technology needed in response to the challenges of today's business environment using SOA.

#### 3.2.3.1 Software as a Service (SaaS)

Software as a service is when service computing is used to develop a piece of software that resides in the cloud, and can be accessed on demand over the Internet (Vaquero, Roderom-Merino, Caceres and Lindner, 2009; Miete, 2009). Salesforce and Zoho are examples of organisations offering software products as services (Salesforce.com, 2009; Zoho.com, 2009). Customers for such software include developers, architects, analysts, business partners and end-user customers (Zoho.com, 2009; Microsoft.com, 2009).

Salesforce specialises in Customer Relationship Management (CRM) software that can be accessed on the Internet; and service consumers only pay for what they use (Salesforce.com, 2009; Hayes, 2008). Figure 3.1 is an example of a free demonstration version of the Salesforce system. The figure shows the system components on the tabs at the top, as well as other services that can be run when a consumer has logged on to the Salesforce service website (Salesforce.com, 2009).

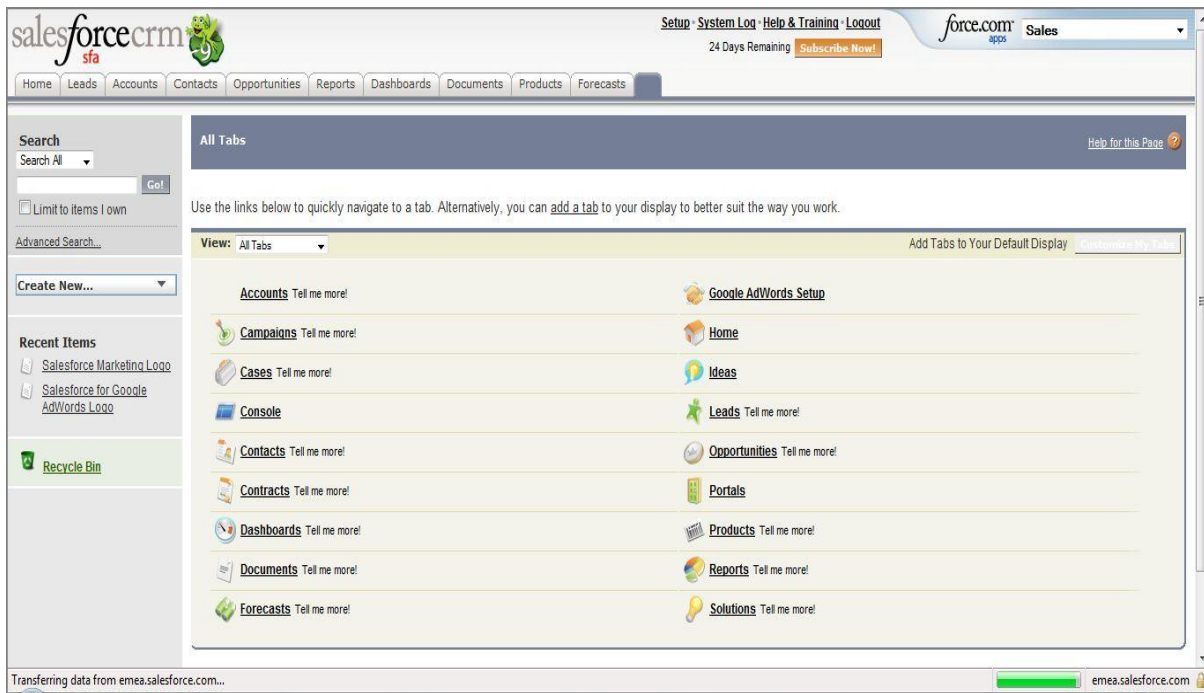


Figure 3.1: Salesforce CRM service page (Salesforce.com, 2009)

Zoho services provide functions such as reporting tools, invoicing, word-processing, project-management software, as services that can all be accessed online at a fee. Customers can log in using a username and password; create a document and keep it on a Zoho server. Figure 3.2 shows an interface on an invoicing service from a free trial version of Zoho (Zoho.com, 2009).

The screenshot displays the Zoho Invoice web application. The top navigation bar includes links for Switch To, Help, Feedback, Forums, Subscription, Settings, and Sign Out (fntawanga). A search bar for invoices is also present. The main navigation menu has tabs for Home, Estimates, Invoices, Customers, Items, and Reports. On the left sidebar, there are sections for 'New Invoice', 'Invoices' (with sub-links like All Invoices, Open Invoices, Drafts, etc.), 'Payments Received', and 'Recurring Profiles'. The central area is titled 'Add New Invoice' and contains a form with the following fields:

- Select Customer:** Emile Senga
- Template:** Classic
- Currency:** MWK
- Exchange Rate:** 1 MWK = 1 MWK
- Invoice Date:** 05-11-2009
- Terms:** Net 15
- Due Date:** 05-26-2009
- Invoice#:** Will be auto generated
- P.O#:** (empty)
- Late Fee:** No Late Fee

Below the form is a table for invoice items:

Item	Description	Qty	Rate	Discount%	Tax1	Tax2	Amount
Maise	Cooked Supper	1.00	1200.00	0.00	-NA-	-NA-	1200.00

Figure 3.2: Zoho online invoicing service (Zoho.com, 2009)

### 3.2.3.2 Cloud computing

Cloud computing is when highly scalable computer hardware resources, such as processing power, storage and network bandwidth are provided and accessed as a service from the Internet at a fee; and customers only pay for what they have used (Vaquero, *et al.*, 2009; Brantner, Florescu, Graf, Kossmann and Kraska, 2008; Hayes, 2008). Amazon Web Services (AWS) is currently the most prominent utility-computing provider (Brantner, *et al.*, 2008; Hayes, 2008). Another example is Microsoft Azure platform service (Microsoft.com, 2009).

Amazon Elastic Computing Cloud (EC2) is one example of AWS application. EC2 is a utility-computing application that provides customers with scalable computer hardware, such as CPU and disks that can be accessed as a service on demand from the cloud (Amazon Web Services, 2009). Amazon simple storage service (S3) is another example from the AWS list. S3 provides scalable and highly available storage service at a low cost (Brantner, *et al.*, 2008).

Other AWS include Amazon SimpleDB, Amazon cloudfront and simple queue service (SQS) (Amazon Web Services, 2009).

Microsoft Azure Platform services provide a run-time environment in the cloud that delivers on-demand computing, storage, and automated systems management (Microsoft.com, 2009). Developers can host, scale, and manage web applications on the Internet through Microsoft data centres using Azure. Microsoft Azure services platforms include Azure for developers, corporate, system integrators, independent software vendors (ISV) and business. Figure 3.3 is an overview of Azure services platform (Microsoft.com, 2009).

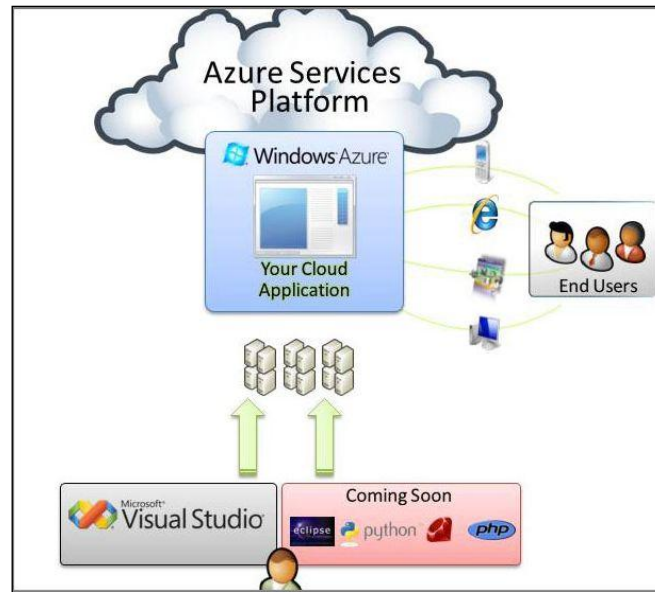


Figure 3.3: Microsoft Azure Platform service (Microsoft.com, 2009)

Benefits of service and utility-computing include (Buyya, *et al.*, 2009; Murray, 2009):

- Can be accessed and utilised anywhere, anytime;
- Cost effective, as customers pay for what has been used and there is no need for upfront investment; and
- Customers do not need to take other non-functional requirements, such as security, scalability, availability, updates, efficiency and implementation complexities into consideration, as these are taken care of by the service providers.

Service and utility-computing is a promising challenge to software providers to create software that can be invoked as services and serve millions of customers, as opposed to having a piece of software running on individual machines (Miete, 2009). The underlying factor for such a vision is the wide acceptance of the Internet and latest technologies, such as web services and SOA.

### 3.3 THE IMPLEMENTATION OF SYSTEMS USING SOA

A number of frameworks used to implement SOA-based systems have been developed. The frameworks include design and run-time capabilities, as well as all the software functionality an enterprise requires to build and operate an SOA (Fiorano Software Inc., 2008).

A highly distributable communication and integration backbone is required to implement an SOA and this is provided by an ESB (Papazoglou and van den Heuvel, 2007). Several software vendors have developed enterprise and open source ESB which can be used to implement SOA. An ESB does not implement an SOA; however, it physically provides an implementation backbone and a framework by which services can communicate in SOA (Papazoglou and van den Heuvel, 2007; IBM, 2007). Section 3.5.2 further discusses the ESB in more detail.

IT vendors have come up with various SOA architectures (Microsoft.com, 2009; IBM, 2007). Figure 3.4 shows the IBM SOA reference architecture that demonstrates how SOA can be implemented. The architecture shows the vital role an ESB plays in communicating various services and disparate applications within an enterprise. At the core of the architecture are services that communicate by using various standards and protocols, for example, interaction services, process services and business services.

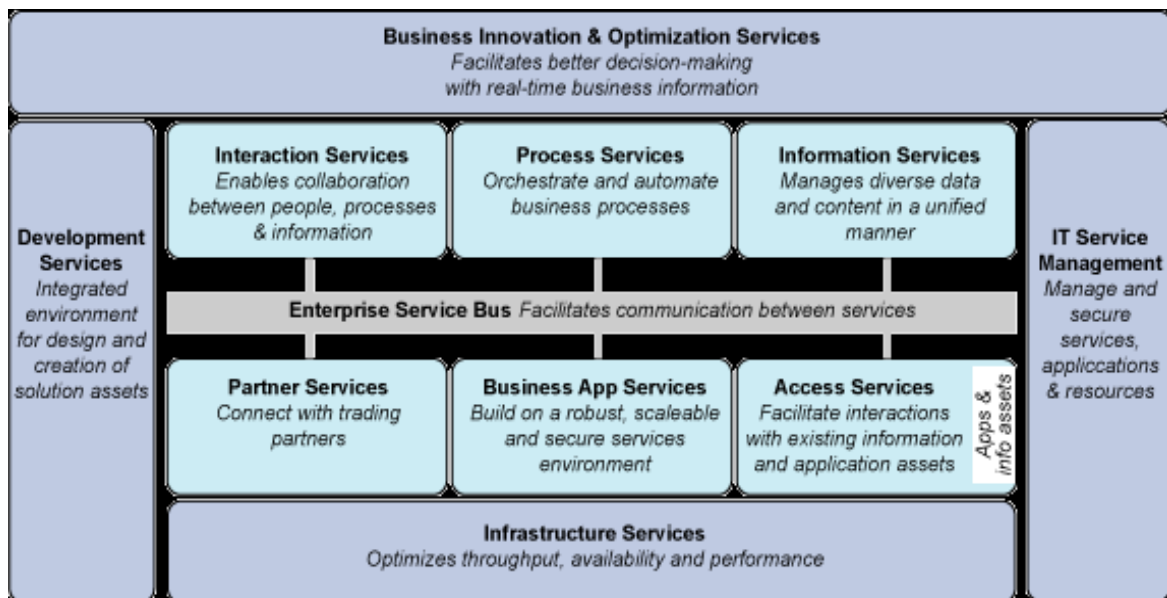


Figure 3.4: SOA reference architecture (IBM, 2007)

The next section discusses how services interact in an SOA.

### 3.4. SERVICES INTERACTION IN SOA

Services in SOA use the *find-bind-execute* paradigm, as shown in Figure 3.5 to interact and exchange messages between each other (Mahmoud, 2005). In this paradigm, service providers register their functions in a service registry and service requestors discover the available services by accessing the service registry.

Figure 3.5 shows the service-interaction model, as discussed by Erl (2005). The figure shows how a service provider/producer registers or deploys itself into a service registry by using web service description language (WSDL) and how a service consumer can find out the available services using universal description, discovery and integration (UDDI).

The figure also shows how services communicate and exchange messages using standards such as SOAP and hypertext transmission protocol (HTTP). Section 3.5 discusses the outlined protocols, standards and other supporting technologies used for service interaction in SOA.

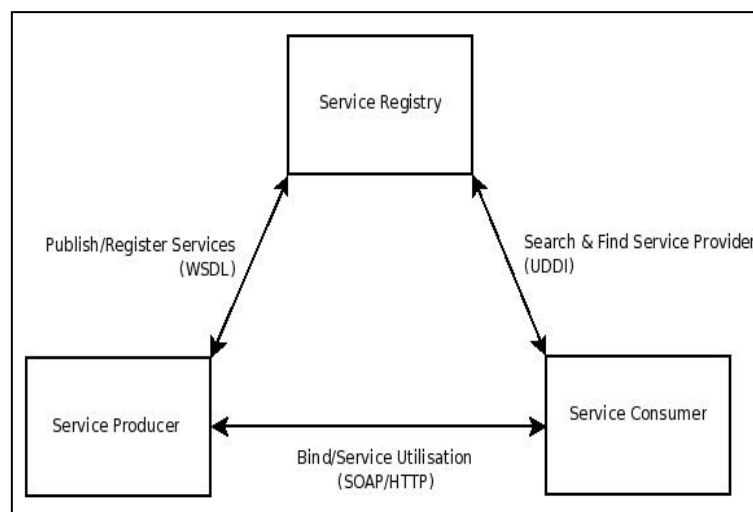


Figure 3.5: SOA's Find-Bind-Execute paradigm (Mahmoud, 2005; Erl, 2005)

### 3.5 SOA ENABLING TECHNOLOGIES

SOA can be implemented using several techniques, for example, Grid computing. However, web services constitute the most commonly used implementation technique for SOA in the e-commerce domain (Pastore, 2008; Erl, 2005). This section discusses the concept of web services by analysing what they are and how they are used in SOA environments. Other supporting standards and protocols used along with web services will also be discussed.

### 3.5.1 Web services

A web service is defined as a software system designed to support interoperable machine-to-machine interaction over a network (Votis, Alexakos, Vassiliadis and Likiothanassis, 2008; Schneider, 2007; W3C, 2004). Web services are self-contained, modular applications that can be described, published, located and invoked over a network or the Internet. Two widely accepted techniques to implement web services exist: Representational State Transfer (REST) and SOAP web services techniques.

Web services enable services in SOA to communicate by exchanging messages expressed in formats such as extensible mark-up language (XML) and Java Script Object Notation (JSON) (Sommerville, 2007; Schneider, 2007; Barry, 2003).

SOA being a distributed architecture, services can be located and executed at geographically distributed computers. Web service is a common form of service used to implement SOA; and hence any further reference to service will refer to web service.

Web services implementation is governed by the web service architecture (WSA) that outlines the main component of the web service implementation which can be applied to SOA. The architecture was proposed by W3C in 2004 (W3C, 2004). Figure 3.6 shows the structure of the web service architecture (WSA), as discussed by W3C (2004). The WSA has the following four main models;

- *Policy model*: Focuses on constraints on the behaviour of agents and services;
- *Message-oriented model*: Focuses on the structure of messages, the relationship between message senders and receivers and how messages are transmitted;
- *Resource-oriented model*: Focuses on the key features of resources that are relevant to the concept of resource, independent of the role the resource has in the context of web services; and
- *Service-oriented model*: Clarifies the relationship between agents and services it requests and provides.

WSA forms a basis for SOA implementation using web services (W3C, 2004).

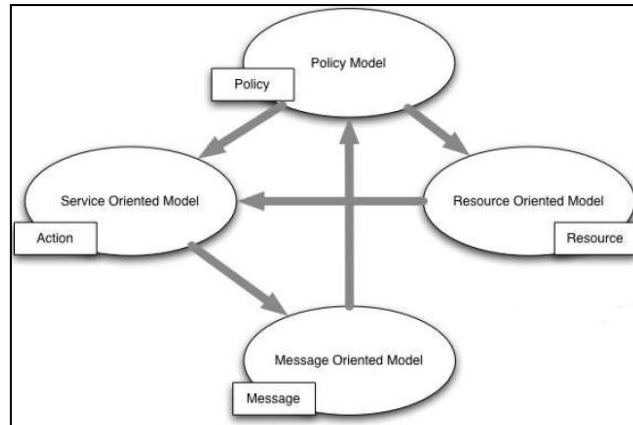


Figure 3.6: Meta model of web service architecture (W3C, 2004)

### 3.5.2 Enterprise Service Bus (ESB)

An Enterprise Service Bus (ESB) is a software infrastructure that enables the integration and re-use of business components. It is a highly distributable communication and integration backbone generally required to implement an SOA (Papazoglou and van den Heuvel, 2007; Josuttis, 2007). Several software vendors have developed commercial and open source ESB which can be used to implement SOA, for example, Sun Java and Oracle.

An ESB does not implement an SOA; however, it physically provides an implementation backbone for SOA (Papazoglou and van den Heuvel, 2007; IBM, 2007).

The ESB is an open, standard-based message bus designed to enable implementation, deployment and management of SOA-based solutions focusing on assembling, deploying and managing distributed SOA across various platforms and languages (Papazoglou and van den Heuvel, 2007; Josuttis, 2007). ESB acts as both a transport and transformation facilitator to allow for the distribution of services over disparate systems and computing environments (Papazoglou and van den Heuvel, 2007).

An ESB also deals with security, reliability service management and monitor logging (Josuttis, 2007). Figure 3.4 shows the IBM's SOA reference architecture which in turn shows various players in SOA and how the ESB provides a communication backbone to the whole architecture (IBM, 2007).

Using an ESB is beneficial as compared with traditional point-to-point connections. ESB is capable of relieving the customer of knowing the actual address of the provider (Josuttis, 2007). Communication costs when using point-to-point can be calculated as follows: for  $n$



points, each node will need  $(n-1)$  connections which brings the total number of connections to  $[n(n-1)/2]$  (Josuttis, 2007). ESB requires only one connection for each node and hence for  $n$  nodes the connection cost is reduced by a factor of  $(n-1)$  (Josuttis, 2007).

### 3.5.3 Standard technology and protocols used in SOA

A number of organisations working towards ensuring interoperability, application-to-application communication, flexibility and scalability of services in SOA exist. Examples of organisations that are working towards the standardisation of protocols and other technology used in SOA include the W3C, OASIS, WS-I and other companies, such as IBM and Microsoft.

This section discusses the standards developed and being enforced in SOA by W3C and OASIS, namely: SOAP, WSDL and UDDI (W3C, 2007; OASIS, 2004). XML is also discussed in this section as an underlying technology in SOA.

#### 3.5.3.1 SOAP

SOAP, initially used as an acronym for Simple Object Access Protocol, is a protocol with its main purpose being to define a standard message format for messages transferred between services in SOA or simply over a network (Schneider, 2007; Erl, 2005). SOAP provides web services an envelope with a standard, extensible, composable framework for packaging and exchanging XML messages (W3C, 2004; Barry, 2003).

The envelope contains two parts (Barry, 2003):

1. An optional header with information on authentication, encoding of data and how the recipient should process the message; and
2. The body that contains the message that can be defined using WSDL.

SOAP messages can be carried from one service to another by a number of protocols, for example HTTP, SMTP, FTP or a proprietary messaging protocol (Papazoglou and van den Heuvel, 2007; W3C, 2004). The current version of SOAP at the time of writing the dissertation was 1.2 (W3C, 2007).

#### 3.5.3.2 Web Service Description Language (WSDL)

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information (W3C,

2004). WSDL provides a formal definition of the endpoints interface, so that requestors wishing to communicate with the service provider know exactly how to structure request messages. WSDL also establishes the physical location of the service (Erl, 2005).

WSDL describes web services starting with the messages that are exchanged between requestor and provider agents (W3C, 2004; Barry, 2003). Web service definitions can be mapped to any implementation language, platform, object model and messaging system (W3C, 2004). Service providers design and implement services using any programming language. However a standard WSDL is used to provide specifications of the service (Sommerville 2007; Papazoglou and van den Heuvel, 2007; Barry, 2003).

The current WSDL version at the time of writing the dissertation was 2.0 (W3C, 2007). Figure 3.7 is a screenshot of a WSDL that shows the different parts of a WSDL file used for communicating with other services, for example the message part, port types, binding style and input and output.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="FelixWSDL" targetNamespace="http://j2ee.netbeans.org/wsdl/FelixWSDL"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://j2ee.netbeans.org/wsdl/FelixWSDL">
  <types/>
  <message name="FelixWSDLOperationRequest">
    <part name="part1" type="xsd:string"/>
  </message>
  <message name="FelixWSDLOperationResponse">
    <part name="part1" type="xsd:string"/>
  </message>
  <portType name="FelixWSDLPortType">
    <operation name="FelixWSDLOperation">
      <input name="input1" message="tns:FelixWSDLOperationRequest"/>
      <output name="output1" message="tns:FelixWSDLOperationResponse"/>
    </operation>
  </portType>
  <binding name="FelixWSDLBinding" type="tns:FelixWSDLPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="FelixWSDLOperation">
      <soap:operation/>
      <input name="input1">
        <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/FelixWSDL"/>
      </input>
      <output name="output1">
        <soap:body use="literal" namespace="http://j2ee.netbeans.org/wsdl/FelixWSDL"/>
      </output>
    </operation>
  </binding>
  <service name="FelixWSDLService">
    <port name="FelixWSDLPort" binding="tns:FelixWSDLBinding">
      <soap:address location="http://localhost:${HttpDefaultPort}/FelixWSDLService/FelixWSDL">
    </port>
  </service>
</definitions>
```

Figure 3.7: An example of a web service description language

### 3.5.3.3 Universal Description, Discovery and Integration (UDDI)

Universal Description, Discovery and Integration (UDDI) is a platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. The focus of UDDI

is the definition of a set of services supporting the description and discovery of (OASIS, 2004; Uddi.org, 2004):

- Businesses, organisations, and other web services providers;
- The web services they make available; and
- The technical interfaces which may be used to access those services.

The current UDDI Version at the time of writing the dissertation was 3.0.2 and its specification describes the web services, data structures, and behaviours of all instances of a UDDI registry (OASIS, 2004; uddi.org, 2004).

### 3.5.3.4 Extensible Mark-up Language (XML)

Extensible Mark-up Language (XML) is a general purpose specification for creating custom mark-up languages. The basic structure of XML is a document that has content and an indication of what role the content plays. The main use of XML is to help applications share data over the Internet (XML.com, 2008; Barry, 2003). XML can be used to share and define data storage (Barry, 2003).

### 3.5.4 Java Business Integration (JBI)

JBI is a Java-based standard that defines run-time architecture for plug-ins to interoperate via a mediated-message exchange model (Apache.org, 2009). Messages between components are mediated by the Normalised Message Router (NMR). JBI has two major components, the binding components and service engines (Apache.org, 2009).

The binding components enable services to connect to the JBI by providing necessary protocols, such as HTTP, while the service engines host the services by providing business logic and integration services (Mahmoud, 2005). Figure 3.8 gives an overview of JBI components and how they connect to the NMR (Mahmoud, 2005) and Figure 3.9 shows the JBI components (Figure 3.8) when the JBI node is expanded, as in Netbeans IDE 6.5. Figure 3.9 shows the details of what sub-components are contained in each JBI component, such as, the binding component has database, file, ftp, http, jms and ldap binding styles.

A number of SOA implementation frameworks are based on JBI, for example, Apache Service Mix and Open ESB.

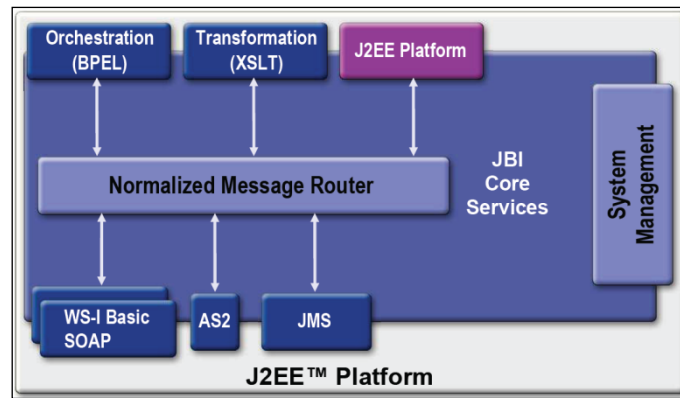


Figure 3.8: JBI components

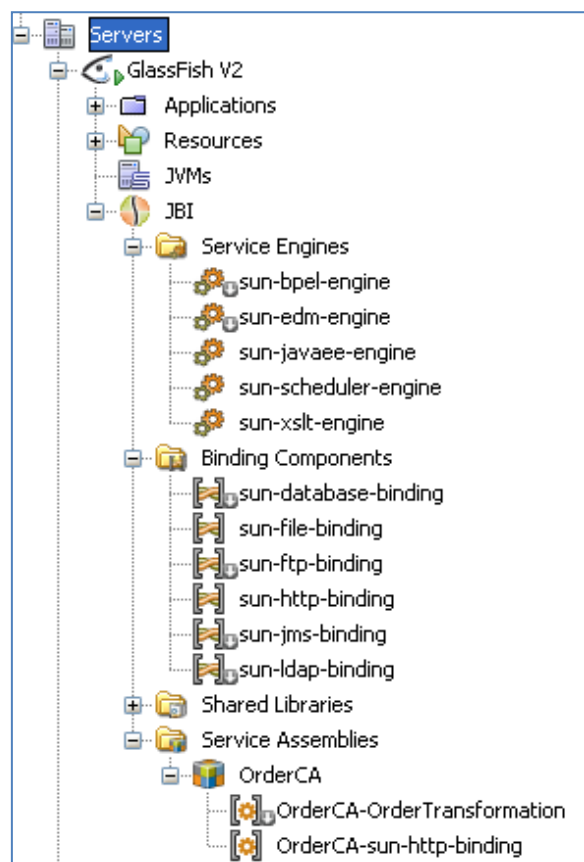


Figure 3.9: Expanded JBI components from Netbeans IDE 6.5

### 3.5.5 Business Process Execution Language (BPEL)

BPEL is a platform independent, XML-based language used to specify and orchestrate business processes based on web services (Mahmoud, 2005; Erl, 2005). Section 3.2.2 discussed BPM as a tool that enables businesses to align their processes with technology towards achieving goals. Business processes are composed of series of interacting services in SOA; and BPEL specifies the order in which such services have to be executed to achieve a

process goal (Mahmoud, 2005; Erl, 2005). BPEL performs its activities in two broad ways, namely basic and structured ways. The activities are discussed as follows:

Basic activities include:

- Invoke;
- Receive;
- Reply;
- Assign;
- Throw;
- Wait; and
- Empty.

Structured activities include:

- Sequence;
- Flow;
- Switch; and
- Scope.

Figure 3.10 shows the BPEL structure with the business process flow section containing various BPEL activities.

```

<process>
  <!-- Definition and roles of process participants -->
  <partnerLinks> ... </partnerLinks>
  <!-- Data/state used within the process -->
  <variables> ... </variables>
  <!-- Properties that enable conversations -->
  <correlationSets> ... </correlationSets>
  <!-- Exception handling -->
  <faultHandlers> ... </faultHandlers>
  <!-- Error recovery - undoing actions -->
  <compensationHandlers> ... </compensationHandlers>
  <!-- Concurrent events with process itself -->
  <eventHandlers> ... </eventHandlers>
  <!-- Business process flow -->
  (activities)*
</process>

```

Figure 3.10: BPEL Structure

The discussed standards and technologies used in SOA are some of the contributing factors to its rise in adoption. These standards and technologies bring along a number of benefits when used to implement SOA. The next section discusses the benefits of SOA.

## 3.6 BENEFITS OF SOA

SOA can be regarded as the natural evolution of previously distributed architectures, such as DCOM and CORBA; as such SOA has been developed to overcome problems that were identified with DCOM and CORBA architectures (Erl, 2005). There are a number of benefits and reasons why businesses are adopting SOA to develop systems; and this section highlights some of the most notable benefits, according to Erl (2005) and other SOA software vendors such as IBM (2007).

### 3.6.1 Improved integration (interoperability)

SOA can be built of solutions that consist of inherently interoperable services and therefore allow different technologies to be dynamically integrated, independently of the system's platform in use (IBM, 2007; Erl, 2005). Web services are developed based on agreed standards and protocols, thus the potential is there to build cost-effective cross-application integrated systems (IBM, 2007; Erl, 2005).

This implies that various services developed using various technologies can interoperate provided the implementation abides by the service standards.

### 3.6.2 Inherent re-use

Loosely coupled components in SOA support component re-use which can lead to reduced development efforts, costs and improved IT project delivery (IBM, 2007; Erl, 2005). Services that perform certain business functions can also be re-used to build much larger composite systems to perform other business functions as well (Erl, 2005).

### 3.6.3 Leveraging of legacy investment

SOA assists in the easy introduction of new business systems and the integration of legacy systems (IBM, 2007; Erl, 2005). The lifetime of mission critical legacy systems can be extended by wrapping them into a service; and SOA enables these systems to be dynamically accessible as web services (Canfora, *et al.*, 2008).

### 3.6.4 Best-of-breed alternatives

SOA establishes a vendor-neutral communication framework and this frees organisations from being chained to a single proprietary development or middleware (Erl, 2005). Web services can exchange messages using open standardised protocols; and the message itself is

standardised. These two characteristics have the significant implication of allowing organisations to choose best-of-breed environments for specific applications, as long as it supports the creation of standard web services (Erl, 2005).

### 3.6.5 Organisational agility

Organisational agility refers to the organisation's ability to respond quickly to changes in its operating environment (IBM, 2007; Erl, 2005). SOA can establish a loosely coupled relationship between two enterprise domains which allows each domain to evolve independently and adapt to changes without causing disruptions and expenses (Erl, 2005).

Furthermore, loosely coupled components in SOA can be changed individually without affecting other parts of the architecture. This increases the agility with which technology can respond to business-process changes.

## 3.7 SOA CHALLENGES

SOA poses a number of challenges to organisations and the IT community. This section discusses some of the challenges in an SOA environment.

### 3.7.1 Autonomy and interoperability

These are contradictory properties of services in SOA (Jammes, *et al.*, 2005). One of the major challenges of SOA is to reconcile these opposing terms. Interoperability is the fundamental quality for information systems aiming at enabling large-scale data exchange (Birukou, Blanzieri, D'Andrea, Giorgini, Kokash and Modena, 2007). Apart from technical issues, interoperability includes cultural and organisational aspects.

Autonomy, on the other hand, refers to the operations of a service perceived as being opaque by external components (Papazoglou and van den Heuvel, 2007). This ensures that external components neither know nor care about how services perform their functions. Applications consisting of autonomous services can be viewed as composite, self-reliant services that exercise their own self-governance within SOA (Erl, 2005).

One of the challenges in SOA is to achieve both autonomy and the interoperability of services. A number of organisations are working on standardisation in SOA with the aim of bringing in a compromise between services' autonomous state and their interoperability. Such organisations include WS-I, OASIS and W3C and are discussed in Section 3.9.

### 3.7.2 Security

SOA is a concept for dealing with business processes distributed over different heterogeneous systems. The existing security mechanisms and policies for these systems are likely to differ. Security in SOAs is not clear and different perceptions exist (Cotroneo, di Flora, Graziano and Russo, 2008). Thus, the challenge of introducing a general security concept over many different existing security concepts cannot be ignored (Josuttis, 2007). The process starts with having different user IDs for the same people and results in having different abstractions and processes to introduce roles and user-profiles.

The deployed services in SOA have to be secured in the same way as other systems and the following are the main four security functions required by services (Peterson, 2008):

- *Authentication*: This is the service provider's task to validate the authenticity of the service requestor or consumer (Figure 3.5). In SOA services are deployed into a distributed environment; and service authentication is therefore important when deciding who has access to what service;
- *Authorisation*: It is the service provider's task to authorise service consumers for a particular service;
- *Auditing*: Refers to keeping track of what actions the service-consumer performed with the services in SOA; and
- *Assurance*: Assurance in SOA security refers to the consumers' trust in the service-provider and the service itself: that it is from a reliable source and it will perform the intended function.

Service security and SOA governance are increasingly being considered as two critical factors for SOA success. The next section discusses SOA governance as a tool that is used to manage SOAs.

## 3.8 SOA GOVERNANCE

SOA governance deals with the non-technical aspects of an SOA (Josuttis, 2007). There are a number of SOA non-technical aspects that contribute to the success of an SOA implementation. The goals of SOA governance include (Josuttis, 2007):

- Clarify business case for SOA adoption;
- Adopted or preferred reference architectures;



- Roles and responsibilities in an SOA environment;
- Implementing standards, policies and formats in an SOA; and
- Defining process and lifecycles.

Standards play a very important role in SOA and the standardised implementation of services is the main contributing factor to continued rise in SOA adoption and its implementation (Erl, 2005). Standards improve the order in which things can be done. The next section discusses examples of organisations working towards standardisation in SOA.

### 3.9 ORGANISATIONS WORKING TOWARDS STANDARDISATION IN SOA

Standardisation helps to drive SOA by solving the interoperability challenge in having services developed using different languages and platforms working seamlessly together. Previous architecture platforms were realised within vendor-specific boundaries, environments in which standards that mattered, were proprietary (Erl, 2005).

There are a number of notable organisations that are working towards the standardisation of SOA. This section discusses some of these organisations in SOA, namely: World Wide Web Consortium (W3C), Organisation for the Advancement of Structured Information Standards (OASIS) and Web Service Interoperability Organisation (WS-I).

#### 3.9.1 W3C

Originally founded by Tim Berners-Lee in 1994, W3C has been largely responsible for enhancing the World Wide Web as a global, semantic medium for information sharing (W3C, 2009; Erl, 2005). Its standard releases include HTML, XML-based standards such as XML schema and XSLT, WSDL and SOAP (Erl, 2005).

W3C also produced and regularly updates the Web Service Architecture document which remains a reference point and platform-neutral web service architecture document (Erl, 2005). W3C's main focus is on establishing core, industry-agnostic standards and its work has been noticeable over the past years (Erl, 2005).

#### 3.9.2 OASIS

OASIS was originally founded in 1993 as SGML open and later changed to OASIS to reflect a shift from SGML to XML-related standards (Erl, 2005). OASIS standard releases include

UDDI, WB-BPEL and ebXML, a specification that aims to establish a standardised means of business-to-business (B2B) data interchange (Erl, 2005).

OASIS has also been influential in furthering the development of XML and web services security extensions, such as Security Assertion Markup Language (SAML) and Extensible Access Control Markup Language (XACML) (Erl, 2005). However, the Web Service Security (WSS) technical committee carries out most of the important security-related projects (Erl, 2005). OASIS group's primary focus is on leveraging standards developed by W3C to produce additional specifications that support various vertical industries (OASIS, 2004; Erl, 2005).

### 3.9.3 WS-I

The primary objective of WS-I is to ensure that all standards developed by other organisations promotes interoperability which is one of the SOA goals (WS-I, 2008; Erl, 2005). WS-I was originally founded in 2002 and has membership of major SOA vendors, for example, Microsoft and IBM (WS-I, 2008).

WS-I is best known for the Basic Profile and Basic Security Profile documents. Basic Profile is a document that states the available standards that can collectively form the most desirable interoperable architecture for services in an SOA (Erl, 2005). Basic Security Profile is a recent release from WS-I and its main focus is on the web services and XML security technologies.

## 3.10 SUMMARY

This chapter has discussed the concept of SOA by looking at its definition, enabling technologies, such as for example, web services, JBI, BPEL and WSDL. The chapter also explained how SOA systems can be implemented by studying how services can be created and how they interact in SOA. Benefits and challenges facing SOA implementation have also been outlined in this chapter. Various organisations working toward standardisation in SOA have also been discussed, for example OASIS, WS-I and W3C.

Chapter 4 discusses the analysis and design of the customer profile model, using a service-oriented modeling architecture (SOMA).

# CHAPTER 4: THE ANALYSIS AND DESIGN OF A CUSTOMER PROFILING SYSTEM USING SOA

## 4.1 INTRODUCTION

The conventional software development process involves the use of a number of phases in order to arrive at a desired solution. These phases include problem analysis, requirements elicitation, systems analysis and design, development and maintenance (Sommerville, 2007). In an SOA environment, a similar approach is adopted and it is called Service-Oriented Analysis and Design (SOAD). The phases in SOAD consider issues including service identification, service specification, service development, service deployment and SOA governance (Arsanjani, 2004).

Chapter 3 introduced SOA and discussed how SOA-based systems are implemented, how services interact in SOA, and the benefits and challenges in SOA (Sections 3.3, 3.4, 3.6 and 3.7). The chapter further explored SOA enabling technologies such as web services, ESB and standards that enable the successful implementation of web services and SOA, for example, SOAP, WSDL and UDDI (Section 3.5).

This chapter discusses the process of SOAD, and how this process has been applied and used in this research. A number of SOAD modeling approaches exist. For the purposes of this research, IBM's Service-Oriented Modeling and Architecture (SOMA) approach was used and will be discussed. The chapter discusses how the customer profiling process has been organised as a business process, how various services have been derived and their purpose in the customer profiling process. The chapter further discusses the customer profile Unified Modeling Language (UML) component and the deployment model proposed. This chapter's discussion represents step 2 of the research methodology discussed in Figure 1.1.

## 4.2 SERVICE-ORIENTED ANALYSIS AND DESIGN

SOA is based on services that originate from various business entities, including business process breakdown. Seemingly, the most difficult question SOA architects have to answer is:

*What services should be built to achieve the business goals?* (Erl, 2005; Arsanjani, 2004).

The traditional object-oriented analysis and design process has proved not to be efficient in an SOA environment (Zimmermann, Krogdahl and Gee, 2004). As a remedy for this, the SOAD technique was developed for use in the SOA system development life cycle. SOAD is an interdisciplinary process that assists SOA architects to identify, design and implement relevant services that build an SOA (Arsanjani, 2004; Zimmermann, *et al.*, 2004).

SOAD plays a very important role in the SOA delivery life cycle because it involves a series of phases during which SOA characteristics are incorporated in the service-oriented systems development (Erl, 2005). The SOA delivery life cycle deals with the whole process from establishing the need for an SOA to SOA governance (Section 3.8), while SOAD emphasises phases within the cycle that ensure the end product is indeed SOA, that is, to ensure that the product conforms to SOA characteristics (Erl, 2008).

Erl (2005) outlines six phases in the SOA delivery life cycle, as follows:

- Service-oriented analysis;
- Service-oriented design;
- Service development;
- Service testing;
- Service deployment; and
- Service administration.

IBM's SOMA is a commonly used modeling tool in the SOAD process (Zimmermann, *et al.*, 2004). SOMA was developed by IBM as one way of assisting the analysis, the design and the implementation of services in an SOA environment (Arsanjani, 2004). SOMA is an SOAD modeling tool for designing and realising an SOA that aligns business analyses, processes and goals. In other words, SOMA outlines the key activities required for the analysis and design of SOA-based systems with the focus on achieving business goals through properly structured business processes (Arsanjani, 2004).

This section, therefore, introduces SOAD by examining the following SOMA phases (Arsanjani, 2004):

- Service identification;
- Service specification; and
- Service realisation.

### 4.2.1 Service identification

The service identification process involves the identification of services that form a business process and that can be implemented as self-contained business functions (Josuttis 2007; IBM, 2007; Bell 2006). Service identification is one of the first steps in SOAD (Inaganti and Behara, 2007). The process of identifying appropriate services for an organisation is vital for the successful implementation of SOA. However, this process has proven to be difficult and time consuming (Inaganti and Behara, 2007; Bell, 2006; Levi and Arsanjani, 2002).

Services are usually discovered from various business entities including business processes. Researchers have devised two commonly used approaches to identify services (Josuttis, 2007; IBM, 2007; Inaganti and Behara, 2007; Erl, 2005):

- *Top-Down approach:* In this approach business entities are decomposed into smaller chunks until a basic service level is reached; and
- *Bottom-up approach:* This is the opposite of the top-down approach. Services are identified first and business entities and processes are built from smaller and specific chunks of services.

Research indicates that these two approaches work better when they are used simultaneously rather than employing each method separately (IBM, 2007; Bell, 2006). This top-down and bottom-up mixed approach is known by different names, for example, middle-out, middle-ground, meet-in-the-middle and agile (Josuttis, 2007; Erl, 2005). The top-down and bottom-up mixed approach solves the problems that may be faced when either approach is used separately. For example, when services are identified using the top-down approach only, they may be difficult to implement because the top-down approach does not take into account how the implementation of the service is going to be dealt with, while the bottom-up approach mostly starts by examining implementation (Josuttis, 2007).

Individual services can be identified in a number of specific ways. The following is an example of guidelines that can be used to identify services, as discussed by Hubbers, Ligthart and Terlouw (2007) and Bell (2006):

- Business process decomposition;
- Business function decomposition;
- Business entity decomposition;
- Ownership and responsibility;

- Goal driven;
- Component based;
- Existing supply;
- Non-functional requirements;
- Front-office usage analysis; and
- Infrastructure.

The nature of the business and SOA entry points might prescribe different ways to identify services. SOA entry points are levels on which an organisation would like to start SOA implementation. IBM (2007) outlines five SOA entry-point perspectives, namely:

- People;
- Process;
- Information;
- Re-use; and
- Connectivity.

Business process management, as discussed in Section 3.2.2, plays a significant role in the service identification process and this is where business goals are aligned with IT in SOA (Buyya, *et al.*, 2009). The primary questions that the service identification process aims to answer are (Erl, 2005):

- What services need to be built? and
- What logic or scope of functionality should be encapsulated by each service (granularity)?

Once appropriate services have been identified, it is imperative to classify the services into groups according to the type, functionality and the role they play in an SOA (Josuttis, 2007; Erl, 2005). The benefits of services classification include (Josuttis, 2007; Hynes and Pradhan, 2006; Arsanjani, 2004):

- Helps to identify functional domains and composition that improve re-usability and prevents re-invention;
- Easy to define standards for service classes, for example, setting up policies for different classes;

- Easy to monitor and manage requirements for different service classes that align with business goals; and
- Improve SOA governance (Section 3.8).

Different authors classify services differently; however, the meanings are similar (Erl, 2008; Josuttis, 2007; Erl, 2005). Josuttis (2007) outlined three fundamental classes of services, namely:

- *Basic services*: These services play a role of wrapping the back-end or problem domain so that clients for the basic service can access the back-end by using the common SOA infrastructure. These services provide business functionalities that cannot be broken down further. Basic data and basic logic services are two subclasses of basic services. Examples of such services include creating a new customer (data service) and a service to determine whether a year is a leap year or not (logic service);
- *Composite services*: These are services that are composed of other services to perform a certain business function using the process called orchestration. These services are also called orchestrated services since service orchestration is required to create a composite service. Composite services can access either a single or multiple back-ends. An example of such a service is an update customer details service; and
- *Process services*: These are services that normally have a stable state over multiple calls. An example of such a service is a shopping cart.

Erl (2005) used different terminologies for service classification. The following are classifications according to Erl (2005):

- Application services;
- Business services; and
- Process services.

Another classification was discussed by Kulkarni and Dwivedi (2008) in which they classified services into eight categories as follows:

- Process services;
- Business services;

- Composed services;
- Informational services;
- Data services;
- Utility services;
- Infrastructure services; and
- Partner services.

In this research, Josuttis' (2007) classification has been adopted because it was well explained and was found to be suitable in serving the purposes of this study which was to implement a customer profile in an SOA environment (Sections 1.4; 1.5).

#### 4.2.2 Service specification

Service specification is the description of everything a service consumer would need to know in order to interact with the service provider (Terlouw, 2009; Cummins, 2009; Amsden, 2007). The result of the service specification process is a non-ambiguous description of each and every service in an SOA. Service specification plays a very important role in communicating what a service is all about to business analysts, domain experts, software developers and top management (Terlouw and Maarse, 2009). The service specification also defines how a service can be accessed to provide its service to a consumer.

Figure 4.1 shows a service specification framework developed by Terlouw and Maarse (2009). The framework comprises three basic parts indicated by three separate blocks on the left-hand side of Figure 4.1. The parts cover the following (Terlouw and Maarse, 2009):

- *Who offers the service:* This part of the service specification offers information about the service provider's name and contact details;
- *What the service does:* This part of the service specification discusses what the service does. The information includes: the functions of the service, the requirements to interact with the service, for example, required input and expected output, error messages and quality of service; and
- *How the services can be accessed and utilised:* This part discusses the location, protocols and standards that can be used to interact with the service. For example, location can be the service's server address; communication protocol can be HTTP and SOAP that can be used as message transportation standard.



The left-hand side of Figure 4.1 discusses the general aspects of service specification, while the right hand side focuses on the specific descriptions to be included in a service specification. Section 4.4 discusses how this framework has been used in specifying the services implemented in this research project.

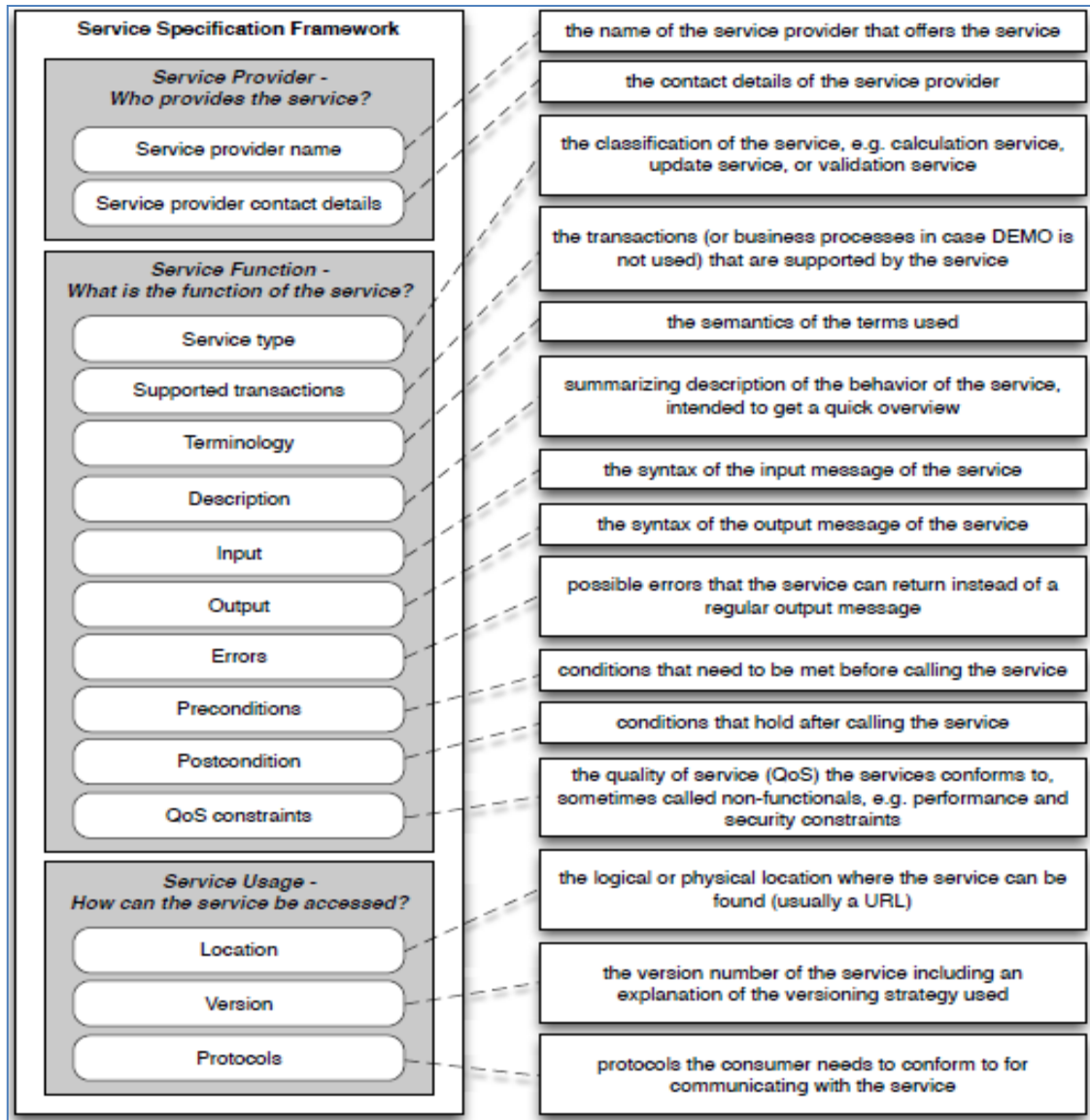


Figure 4.1: Service specification framework (Terlouw and Maarse, 2009)

Specific guidelines for service specification have been developed to help SOA architects define non-ambiguous specifications (Arsanjani, 2004). The following are some of the service specification guidelines, according to Cummins (2009) and Amsden (2007):

- *Service name*: A verb phrase generally used to indicate what the service does;

- *Required and provided interfaces:* Describes the functions of the service, for example, required inputs and expected outputs;
- *Protocols and standards used:* Contains the description of the communication protocols and standards, for example, standards discussed in Section 3.5.3;
- *Constraints:* The intended successful use of the service;
- *Qualities such as, scalability, performance, cost and availability:* These are generally the services' non-functional requirements; and
- *Policies such as security and integrity:* These are the non-functional requirements treated specially in SOA.

The service specification guidelines were used in this research to supplement the service specification description following Terlouw and Maarse's (2009) framework. Terlouw and Maarse's (2009) framework provided the general outline of a service specification, while the guidelines (Cummins, 2009; Amsden, 2007) provided specific details for presenting various aspects of the framework, for example, how a service name should read to reflect the service's purpose.

### 4.2.3 Service realisation

Service realisation or implementation is the process of acquiring a service; and it is conducted once service specification has been completed (Figure 4.2) (Arsanjani, 2004; Papazoglou, 2003). SOA architects can consider a number of service realisation decisions to implement services for an SOA. The following are four commonly used service implementation decisions (Bierberstein, Laird, Jones and Mitra, 2008; Papazoglou, 2003):

- *In-house service design and implementation:* This approach involves the design and implementation of services using in-house facilities, for example, using in-house equipment and personnel. The approach is also called build;
- *Purchasing/leasing/paying:* A service is purchased from a developer or vendor for a fixed fee or pay-per-use method. Subscribe and buy are other names that are used for this service implementation decision. SaaS and cloud computing, as discussed in Section 3.2.3, are typical examples of this approach;
- *Outsourcing:* The design and implementation of services is outsourced. The difference between outsourcing and the buy approach is that in the buy approach, the consumer does not contribute to the specification and development of the

service, while in outsourcing the developer is given the specifications by the consumer; and

- *Adapter and wrapper:* This approach re-uses existing legacy systems and it is sometimes called legacy modernisation. In wrapper, a legacy system is wrapped as a web service and the functions are exposed without any modifications to the code. In adapter the legacy system code is modified or integrated with a new code.

The service realisation decision made in this research was in-house service design and implementation or build. This decision was made in line with the main objective of the research, as discussed in Section 1.5 which is how can one implement a customer profile using an SOA. According to the researcher, no existing customer profile services or similar customer profile model could be found. The build service realisation technique uses the technologies discussed in Chapter 3, such as, WSDL, SOAP and UDDI to implement services that can interact in SOA.

Services in SOA are commonly implemented as web services, additionally, other means can be used to realise an SOA, for example, grid computing (Pastore, 2008; Lopez-Sanz, Acuna, Cuesta and Marcos, 2008). In this research SOAP web services were used to implement the services, as opposed to REST web services (Section 3.5.1). SOAP web services offered more benefits and complemented the objectives of this research. For example, SOAP web services support various standards, such as WSDL, UDDI and SOAP (Section 3.5.3). Chapter 5 discusses how the customer profile services were implemented in this project using the selected SOA implementation tools that included: Netbeans IDE 6.5, GlassFish application server V2, and Microsoft SQL Server 2008 Express.

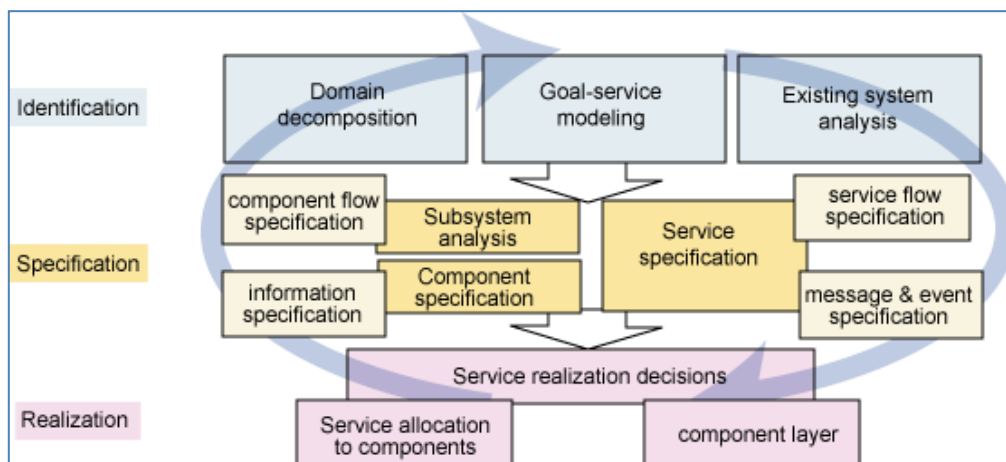


Figure 4.2: Summary of the service-oriented modeling architecture (Arsanjani, 2004)

Figure 4.2 shows a summary of the three discussed phases in SOAD using SOMA: service identification, service specification and service realisation or implementation (Section 4.2). SOAD is an iterative or incremental process in which new services are continuously being discovered and incorporated in an SOA model, as indicated by the circular arrow in Figure 4.2 (Arsanjani, 2004). The downward arrows connecting the phases in Figure 4.2 indicate the order in which the different phases should be completed. For example, service identification has to be followed by service specification and finally by service realisation.

The rectangular boxes within a phase show some of the activities that can be carried out within a phase. For example, domain decomposition can be carried out in service identification to identify services; component specification can be carried out as part of service specification and service allocation to components can be carried out in service realisation, in which functionality is embedded in a service (Arsanjani, 2004).

The next section discusses how SOMA was used in the analysis and design of a customer profiling system using an SOA model.

### 4.3 CUSTOMER PROFILE SERVICE-ORIENTED ANALYSIS AND DESIGN

SOMA was the SOA modeling technique used to conduct the analysis and design of the customer profile services for the project. This section discusses the two SOMA phases carried out in this research project: customer profile service identification and specification. Chapter 5 discusses the third SOMA phase for this project. This is customer profile service realisation. As discussed in Section 4.2.3, the service realisation decision made for this research is build.

#### 4.3.1 Customer profile services identification

Top-down analysis was conducted to identify the services, in order to perform the customer profiling in this research. The top-down approach has been explained in Section 4.2.1. Bell (2006) argues that a top-down approach is the most appropriate approach to start with in the service identification process. Using the top-down approach, service re-use and conceptual agreement of service can be achieved; however, the identified services can be difficult to implement (Bell, 2006). Josuttis (2007) emphasises that a top-down approach helps SOA architects to understand what is really needed and it ensures a clear separation of activities.

The initial step in customer profile service identification was formulating the goal. The main goal of a customer profile in business is to understand customers in order to serve their needs and preferences better (Section 2.3). Customer profiling is a marketing tool; and therefore it can be linked to the marketing process of almost any business (Wagner and Zubey, 2007).

The following sections discuss the services identified, the functionality they provide in the SOA customer profile system and how the services collaborate with each other in order to achieve the main goal. The customer profile service identification process followed the SOMA service identification process, as discussed in Section 4.2.1. Section 4.3.2 provides the service specification which stipulates important aspects for successful interaction with the identified customer profile services, for example, valid input and output parameter types.

#### 4.3.1.1 Register customer service

*Register customer service* was a service to perform the registration of a new customer. Usually, when a customer visits an online website for the first time, the customer is requested to register (Amazon.com, 2009; kalahari.net, 2009). The registration process is normally optional, as customers can browse and purchase without having a user account or a profile on a website.

In this research, customer registration was made compulsory, in order to achieve its goal of creating and maintaining individual customer's profiles. One of the goals of the project is to investigate the creation of customer profiles using SOA; and it was therefore vital to have each user registered in the system. During registration, customers were required to complete an online biographical details questionnaire. An analysis of this information requested by different sites was presented in Table 2.1.

Participants were required to complete an online questionnaire that was to assess their initial product knowledge for the products used in this research (Section 2.3.3). A survey was conducted with a sample of 30 participants to validate the questionnaire used to assess customers' initial product knowledge for the creation of individual customer profiles (Ntawanga, *et al.*, 2008a).

The *register customer service* can be accessed by any application running in a mobile or PC environment, and implemented by using any programming language. When required parameters were collected and validated, the *add customer service* was invoked to save the details into a customer profile database.

#### 4.3.1.2 Add customer service

*Add customer service* was one of the four basic: create, read, update and delete (CRUD) type of services that this system had. Its main purpose was to add valid data input from a legitimate service consumer into a database, regardless of the source, provided the data met the specifications required to invoke its service. Information to this service was sent as standard text using SOAP (Section 3.5.3.1), and to avoid possible denial of service, data specification was to be followed.

*Register customer service* can access this service to save customer details into a database upon meeting the specified requirements for interaction, as shown in Table 4.1 (Section 4.3.2).

#### 4.3.1.3 Get profile service

*Get profile service* was responsible for retrieving the profile of a particular customer. The basic functionality was reading the customer profile database and retrieving a customer's profile. The retrieved customer profile was used to provide personalised product information to the customer, while interacting with the e-commerce web application that was developed to test the SOA-based customer profile system.

This service offered its functionality to any service that sent a request for a customer's profile, provided the requestor met the conditions for the interaction, as shown in Table 4.1 (Section 4.3.2). Only registered customers would have their profiles available for access and could be utilised for personalisation in this research project.

#### 4.3.1.4 Update profile service

*Update profile service* was responsible for updating an individual customer's profile. The clients for this service were responsible for the provision of legitimate parameters that defined the customer's profile, as shown in Table 4.1 (Section 4.3.2). The *update profile service* implemented in this research was a composite service (Section 4.2.1) using three other implicit services, namely:

- *Monitor customer activities*: This service was monitoring and recording customers' activities during the interaction session on the client's side;
- *Analyse customer activities*: This service was responsible for analysing and classifying the recorded activities of the customer and deriving a new profile; and

- *Compare profile:* This service utilised the *get profile service* to retrieve a customer's old profile and to compare it with the newly derived profile. When the two profiles differed, the derived profile was retained and flagged as a new profile, otherwise the update was abandoned.

#### 4.3.1.5 Identify customer service

*Identify customer service's* main function was to determine whether a customer was registered and had a profile available for access in the database. *Get profile service* was invoked when the customer existed; otherwise *register customer service* was invoked to ask the customer to register in the system. The *identify customer service* can be accessed and utilised by any other service when the required parameters - which are login details, email address and username - are provided.

#### 4.3.1.6 Delete customer service

The *delete customer service* was accessed and invoked when customers opted to remove their details from the system. A number of e-commerce websites that store customer information do not provide an option for customers to remove their details once they have registered (Amazon.com, 2009; kalahari.net, 2009). In this research, this service was to determine:

*To what extent can optional deletion of customer profiles improve customers' trust and honesty when providing details for customer profiling?*

Research indicates that customers do not trust online websites that collect their personal information; and as a consequence they provide information that does not reflect their true opinion (Lee, *et al.*, 2008; Chang, *et al.*, 2006). A proposal from this research is that when users have more control over their profiles, for example, knowledge of how the system rates them, being able to edit and delete their profiles, then the more they will be willing to provide relevant and accurate information for profiling.

The next phase that follows service identification is service specification (Figure 4.2). Service specification provides the necessary details a service consumer needs to know before interacting with the service (Section 4.2.2). The following section discusses the customer profile services specification, using the framework proposed by Terlouw and Maarse (2009) (Section 4.2.2).

### 4.3.2 Customer profile services specification

Service specification is a crucial part in an SOA implementation procedure, as it stipulates and communicates all that is required for successful interaction between the service producer and the service consumer (Figure 3.5) (Amsden, 2007). This means that incorrect or incomplete specifications can result in an ineffective SOA implementation. That is to say, service consumers cannot find services, or invalid input can be sent to service producers. This section discusses the specification for each customer profile service identified and was discussed in Section 4.3.1 using the framework discussed in Section 4.2.2. The following items of the specification framework were used in this research to specify the customer-profile services:

- Name and contact details of the service provider;
- Type and description of the service;
- Input required by the service and output generated from the computation;
- Errors and notifications that can be expected and notifications when an error occurs;  
and
- Locations and protocols used to get to the service.

As discussed in Section 4.3.1, the customer profile services perform different functions, such as, identifying a customer, updating a profile and deleting a profile; and therefore it follows that the specifications of the different customer profile services will also be different. For example, some services cannot display an error message to the service consumer because they are invoked by another service within the customer profile services. Each service belongs to the Department of Computing Sciences at NMMU. Table 4.1 shows service specifications for all the services implemented in this project. Section 4.4 extends the customer profile services specification by discussing the customer profile data requirements.



Register Customer	
Type and description	UI and Input validation service.
	The main function of the service is to validate user input, for example, biographical details.
Input	Customer details with various data types.
Output	Valid parameters sent to add profile service.
Error	Invalid parameter types and missing required input, for example, customer
Notifications	Error message displayed on the interface in red colour.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP
Add Customer	
Type and description	Create record service.
	The service is responsible for adding customer's details and profile into a
Input	Valid input from register customer service.
Output	Return a unique customer ID.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP
Get Profile	
Type and description	Read record service.
	The service's function is to retrieve a particular customer's profile.
Input	Unique customer ID.
Output	A customers' profile.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP
Update Profile	
Type and description	Update record service.
	The service is responsible for updating a customers' profile.
Input	Unique customer ID and new profile ratings.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP
Identify Customer	
Type and description	Find record service.
	The service's function is to authenticate a customer.
Input	Log in details (email and password).
Output	Unique customer ID.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP
Delete Customer	
Type and description	Delete records service.
	The service is responsible for the removal of all customer details from the
Input	Unique customer ID.
Location URL	Masters-01
Protocols	SOAP over http and TCP/IP

Table 4.1: Customer profile services specification

Customer profile information (Section 2.3.1) was stored in a customer profile database. The customer profile information was modelled in such a way that it was easy to capture, store and update in the database. For successful service interaction, the data requirements had to match with the service specification requirements. For example, proper data types had to be passed to services. The next section will present the data model used in this research to manage the customer profile information. The customer profile data model extends the customer profile service specification, as discussed in Section 4.3.2.

#### 4.4 CUSTOMER PROFILE DATA MODEL

The customer profile data model discusses the data types and requirements for the customer profile. Data types and tables required to maintain customer profile information will be discussed in this section and the discussion extends the customer profile service specification, as outlined in Section 4.3.2.

Services in SOA communicate by sending standardised text messages that have to conform to the requirements of the service provider; this is done by using techniques such as SOAP over HTTP (Section 3.5.3). It was therefore necessary to specify all the data types that service consumers need to provide in order to communicate successfully with the customer profile services and database. The following database tables were used to store customer profile information in the system:

- *Basic profile table*: Contains a basic profile for customers that includes their biographical details;
- *Product knowledge rating table*: Stores the customer profile ratings;
- *Activity log table*: Stores the customers' recorded activities during the interaction session;
- *Identify customer table*: Stores the customers' authentication details.

The database tables will be used for interface purposes with basic CRUD services of the system. Table 4.2 is a representation of the table structure and data types used for the customer profile web services.

TABLE	ATTRIBUTE	DATA TYPE
Basic profile	Customer_id	Integer
	Customer_name	String
	Customer_date_of_birth	Date and Time
	Customer_gender	String
	Customer_e-mail	String
	Customer_city	String
	Customer_phone_number	String
	Customer_password	String
Product knowledge rating	Customer id	Integer
	Wine_rating	Integer
	Grocery_rating	Integer
	Electrical_rating	Integer
	Date	Date and Time
Activity log	Customer_id	Integer
	Total_time_on_site	Integer
	Time_spent_on_wine	Integer
	Time_spent_on_groceries	Integer
	Time_spent_on_electrical	Integer
	Number_of_wines_browsed	Integer
	Number_of_groceries_browsed	Integer
	Number_of_electrical_browsed	Integer
	More_wine_info.	Integer
	More_grocery_info.	Integer
	More_electrical_info.	Integer
	Date	Date Time
Identify customer	Customer_id	Integer
	Customer_e-mail	String
	Customer_password	String

Table 4.2: Table structure and data types for the customer profile database

#### 4.5 CUSTOMER PROFILE SERVICES CLIENTS

Services in SOA interact and serve other services to successfully complete various tasks (Erl, 2005). In this research project a number of services were designed as customer profile service clients. The purpose of the client services was to evaluate the implementation of a customer profile in an SOA environment.

An e-commerce website was developed with functionalities to access and utilise the customer profile services. Microsoft Visual Studio 2008 IDE was used to create a user interface for the

e-commerce website. Products to be used in the system were introduced in Section 2.3.3. Chapter 5 discusses how the customer profile service clients were implemented, using various available technologies and implementation tools.

One of the objectives of this research was to complete a comparison between the customer-profile system implemented using SOA and the object-oriented customer profile running on a client-server architecture (Section 1.5). The following section discusses the analysis and design of an object-oriented customer profiling system.

## 4.6 OBJECT-ORIENTED CUSTOMER PROFILING SYSTEM

An object-oriented customer profiling system was designed to assist in completing the research objective of comparing the SOA implemented customer profile system and an object-oriented customer profiling system, running on a client-server architecture (Section 1.4). The designed system was similar to the system implemented in a study by Ntawanga, *et al.* (2008b) as was discussed in Section 2.3.3.

In this study an object-oriented customer profile was designed to be utilised by the website, as discussed in Section 4.5. This inferred that two customer profiling systems with a similar interface were designed, one implemented using object-oriented principles running on client-server architecture, and another application implemented using SOA principles and running in an SOA environment.

The object-oriented customer profile system had the customer profile class with the customer profiling services (Section 4.3.1) designed as its objects. Different functions were incorporated into objects to perform the customer profiling in this system.

Once the analysis and design had been completed, a customer profile component and deployment model diagram was proposed. The objectives of the model were to:

- Visually present the outcome of the analysis and design of the customer profile using an SOA; and
- Guide implementation, as will be discussed in Chapter 5.

The next section presents the background to software engineering model design, and how a customer profile model was developed.

## 4.7 CUSTOMER PROFILE MODEL USING SOA

In a number of fields, models are widely used during project development as a means of communicating the solution among the project stakeholders (Mohagheghi and Agedal, 2007; Rumbaugh, Jacobson and Booch, 2004). A model improves the general understanding of the project by portraying only the important aspects and omitting the rest (Rumbaugh, *et al.*, 2004). In other words, a model simplifies the reality. In software engineering, model-driven development continues to receive increased attention (Eichelberger, 2008; Paterno, 2000). This section discusses the proposed customer profile model, using SOA.

### 4.7.1 Model definition

A number of definitions exist for a model. However, for the purpose of this research the following definition was adopted:

*“A model in software engineering is defined as a representation of a system that focuses on the system’s major aspects and hides the details”* (Mohagheghi and Agedal, 2007).

Models in software engineering are represented visually by one or more diagrams. Nevertheless, text is also used (Stevens and Pooley, 2006). Different models can be produced by using modeling languages during the system’s development life cycle (Stevens and Pooley, 2006).

### 4.7.2 UML

Modeling languages define the collection of model elements. A number of modeling languages exist; however, OMGs UML is a widely accepted modeling language in software engineering (Uml.org 2009; Eichelberger, 2008; Booch, Rumbaugh and Jacobson, 2005; France, Evans, Lano and Rumpe, 1998). According to Rumbaugh, *et al.* (2004), UML is a general purpose visual modeling language used to specify, visualise, construct, and document the artifacts of a software system.

The term “*unified*” in UML signifies the language’s amalgamation of different aspects that are included in UML. For example, UML can be used to derive models for different stages during the project’s life cycle; it can be used to design models across different domains and it is implementation-independent (Rumbaugh, *et al.*, 2004).

There are two groups of UML model diagrams that can be used during the software development life cycle: *structural* and *behaviour* diagrams. Structural diagrams present the visual static aspects of the system (Booch, *et al.*, 2005). Examples of structural UML diagrams include:

- *Class diagram*: These diagrams show the set of classes, interfaces and the relationships between them (Booch, *et al.*, 2005; Paterno, 2000);
- *Component diagram*: These diagrams show the internal parts, connectors and ports that implement the component (Booch, *et al.*, 2005; Paterno, 2000);
- *Composite structure diagram*: These diagrams show the internal structure of a class. The difference between these diagrams and component diagrams is small and they are mostly regarded as component diagrams (Booch, *et al.*, 2005);
- *Object diagram*: Shows the set of objects and their relationships;
- *Artefact diagram*: Shows the relationship between artefacts and the classes they implement; and
- *Deployment diagram*: These show the set of nodes and their relationships in a run-time environment.

UML behaviour diagrams depict the system's dynamic aspects. The system's dynamic aspects include the message inflow and outflow, and the physical movements of objects over a network (Booch, *et al.*, 2005). Examples of these diagrams include (Booch, *et al.*, 2005; Rumbaugh, *et al.*, 2004):

- Use-case diagrams;
- Sequence diagrams;
- Communication diagrams;
- State diagrams; and
- Activity diagrams.

#### 4.7.3 Customer profile UML composite and deployment model

The model that this research proposes is represented by a UML composite and deployment diagram (Figure 4.3). The mixed characteristics of the UML composite and deployment diagram made it a suitable choice to present the model this research proposed. The nodes in a UML deployment and composite diagram represent run-time resources, such as a computer,

device or memory (Rumbaugh, *et al.*, 2004). A node encompasses artefacts such as databases, components and executable web files.

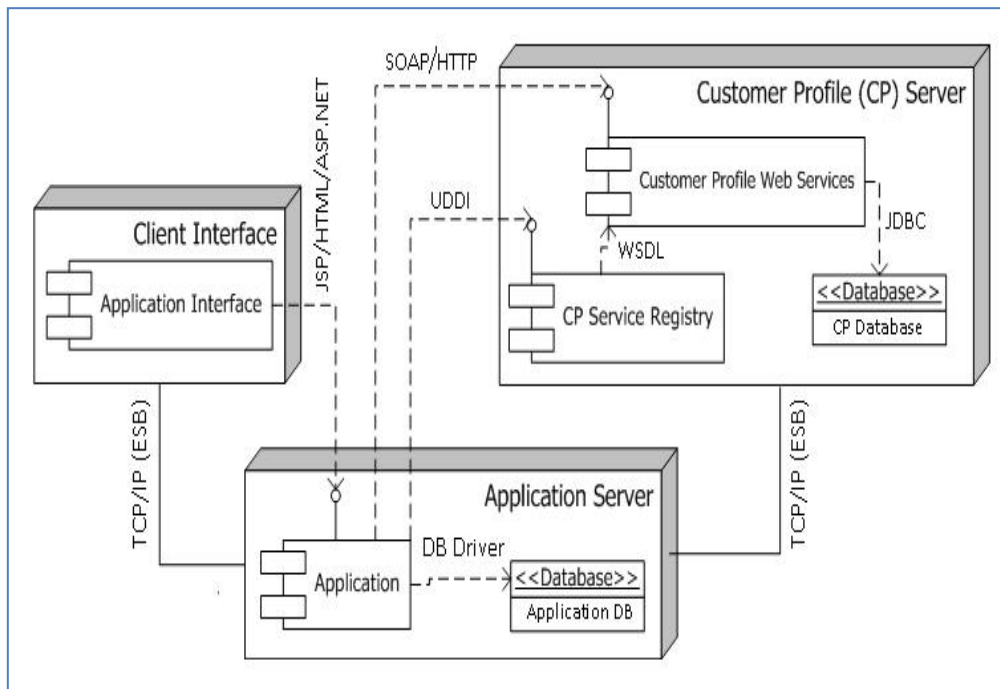


Figure 4.3: Customer profile model using SOA

The customer profile UML composite and deployment diagram of the proposed model shows the relationship between its classes and the physical configuration of software and hardware. Figure 4.3 shows the model's three main nodes, namely:

- *Customer profile server*: This represents the physical environment in which the customer profile web services were deployed and from which it was running;
- *Application server*: This represented the environment in which the e-commerce web application utilising the customer profile web services was deployed and from which it was running; and
- *Client interface*: This represented the environment from which a customer would access the application, for example, a desktop computer.

The model's four components are:

- *Customer profile web services*: This represents the actual customer profile web services. All the customer profile web services, as discussed in Section 4.3.1, were implemented in this component;

- *Customer profile service registry:* This contains the description of the customer profiling services identified in Section 4.3.1. The description is similar to the service specification, as discussed in Section 4.3.2. Section 3.5.3 discussed the UDDI that was used to describe the services in the customer profile service registry;
- *Application:* This component consisted of an e-commerce web application that was developed to utilise the customer profile web services and is discussed in Section 4.6; and
- *Application interface:* This component represents the interface on which the end-user accessed the application. The interface could be running on a browser, such as Microsoft Internet Explorer or Mozilla Firefox.

Figure 4.3 also shows various technologies that can be used for communication. For example, network protocol, back-end and user interface implementation tools. Nodes are connected using TCP/IP protocol on an ESB. The components are connected to each other using various technologies. For example, a client interface can be implemented on the application server using HTML, ASP.NET or JSP, running on a PC or in a mobile environment. An application server accesses the customer profile web services using SOAP over HTTP protocol. The connection between the service registry component, customer profile component and application component are handled by the WSDL and UDDI respectively.

Database drivers, such as the Java Database Connectivity (JDBC), can be used to connect the independent customer profile information database to the customer profile web services. The customer profile server is responsible for managing its own database and connection. This infers that an application accessing and utilising the customer profile service does not need to consider the customer profile data repository; and this is one of the characteristics of services interaction, as discussed by Erl (2005). The model's services interaction is based on the find-bind-execution of a service interaction model, as discussed in Section 3.4.

The purpose for proposing the model in this research was to act as an implementation guide and to visually describe how a customer profile can be implemented and utilised as a service in an SOA environment. The next section discusses the SOA design principles.



## 4.8 SOA DESIGN PRINCIPLES

Erl (2008) compiled SOA design elements that can be followed during the SOAD process, for example, design paradigms, design principles and design patterns. Eight SOA design principles, as explained by Erl (2008), were followed to implement the customer profile while using SOA in this research.

Design principles can be defined as generally highly recommended guidelines that are used to build a solution with certain goals in mind (Erl, 2008; Sharp, Rodgers and Preece, 2007). They provide designers with the do's and don'ts in designing a solution within a certain domain - in order to avoid coming up with a bad design. Design principles are effective when incorporated during SOAD as an extension to the service specification (Section 4.2.2). The following service design principles were followed during the design and implementation of the customer profile web services, in order to ensure that the end product was indeed developed using SOA principles (Erl, 2008):

- *Standardised service contract*: Services express their functionalities through the use of service contracts that stipulate what the service does. Therefore standardised service contract design plays a vital role in ensuring that the service is accessible and usable;
- *Service loose coupling*: Loose coupling promotes the independent nature of services in SOA while maintaining interoperability;
- *Service abstraction*: The ability of a service to hide implementation details and only expose the functionality it performs;
- *Service re-usability*: Service re-usability refers to the ability of the service to be re-used as an enterprise resource. This principle is helpful in deciding whether new services have to be built or existing ones have to be re-used;
- *Service autonomy*: Service autonomy refers to the ability of the service to be aware of the environment it is operating in. This principle promotes reliability and behavioural predictability which is very important for service interoperability in SOA;
- *Service statelessness*: Services in SOA needs to be stateless, that is, they should not keep the state of the previous interaction. This is very important in promoting loose coupling and service re-usability of the services;

- *Service discoverability*: Service discoverability is the ability of the service to be exposed and available for discovery by any service consumer. Figures 3.5 and 4.3 show how this can be achieved by the use of a service registry; and
- *Service composability*: This refers to the ability of the service to be composable with other services to perform different functionalities. Using languages such as BPEL (Section 3.5.5) a number of services have to be composable and performed in order to perform certain functionality. For example, in this research, the update profile service is composed of a number of services (Section 4.3.1.4).

Service inter-operability is the overarching service design principle that is achieved with a contribution from each of the eight service design principles outlined above (SOA Patterns, 2009; Erl, 2008). These design features were incorporated into the services implementation, by using the selected implementation tools, and were used to evaluate the SOA implementation, as will be discussed in Section 6.8.

## 4.9 SECURITY DESIGN

Apart from security being one of the challenges in SOA (Section 3.7.2), customer profile information requires security to prevent unauthorised access and unauthorised disclosure of customer information. Proper security policies implemented in the customer profile web services would provide security to the customer profile information's management.

The service security can be classified into a number of classes in SOA, for example, authorisation, authentication, confidentiality and integrity (Cotroneo, *et al.*, 2008; Birukou, *et al.*, 2007). Security in SOA can be applied during service interaction. This involves service registration, service discovery and service interaction (Cotroneo, *et al.*, 2008; Erl, 2005).

A number of service security protocols, such as the universal plug-and-play (UPnP) and service location protocol (SLP) (Cotroneo, *et al.*, 2008) exist; however this research did not implement a complicated service security. The motive for this was that service security, as a non-functional requirement in this project, was outside the scope of the project. Furthermore, the services were evaluated on a departmental network which has already implemented security policies.

#### 4.10 SUMMARY

This chapter has explained the process of SOAD using IBM's SOMA. The processes involved in SOMA, namely: service identification, specification and realisation have also been discussed. Customer profile services that were identified have also been explored and specified using the specification framework, as proposed by Terlouw and Maarse (2009). This chapter has also discussed the data model by examining customer profile tables and data types to be used in the implementation. A customer profile UML component and deployment model diagram that guides implementation, SOA design principles and the issue of security has, in addition, been explained.

Chapter 5 will cover the implementation of the two customer profile systems, one implemented using SOA and the other implemented using object-orientation. This chapter will present the implementation of both the customer profile systems and e-commerce web application services that utilise the customer profile services.

## CHAPTER 5: CUSTOMER PROFILING SYSTEM IMPLEMENTATION USING SOA

### 5.1 INTRODUCTION

Chapter 4 discussed the analysis and design of the customer profile using SOA. The chapter explored how customer profile services were identified and specified using IBM's SOMA (Section 4.3). An analysis of the customer profile functionalities was also presented in Chapter 4. The customer profile services include: register customer, add customer, get customer profile, update customer profile, identify customer and delete customer profile. The chapter further discussed the proposed customer profile UML component and deployment model (Figure 4.3).

This chapter will discuss the implementation of the customer profiling system using SOA. The customer profile implementation in SOA involves the identification, evaluation and selection of SOA implementation framework (SOAIF), IDE, programming language and DBMS used in this research. The implementation of the customer profile web services and an end-user e-commerce web application to utilise the back-end SOA-based customer profile services for personalisation will be explained.

The main focus of this chapter is to outline the SOA implementation of the customer profile, and how it is utilised by the e-commerce web application as a service in an SOA environment. This chapter's discussion represents steps 3 and 4 of the research methodology discussed in Figure 1.1.

### 5.2 IMPLEMENTATION TOOLS

This section discusses the implementation tools used in this research and the reasons for the selection of a particular tool. Various tools were used in this project to implement an SOA-based customer profile that was able to interoperate seamlessly and provided its services to the e-commerce web application. Implementation tools discussed in this section include: SOAIF, customer profile web services' implementation tools, end-user e-commerce web application implementation tools and the DBMS implementation tool.

### 5.2.1 SOAIFs

SOAs are normally implemented using an SOAIF that acts as a backbone to the implementation. Common SOAIFs are in the form of ESBs, as discussed in Sections 3.3 and 3.5.2. An investigation and evaluation of available SOAIFs was conducted. The aim was to find a suitable framework to be used in this research that would support the research objectives.

Several commercial and open source SOAIFs exist. Examples of commercial SOAIFs include: IBM Web Sphere, Microsoft BizTalk server and Oracle SOA. Open source SOAIFs include: Apache ServiceMix, Logic Blaze, Elemenope and Project Open ESB. The evaluation conducted focused only on open source frameworks. The objectives for evaluating open source frameworks were:

- It was a directive from the initial research sponsors to have the project implemented using open source implementation tools;
- Open source tools are widely used in organisations to develop systems. For example, the BBC news website and Forrester research reported that the British Government and other European countries are promoting the use of open source tools (BBC, 2009; Hammond and Lo Giudice, 2008);
- Open source tools are suitable for research purposes, as there are no upfront investments made as compared to commercial software development tools (Java.sun.com, 2009); and
- Open source tools can be customised to suit particular users' needs because the code is available.

The following open source SOAIFs were identified, analysed and evaluated:

- *Apache ServiceMix*: Apache ServiceMix is an open source ESB that contains SOA implementation functionalities (Apache.org, 2009). Apache ServiceMix is built following the JBI specification JSR 208. As discussed in Section 3.5.4, JBI enables different components and services to interoperate seamlessly (Java.sun.com, 2005). Figure 5.1 is a diagram of Apache ServiceMix ESB. Apache ServiceMix ESB supports various SOA technologies, as discussed in Section 3.5, for example, BPEL, as shown in Figure 5.1;

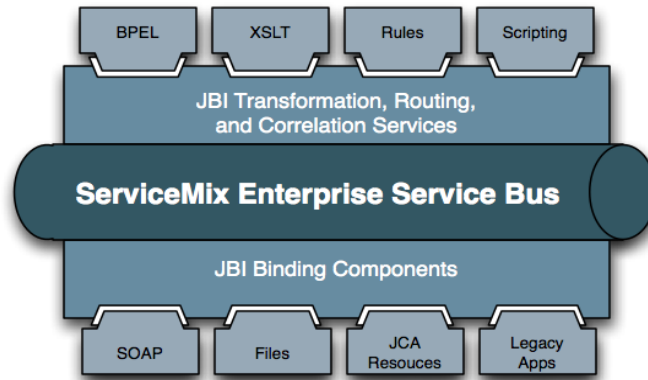


Figure 5.1: Apache ServiceMix ESB (Apache.org, 2009)

- *Open ESB*: Open Enterprise Service Bus (Open ESB) is an open source SOAIF developed by an open source community under the supervision of Sun Microsystems (Java.sun.com, 2005). Open ESB is strongly linked with the Netbeans IDE and the GlassFish Application Server (Java.sun.com, 2009). GlassFish ESB is an Open ESB distribution containing tested core run-time and the most essential components that come with Netbeans IDE (Java.sun.com, 2009). Open ESB is based on JBI architecture. Figure 5.2 shows a diagram of Open ESB; and

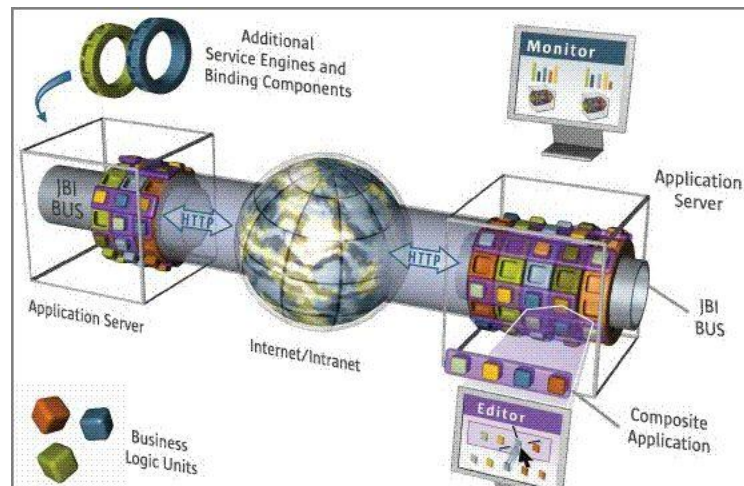


Figure 5.2: Open ESB architecture (Java.sun.com, 2009)

- *Elemenope*: Elemenope, pronounced as L-M-N-O-P is an Enterprise Application Integration (EAI), SOA, and a general messaging framework implementation framework that is freely available for developers (Elemenope.org, 2009). As opposed to Project-Open ESB and Apache ServiceMix, Elemenope does not implement the JBI.

The following sub-sections discuss the how the evaluation was conducted, the metrics that were used to evaluate the frameworks and the results of the evaluation. Apache ServiceMix and Open ESB were the frameworks that were tested and evaluated as they were similar and are widely used, as compared with Elemenope ESB.

#### 5.2.1.1 SOAIF evaluation procedure

During the evaluation process a service was developed in order to test the frameworks' capabilities in handling services deployment, as well as their interaction. The application was developed using Java Web Services (JWS) in Netbeans IDE 6.5 and it was deployed and tested using the two afore-mentioned frameworks.

#### 5.2.1.2 SOAIF evaluation metrics

Software engineering and usability metrics were used to evaluate the SOAIFs in order to determine a suitable framework for the purposes of this project. The metrics that were used for evaluating the frameworks were as follows:

- a. Software engineering metrics included the following (Sommerville, 2007):
  - Open source vs. commercial tools;
  - Dependability; and
  - Service re-usability.
- b. Usability engineering metrics included the following (Tullis and Albert, 2008):
  - Learnability and efficiency;
  - Overall user experience; and
  - Error recovery.

A number of instruments were used to measure the metrics, for example, the number of steps required to have the service deployed was recorded and used as a measure for learnability and efficiency of a framework. Error reporting by the framework was used as a measure for error recovery and documentation, and available support in terms of training and forums was used as part of the measure for dependability, as well as learnability.

The following section discusses the results of the evaluation and the selected framework used in this research.

### 5.2.1.3 Evaluation results

Data analysis was conducted when all the evaluation metrics data for the two SOAIFs were collected and the results indicated that the Open ESB provided better tools to support the SOA implementation of this research. Open ESB is distributed as GlassFish ESB and is integrated in Netbeans IDE together with the GlassFish application server. This made it easier to have services developed, tested and deployed within the IDE. The graphical user interface in project Open ESB's GlassFish Application Server console (Figure 5.3) makes it easier to learn and use than the Apache ServiceMix which has a command line interface (Apache.org, 2009; Bayer, 2008).

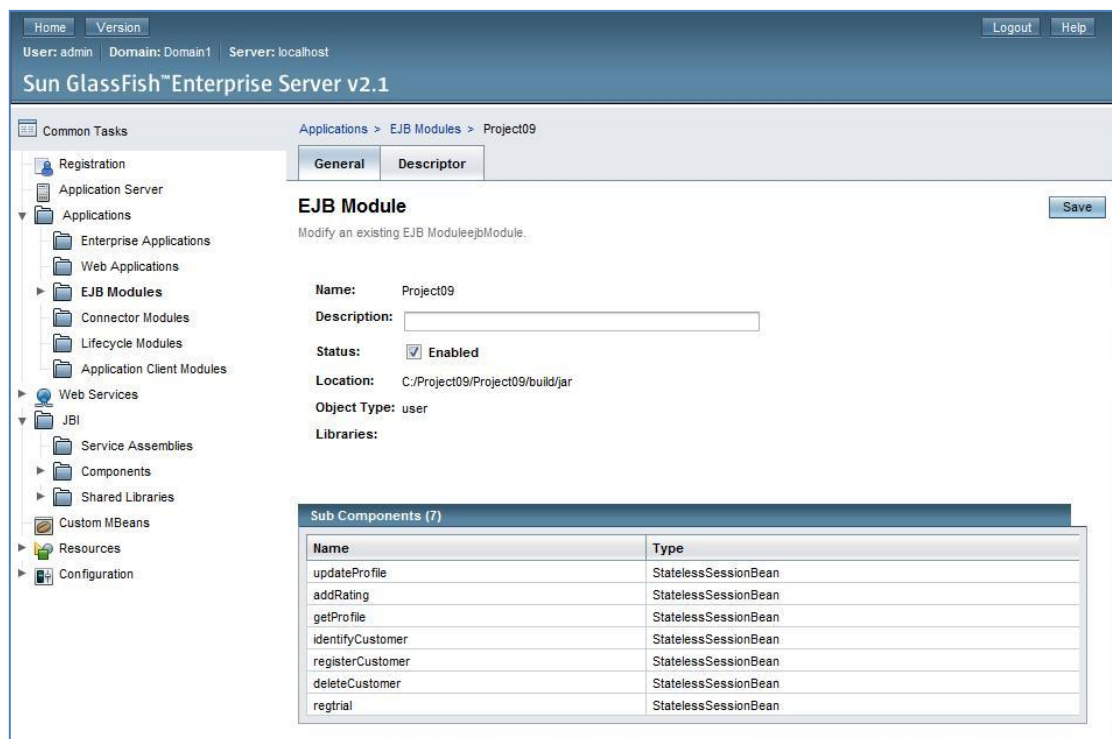


Figure 5.3: GlassFish application server console

Both project Open ESB and Apache ServiceMix frameworks were found to be based on JBI. However, Open ESB had better JBI integration tools than Apache ServiceMix ESB, as can be seen from Figure 3.9; this shows the JBI component for Open ESB from the Netbeans IDE 6.5 interface. Open ESB is developed by the community, and as such it has more documentations and forums than Apache ServiceMix (Bayer, 2008; Java.sun.com, 2009).



Open ESB enjoys extensive support from major IT vendors, such as IBM, Sun and Microsoft (Java.sun.com, 2009; IBM, 2007). Furthermore, free online training is offered to registered members using Open ESB (Java.sun.com, 2009).

#### 5.2.1.4 Limitations of the evaluation

Lack of SOA experts who could have assisted evaluating the frameworks and lack of well-known existing SOA systems constituted the main limitations in this evaluation.

### 5.2.2 Back-end and front-end implementation tools

The customer profile web services were implemented using two types of implementation tools, namely: web services implementation tools and database management tools. This section discusses the Netbeans IDE as the customer profile web services implementation tool, Microsoft Visual Studio that was used to develop the e-commerce web application that utilised the customer profile web services and Microsoft SQL Server 2008 Express as the DBMS used in the implementation of the research.

#### 5.2.2.1 Netbeans IDE

Netbeans IDE is a Sun Microsystems IDE software that is available at no fee from the Internet; and the current version is 6.7.1 (Netbeans IDE, 2009). Netbeans IDE supports a number of technologies that can be used to develop desktop, web and mobile applications, for example, Java Standard Edition, Java FX, Java Micro Edition, Ruby, Groovy, PHP, and Java Enterprise Edition. Furthermore, Netbeans IDE supports the following servers that can be used as running environments for applications: GlassFish Application Server V 2.1 and V 3 Prelude, and Apache Tomcat server (Netbeans IDE, 2009).

Figure 5.4 shows the interface for Netbeans IDE version 6.5 that was used to develop the customer profile web services. The figure shows a toggle for project explorer on the left-hand side and the code window on the right-hand side.

Java Development Kit (JDK) plug-in software was required before the installation of Netbeans IDE, and the JDK 1.7 was the version that was used. During installation various features of Netbeans could be customised, for example, adding or removing supported technologies.

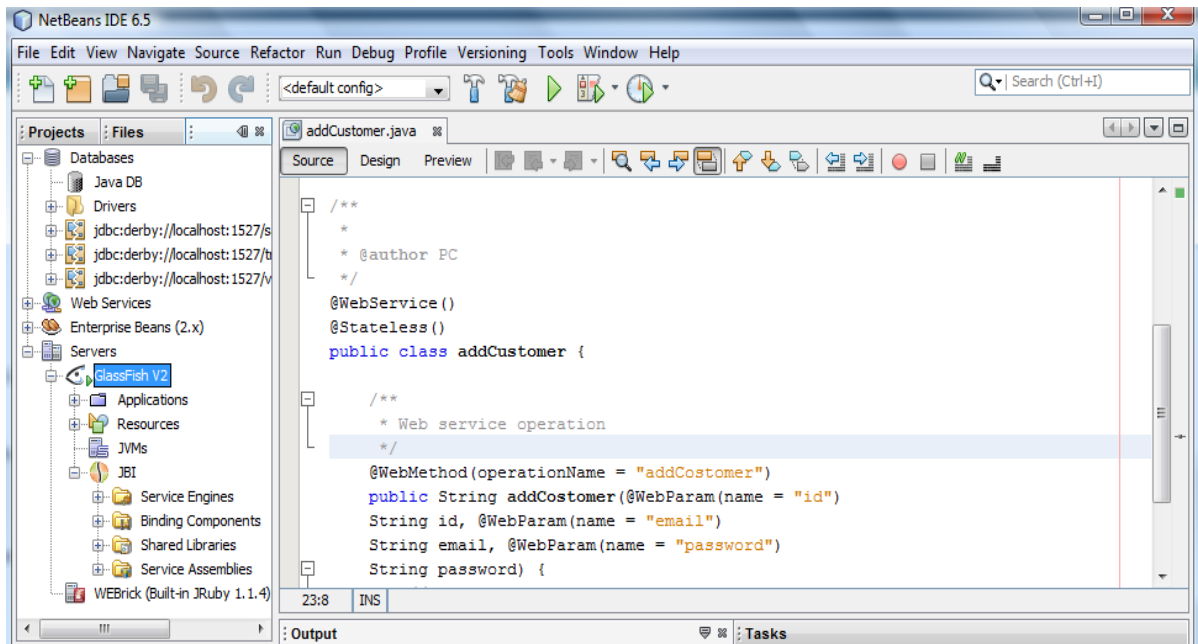


Figure 5.4: Netbeans IDE 6.5 interface

### 5.2.2.2 Microsoft Visual Studio IDE

Microsoft Visual Studio is a commercial IDE from Microsoft (Microsoft.com, 2009). The IDE provides a programming environment for a number of languages, for example, Visual Basic, J# and C# that developers can use to implement desktop, web and mobile applications (Microsoft.com, 2009). The latest version of visual studio is Microsoft Visual Studio 2010 which is currently available in a beta version (Microsoft.com, 2009).

Microsoft Visual Studio 2008 was the version that was used in this project to develop an e-commerce web application that interacted with JWS-based customer profile. Figure 5.5 shows the interface for Microsoft Visual Studio 2008.

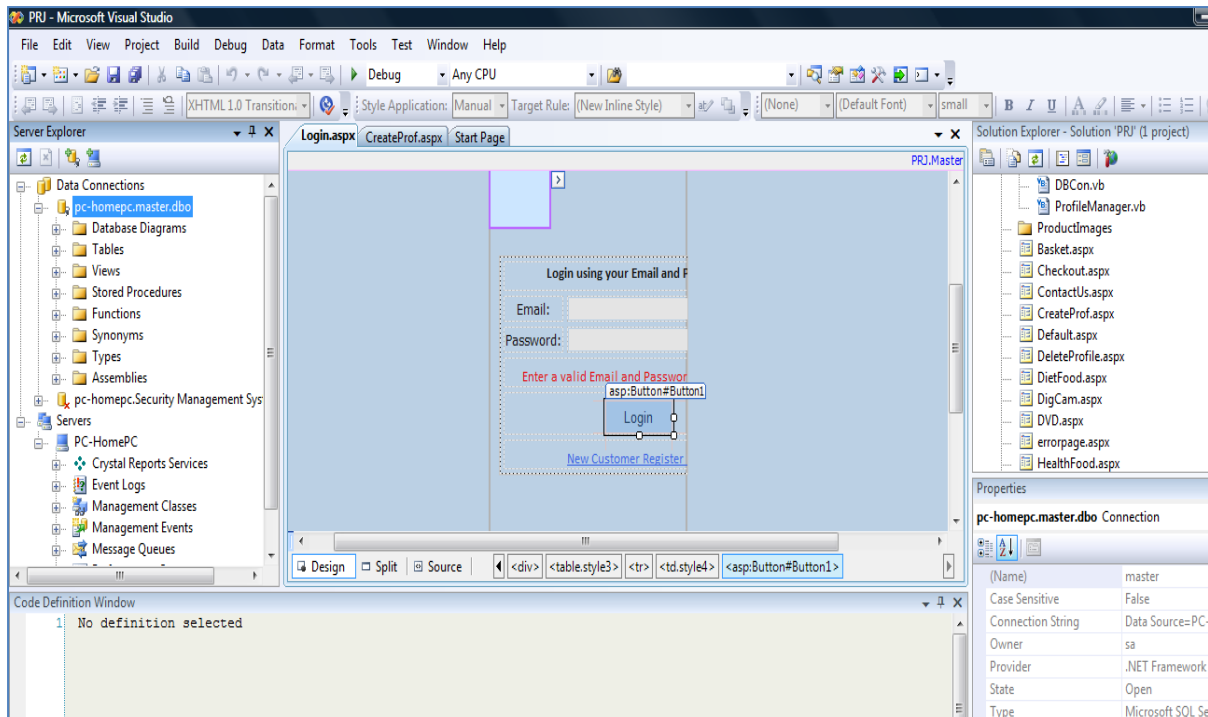


Figure 5.5: Microsoft Visual Studio 2008 interface

An e-commerce web application was implemented using Microsoft .NET framework technologies to utilise the JWS-based customer profiling services. The following were the objectives for choosing .NET framework as the implementation framework for e-commerce web application:

- To test JWS-based customer profile services interoperability capabilities between implementation frameworks, such as Sun Java framework and the .NET framework; and
- As a proof of the concept for the service nature of the customer using SOA. That is, different applications can access customer profile as a service, regardless of implementation platforms (Figure 4.3).

### 5.2.2.3 Microsoft SQL Server Express

A database was required to store customer profile information, as well as the e-commerce application data. Microsoft SQL Server 2008 Express was selected as the DBMS used in this research to implement the database. Microsoft SQL Server possesses a number of advantages that made it a suitable DBMS for this project, for example (Microsoft.com, 2009):

- *Supports web-based applications:* The research domain was customer profile for e-commerce web applications, and therefore it was imperative to use a DBMS that supports web applications;
- *Supports rich extensible mark-up language (XML):* Most SOA standards used in this research are based on XML (Section 3.5) and as such they work better with a Microsoft SQL server;
- *Is a simplified database;* and
- *Is highly scalable:* Depending on the number of new customers, a highly scalable DBMS is crucial for the success of the customer profiling system.

Microsoft SQL Server 2008 Express was installed and used as the customer profile data repository. Microsoft SQL Server 2008 Express database components that were installed included: database management tools, database engine and SQL client connectivity SDK. The following comprised the required supporting software for the successful installation of Microsoft SQL Server 2008 Express:

- Microsoft .NET Framework 3.5 or above;
- Windows PowerShell;
- Windows installer 4.5 SDK; and
- Microsoft SQL Server 2008 Express installation files.

The files for all this supporting software can be downloaded at no fee from the Microsoft website (Microsoft.com, 2009).

Microsoft SQL Server Java database connectivity (JDBC) driver is required for Java applications to use Microsoft SQL Server database. However, at the time of implementation, Microsoft SQL server JDBC driver could not work with Netbeans IDE 6.5 and jTDS driver was the driver that was used to link the database and Netbeans IDE 6.5 (jtds.sourceforge.net, 2009). JTDS is an open source Java driver for Microsoft SQL Server and it is added into the Netbeans IDE 6.5 by copying and adding a *.jar* file into the application's library folder.

The implementation tools discussed were evaluated based on characteristics that were suitable to support the implementation of the customer profile model, as discussed in Section 4.7. The next section discusses how the implementation of the model (Figure 4.3) was conducted using the implementation tools already discussed in Section 5.2.

## 5.3 IMPLEMENTATION

Implementation of the customer profile services was done by implementing the SOAP JWS as Enterprise Java Beans (EJB) in Netbeans IDE 6.5, and the e-commerce website was developed using ASP.NET in Microsoft Visual Studio 2008. This section discusses the implementation process by examining how the customer profile services and e-commerce website functionalities were implemented using the tools and frameworks discussed in Section 5.2. Database implementation, using Microsoft SQL Server 2008 Express, is also discussed in this section.

### 5.3.1 Customer profile web services implementation

Section 4.3 discussed the following customer profile services identified during customer profiling SOAD: register customer service, add customer service, get profile service, update profile service, identify customer service and delete customer service. This section provides the details on how these services were implemented, with exposed methods and by using JWS. The motive for choosing JWS was that web services can be easily developed using JWS in Netbeans IDE and can be deployed using GlassFish Application Server.

In addition, JWS and Netbeans IDE 6.0 provide rich components for designing and deploying component-based systems. For example, Netbeans has a graphical user interface for designing both WSDL and BPEL (Java.sun.com, 2005).

A number of ways to develop Java Platform, Enterprise Edition (Java EE) applications in Netbeans IDE exist; however, EJB modules provide better tools for exposing business functions, as web services (Netbeans IDE, 2009). This research implemented the JWS using EJB modules. SOAP web services were the preferred type used to implement the customer profile web services as compared with REST web services. This was discussed in Sections 3.5.1 and 4.2.3.

Figure 5.6 shows the general outline of a JWS that was implemented as EJB modules for the customer profile. The figure shows what classes are imported for a JWS, such as, `javax.jws.WebMethod`, `javax.jws.WebParam`, `javax.jws.WebService` and `javax.ejb.Stateless` that is used for EJB implementation. The first three classes can be imported by the following single statement, `javax.jws*`. These classes contain definitions for different Java API for XML Web Services (JAX-WS) annotations that are

useful in defining classes to be exposed as web methods for the web services (Java.sun.com, 2005).

Annotations provide data on the program that are not part of the program itself (Java.sun.com, 2009). The following annotations were implemented in the customer profile services (Java.sun.com, 2005):

- `@WebService`: The purpose of this annotation is to tell the Java interpreter that the following code is to be exposed or published as a web service;
- `@Stateless`: The purpose of this annotation is to implement the statelessness nature of SOA web services;
- `@WebMethod`: The purpose of this annotation is to expose a method as a web service operation; and
- `@WebParam`: The purpose of this annotation is to declare variables that are mapped as input parameters for a web method.

A number of annotations exist for the JAX-WS specification. However, the above annotations were relevant to the goals of the customer profile web service implementation (Java.sun.com, 2005).

```
package <name>;
import javax.ws.WebMethod;
import javax.ws.WebParam;
import javax.ws.WebService;
import javax.ejb.Stateless;
@WebService ()
@Stateless ()
public class <service name>
{
    public <constructor> () {}
    public <individual classes>
        {<code to perform functions>}
    @WebMethod (operationName = "<web service function>")
    public <return type> <web service function>(@WebParam(name = "<parameter>")
        <parameter type> <parameter name>)
        { //TODO write your implementation code here: Return <value>; }
}
```

Figure 5.6: A general outline of customer profile web services

The services discussed in Section 4.3.1 were implemented using the structure in Figure 5.6. Different classes, for example, a class that accepted customers' login details (email and password) and checked whether the customer exists in the database, were created in the JWS EJB module, as shown in Figure 5.6. The individual classes were called in the `@WebMethod`

section to perform various functions and provide a computational value to the calling service. This computational value, if any, was returned at the end of the web service's operation. Once a service had been implemented it was tested with test data and input parameters, if any, to ascertain whether it had performed the intended function correctly.

Netbeans IDE 6.5 provides a quick way to test the functionality of implemented JWS. Once the service had been implemented, it had to be tested and deployed to make it available for use by other services. The service tester, as shown in Figure 5.7, is generated when a service is being tested. The tester requests input parameters and shows their types, provides textboxes that are used to capture the input and a method call that implements the service is indicated by a command button located to the left of the first textboxes (if any).



Figure 5.7: Web service testing page

The testing process required the following to be working and running:

- GlassFish application server;
- Microsoft SQL Server; and
- An active JDBC connection.

The testing procedure involves an application to be deployed first. There are two ways that can be used to test a web service. First, by right clicking a deployed service in the IDE and selecting “Test web service”; or secondly, by entering the following generic web service address in a browser address bar of a web browser:

`http://<Server Name>:<Port Number>/<Web Service Name>/<Web Service Name>?Tester`

Figure 5.7 shows a generated screenshot of a web service tester. The figure shows a link to the WSDL file, the exposed method(s) and textboxes for input parameters, if the service requires any input. When valid input parameters are entered and the method is invoked by clicking on the button on the left of the input parameter text boxes, the web service performs the computation and the results are displayed.

### 5.3.2 Customer profile web services clients' implementation

An e-commerce web application was developed with the purpose of utilising the customer profiles web services. The application was developed using ASP.NET 2.0 and Visual Basic in Microsoft Visual Studio 2008. The reasons for the choice of these implementation tools have already been discussed in Section 5.2.2.2. The roles of the e-commerce website in this research were:

- To provide an interface for end-user evaluation of SOA and client-server back-ends customer profile; and
- To demonstrate that an SOA-based customer profile can be utilised by any application implemented using any programming language as a plug-in and accessing the customer profile as a service.

The website had the following implemented pages:

- Login;
- Registration;
- Customer-product knowledge profile creation;
- Product;
- Shopping basket;
- Checkout;
- View profile;
- Delete profile;
- Contact us; and
- Logout.

Figure 5.8 shows an example of the interface that was implemented in the e-commerce web application. The figure shows a red wine product page with e-commerce functionalities at the top of the page, product listing in the left-hand section of the page and varying levels of



product information displayed on the right-hand section of the page. The middle section of the page shows the product pictures, name, unit price and quantity drop-down list to enable users to select quantities to purchase.

The e-commerce website that was implemented had two sets of pages. The first set consisted of pages that implemented e-commerce functionalities, such as, shopping basket, check out and product listing. The second set consists of the pages that implemented the interaction with the customer profile services to perform customer profiling functionalities, such as, creating a profile, retrieving a profile and deleting a profile.

The first set of pages used customer profile information obtained by the second set whenever a customers' profile was needed during their operation. For example, product listing pages used the product knowledge profile ratings obtained by the home page to display personalised product information for a particular customer. This was done to reduce the number of repetitive calls to services in order to improve the system's response time. Table 5.1 (Section 5.4) shows how the pages interacted with the customer profile web services.



Figure 5.8: An e-commerce application red wine page

### 5.3.3 Database implementation

Two databases were implemented in this project. The first database was a customer profile database and the second database was the e-commerce application database. The databases were implemented using Microsoft SQL Server 2008 Express edition, as discussed in Section 5.2.2.3. The customer profile database was used to store customer profile information and the e-commerce application database was used to store product and e-commerce interaction information, for example, product information, shopping history and basket information. The structure of the customer profile database has already been discussed in Section 4.4.

A number of stored procedures and custom SQL commands were implemented to enable CRUD operations on the databases. This section discussed the implementation of various components of the customer profiling system within the nodes (Figure 4.3). The next section will discuss how the components communicated and interacted with each other in an SOA paradigm, both within and outside the nodes.

## 5.4 E-COMMERCE WEB APPLICATION AND CUSTOMER PROFILE WEB SERVICES INTERACTION

The e-commerce web application interacted with the customer profile web services following the service interaction model, as discussed in Section 3.4, where the e-commerce web application is the service consumer and the JWS-based customer profile system is the service provider (Figure 3.5). A customer profile registry was implemented and the registry contained the customer profile services specifications, as discussed in Section 4.3.2.

SOAP over HTTP was used to transport parameters and computation results between the service consumer and the service provider. The parameters and generated output are normally transported and exchanged between services in a standard text format, as was discussed in Sections 3.5.3 and 4.4. Figure 5.9 shows a screenshot of the SOAP request-and-response for an identify-customer service that takes *email* and *password* as input parameters and conducts the computation to give a customer *id* as the output when a customer exists in the database.

<b>SOAP Request</b>
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;S:Header/&gt;   &lt;S:Body&gt;     &lt;ns2:Search xmlns:ns2="http://Project09/"&gt;       &lt;email&gt;10&lt;/email&gt;       &lt;passwd&gt;fe&lt;/passwd&gt;     &lt;/ns2:Search&gt;   &lt;/S:Body&gt; &lt;/S:Envelope&gt; </pre>
<b>SOAP Response</b>
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;S:Body&gt;     &lt;ns2:SearchResponse xmlns:ns2="http://Project09/"&gt;       &lt;return&gt;0&lt;/return&gt;     &lt;/ns2:SearchResponse&gt;   &lt;/S:Body&gt; &lt;/S:Envelope&gt; </pre>

Figure 5.9: SOAP request and response

Table 5.1 shows how the various web pages of the e-commerce web application interacted with the customer profile web services. The table shows the interactions in terms of the parameters passing between the service consumer and the service producer.

Web page	Service Provider	Input Parameters	Output Parameters
Login	Identify Customer	Email, password	Customer ID
Home	Get profile	Customer ID	Product knowledge ratings
Register customer	Register Customer	Customer biographical details	Customer ID
Customer product knowledge profile creation	Add Customer profile	Customer ID, customer product knowledge ratings	Null
View profile	Get profile	Customer ID	Product knowledge ratings
Check out	Update profile	Customer ID, customer activity log	Null
Delete profile	Delete profile	Customer ID	Null

Table 5.1: Web pages and customer profile service clients' link

### 5.4.1 Creating a link between service consumer and service producer in Microsoft Visual Studio

A web reference has to be made in the main program or the service that intends to interact with a service provider. Microsoft Visual Studio provides a graphical user interface (GUI) wizard for adding a web reference to a service. An explicit path for the web services WSDL file is required to get access to the service's exposed method and the following is the general syntax for the web reference to a WSDL file (Colgrave, Heritage, Marsh and Skinner, 2007):

`http://<Server Name>:<Port Number>/<Web Service Name>/<Web Service Name>?WSDL.`

Web services' exposed methods are displayed by name when reference has successfully been made to a provider web service. The default name for a web reference in Visual Studio is normally the name of the web services' source server. A number of supporting files are created in a Visual Studio web reference folder for each web reference made, for example, *.disco* and *.xsd* (Colgrave, *et al.*, 2007).

These extra files store other information of a referenced web service, such as the location information for the web service. This process enables a web service to be utilised by another service.

### 5.4.2 Utilising a web service in Microsoft Visual Studio

Once a web reference has been made to a service, an implementation can be done to access and utilise the service's methods. Figure 5.10 shows the implementation of accessing web services exposed methods. The methods to be exposed are declared by having a *@WebMethod* annotation in a JWS. Section 5.3.1 discussed the web services' implementation, using Netbeans IDE. The code for web service client (e-commerce web application) was developed using Microsoft Visual Basic and ASP.NET 2.0, as discussed in Section 5.3.1.

```
Dim wsClient As New <Web Reference Name>.<Web Service Name>()
wsClient.<Exposed Method>(Parameter list)
```

Figure 5.10: Accessing web services methods

Successful interaction with the customer profile web services depended on proper parameters passed. The parameters passed to a customer profile service provider had to conform to the specification, as was discussed in Section 4.3.2.

## 5.5 SUMMARY

This chapter has discussed the implementation of the project by looking at the selection of implementation tools and how the tools were used to develop the customer profile in an SOA environment. The chapter has discussed the Open ESB as the selected SOAIF that was used for implementation. Netbeans IDE 6.5 was selected as the IDE to be used to implement back-end customer profiling services; and Microsoft Visual Studio 2008 was used to implement the customer profile service clients. Microsoft SQL Server 2008 Express was selected as the DBMS used to implement data repositories for the project.

The following components of the system were successfully implemented using the mentioned implementation tools: customer profile web services, namely: register customer, add customer, get profile, update profile, identify customer, and delete profile, an e-commerce website that utilised the customer profile web services, a database for storage of customer profile information and e-commerce application data. Discussion on how services were implemented, and how they were accessed and utilised by the service client, the e-commerce application, has already been presented.

Chapter 6 discusses the research project's evaluation and results analysis. The chapter explains the pilot study and the main evaluation techniques that were utilised in this research.

# CHAPTER 6: EVALUATION AND ANALYSIS OF RESULTS

## 6.1 INTRODUCTION

Chapter 5 discussed the implementation of the customer profile using SOA and the e-commerce web application that was developed to test the back-end SOA-based customer profile. The chapter also reviewed the implementation tools selected for the project, including the SOAIF, IDE and DBMS (Section 5.2). Furthermore, the chapter explained how the customer profile web services were implemented and utilised by an online website to provide personalised product information to customers according to the customer's current and up-to-date profiles in an SOA environment (Sections 5.3 and 5.4).

This chapter focuses on the evaluation of the project by discussing the evaluation procedure employed in the research. The purpose of the evaluation was to assess the usability of the customer profile web services implemented and how the implementation met SOA design principles. The evaluation was conducted in three phases:

- The first phase was a pilot study conducted by experts to test the JWS-based customer profile and the web application user interface before the main evaluation;
- The second phase was the main end-user evaluation that was conducted with a sample of 30 participants; and
- The third phase was an SOA implementation evaluation that was conducted based on the SOA design principles, as discussed in Section 4.8.

The end-user evaluation (second evaluation) was conducted in the NMMU Department of Computing Sciences Usability Laboratory and eye tracking was used to capture additional evaluation metrics. Steps 5 and 6 of the research methodology discussed in Figure 1.1 are represented in this chapter's discussion.

## 6.2 NMMU DEPARTMENT OF COMPUTING SCIENCES USABILITY LABORATORY

As opposed to the pilot study that was conducted online at the user's convenience (Section 6.5.1.2), the main end-user evaluations were conducted in the NMMU Department of Computing Sciences Usability Laboratory. A usability laboratory is a special laboratory that

is designed for usability testing (Sharp, *et al.*, 2007). This section presents an overview of the NMMU Department of Computing Sciences Usability Laboratory.

A usability laboratory provides a controlled environment in which to conduct computer system usability evaluations. The NMMU laboratory is equipped with various evaluation facilities, such as, microphone, video and voice recorders and eye tracking equipment. The laboratory consists of an observer and a participant room separated by one-way glass that enables the observer to see the participant, but not vice versa.

The separate observer and participant rooms ensure freedom and fewer disturbances, as the participant performs the evaluation tasks. The observer and participant communicate by using the microphones and speakers installed in the laboratory. Figure 6.1 shows the observer room on the left-hand side and the participant room on the right-hand side, and some of the equipment used in the laboratory.



Figure 6.1 NMMU Department of Computing Sciences Usability Laboratory observer and participant rooms

The main aims for conducting the evaluation in a controlled environment were:

- To avoid bias by exposing users to the same environment, for example, the same browser and machine specification, such as, memory, processing speed and operating system; and
- To use the eye tracking equipment to collect additional evaluation metrics.

The next section will discuss the instruments that were used to conduct this evaluation.

## 6.3 EVALUATION INSTRUMENTS

The evaluation instruments used in this research include the data-collection instruments, such as questionnaires and eye tracking, and the statistical instruments that were used to analyse the data collected. This section discusses the tools that were used.

### 6.3.1 Questionnaires

Questionnaires are a recognised way of collecting research data (Flick, 2009). In this research, questionnaires were used to elicit participants' views on the implemented customer profile system. Three questionnaires were compiled: a background questionnaire and two different post-task evaluation questionnaires, one administered after each system had been evaluated.

The two post-task evaluation questionnaires were formulated following the guidelines discussed by Tullis and Albert (2008) and Sharp *et al.*, (2007) (Appendices G and H). A combination of both semantic differential and Likert scales were used in the two post-task evaluation questionnaires (Tullis and Albert, 2008; Sharp, *et al.*, 2007). Both qualitative and quantitative data were collected by means of the questionnaires (Section 1.6.4).

This section discusses the three questionnaires used in this evaluation.

#### 6.3.1.1 Background questionnaire

The background questionnaire sourced participants' background details. The questionnaire's three sections (A, B and C) sourced the following information from the participants (Appendix E):

- *Biographical details:* Section A collected the biographical details of the participants, for example, name, gender, age, degree registered for, year of study, ethnicity and home language;
- *Computer and Internet browsing experience.* This section (B) gathered the participants' computer and Internet experience by asking the participants how long they had been using a computer and the Internet, where they had accessed the Internet, the number of hours spent on the Internet per day, and how they rated their Internet experience, for example, as novice, intermediate or advanced; and
- *Online shopping and registration experience:* Information on the participants' online shopping and registration experiences was collected in Section C of the background



questionnaire. Specific questions examined how many online shopping websites the participant had visited during the past year, what type of products and from which websites the participant had purchased, and a list of websites on which the participant had ever conducted online registration.

### 6.3.1.2 Post-task evaluation questionnaire first system

This questionnaire was used to collect information on the participants' opinion on the customer profiling component of the system. Using various semantic differentials, the questionnaire specifically asked participants to rate the following components in four sections (Sections C, D, E and F in Appendix G):

- *Initial and updated profiles:* Participants recorded their initial and updated customer profiles in Section C. The section further requested participants to state whether they agreed with their profiles on a 5-point Likert scale, with 1 being strongly disagree and 5 being strongly agree;
- *Creating a profile:* Section D 1.1 to D 1.3 queried how sufficient and easy participants had found creating a profile to be;
- *Viewing a profile:* The importance of viewing a profile was measured in Section D 1.4 to D 1.6. Furthermore, this section asked whether the participants regarded the displayed profile information to be adequate or not;
- *Updating a profile:* Section D 1.7 to D 1.9 focused on the participants' opinion on the updating of the profile; for example, did they find the process intuitive? And what was the importance of having the profile updated?
- *Deleting a profile:* This section (D 1.10 to D 1.11) examined whether or not it was important for participants to be able to have a profile deleted whenever required;
- *Overall task satisfaction:* Section E used a modified After-Scenario Questionnaire (ASQ) (Tullis and Albert, 2008) to collect the participants' opinion of the systems' functionalities and task completion. A 5-point Likert scale, with 1 being strongly disagree and 5 being strongly agree, was used to collect responses for this part; and
- *Customer profiling components (Qualitative data):* The customer profiling component in Section F collected information on participants' suggestions on any additions or deletions that could be made to the customer profile components.

A 5-point semantic differential scale was used for the create profile, view profile, update profile and delete profile sections. The participants completed this questionnaire after they had completed the interaction with the first system.

### 6.3.1.3 Post-task evaluation questionnaire second system

The two systems used by the participants during evaluation had the same web user interface, but the underlying architecture employed was either client-server (CS) or SOA. This questionnaire aimed at measuring the participants' perceived differences between the two customer profiling systems. The questionnaire specifically asked questions on the following (Sections H, I and J in Appendix H):

- *System performance*: Section H examined the systems' page loading times, response times, task completion times, ease with which tasks could be completed and the overall efficiency of the two systems;
- *Overall system assessment*: The overall system assessment section (I) recorded the participants' general feelings about the system by asking participants questions with semantic differential responses, for example, very terrible to wonderful on a scale of 1 to 5;
- *General comments (Qualitative data)*: The final section (J) sourced the participants' general comments on the systems. Specifically, the questionnaire asked about any observed differences between the systems, what the users liked most and what they liked least about the two systems, any comments the participants had concerning the customer profiling components (Section 4.3.1), and any general comments and suggestions.

The participants completed this questionnaire after they had finished interacting with the second system. In order to collect other evaluation information that was not possible to collect by simply using questionnaires, for example, the actual page loading times; eye-tracking data was used for this purpose. The next section discusses an overview of eye tracking and how it was used in this evaluation.

### 6.3.2 Eye tracking

Eye tracking is a usability method that is commonly used to collect additional usability metrics, such as, participants' eye movements and behaviour metrics during computer systems usability evaluation (Tobii.com, 2009; Tullis and Albert, 2008). Additional metrics

obtained by using eye tracking provided added information on usability findings and sometimes provide further information on why users behaved or performed in a certain way during the evaluation procedure (Tullis and Albert, 2008).

Figure 6.2 shows the Tobii T60 eye tracker that is available in the NMMU Department of Computing Sciences Usability Laboratory. The eye tracking component for the Tobii T60 eye tracker is integrated into the 17" thin-film transistor (TFT) monitor, as shown in Figure 6.2 (Tobii.com, 2009). This allows participants to behave as if they are using normal desktop computers. Eye tracking equipment uses a combination of an infrared video camera and infrared light sources, in order to track where the participant's eye is focusing on the screen (Tullis and Albert, 2008).

The first step in using the eye tracker is calibration, in which the eye tracker tests various positions where the participant focuses on the screen (Pernice and Nielsen, 2009; Tobii.com, 2009). The Tobii T60 eye tracker automatically asks for re-calibration if the previous calibration was not perfect.



Figure 6.2: NMMU Department of Computing Sciences Usability Laboratory Tobii T60 Eye tracker

Different eye tracking data analysis software exists. However, Tobii Studio is the eye-tracking data analysis software that works best with the Tobii T60 eye tracker, and was used in this research (Tobii.com, 2009). Eye tracking software, for example, Tobii Studio, produces a number of reports on the participants' eye tracking recordings. Examples of reports that can be produced include: heat maps, gaze plots and graphs (Pernice and Nielsen, 2009; Tullis and Albert, 2008). Heat maps, which show the participants' eye fixation length

by using colour, was the report that was used in this research because these reports are easy to produce and provide an overview of the areas on which participants' mostly focused. A heat map provides qualitative data (Pernice and Nielsen, 2009).

Area of interest (AOI) is a frequently used evaluation metric used with eye tracking (Pretorius, Calitz and van Greunen, 2005). However, for the purpose of this research, eye-tracking was used to collect information such as, participants' eye fixation points and the actual page loading times of the SOA-based and CS-based customer profiling systems. The reason for not including AOIs was that the emphasis of this research evaluation was not on the user interface, but rather the back-end customer profiling functionality. This additional information was used to validate what the participants answered during the completion of the paper-based post-task evaluation questionnaires.

The data collected by both the questionnaires and eye tracking procedures were used in the statistical analysis that was conducted. Mr. Venter of the Unit for Statistical Consultancy (USC) at NMMU was consulted to assist with the statistical analysis of the data collected in this study as will be discussed in Section 6.6. The next section outlines the statistical instruments that were used to analyse the data collected.

### 6.3.3 Statistical instruments

Microsoft Excel 2007 was the data-capturing and analysis software that was used in this evaluation. The reason for selecting Microsoft Excel as a data analysis tool in this research was that it is widely available and generally used (Tullis and Albert, 2008). In addition, a number of useful statistical calculations for the analysis of this research are available in Microsoft Excel, for example, mean, standard deviation and mode.

This section discusses both the descriptive and inferential statistical methods used in this research.

#### 6.3.3.1 Descriptive statistics

Descriptive statistics provide simple summaries of the data obtained from the sample. In other words, descriptive statistics summarise the basic features of the data obtained in a study without giving their meaning (Research methods, 2009; Tullis and Albert, 2008; Steinberg, 2008). Measures of central tendency and measures of dispersion are two commonly used descriptive statistics used in data analysis (Tullis and Albert, 2008).

The following are the examples of descriptive statistics parameters that were used in this evaluation:

- *Mean*: The mean calculates the average of the occurrences or the data items of a dataset (central tendency);
- *Median*: The median indicates the middle item in the dataset, that is, half of the data items are below this number and the other half of the data items are above this number (central tendency);
- *Mode*: This is the number or items appearing most frequently in the dataset (central tendency); and
- *Standard deviation (Std. Dev)*: This calculates the average variance to the mean or the average linear distance from the mean. Standard deviation is an example of measures of dispersion of the data that measure how dispersed the data items are in a set.

### 6.3.3.2 Inferential statistics

Inferential statistics are used to describe or draw conclusions on the characteristics of the main population, based on the results obtained from the sample of that particular population (Steinberg, 2008). A number of inferential statistics calculations exist; however, there are limitations on their usage. For example, some tests can only be conducted with a certain minimum sample size.

Two types of inferential statistics exist: parametric and non-parametric tests.

Parametric tests are tests that can be applied to data that have a certain value attached to each item, for example, Likert-scale responses. Examples of parametric tests include *t*-test and analysis of variance (ANOVA) (Tullis and Albert, 2008; Steinberg, 2008). Non-parametric tests are tests that can be applied to nominal and ordinal data. Nominal data are those data that have no degree or order, for example, CS and SOA environment, while ordinal data have some degree of order, for example, categories of users can be experts, intermediates or novices. An example of such a test is the chi-test (Tullis and Albert, 2008; Steinberg, 2008).

The *t*-test was selected for the analysis of the data in this evaluation. The reasons for selecting *t*-test were (Tullis and Albert, 2008; Steinberg, 2008):

- The sample size that can be used with a *t*-test has to be less than or equal to 30 items and the sample used for evaluation in this research contained 30 participants; however one participant's data was found to be corrupt during data analysis (Section 6.6); and
- The *t*-test can be used to measure the differences between two unequal samples. In this research the sample used was divided into two unequal sizes, 55% [n=16] of the participants interacted with the CS back-end system first and 45% [n=13] of the participants interacted with the SOA back-end system first (Section 6.5.2.2).

The derived statistical results were represented using various data-representation tools. The next section will discuss the tools that were used to represent the results.

### 6.3.4 Results representation

A number of different data-representation tools exist within Microsoft Excel 2007. The two most frequently used tools are graphs and tables.

- *Graphs*: Graphs play a very important role in presenting data, because graphs are usually easy to read (Tullis and Albert, 2008). Various types of graphs are available and in this research bar graphs, pie charts and line graphs were used to represent the evaluation findings. All bar graphs in this chapter were represented with a standard error bar; and
- *Tables*: Tables were used to present data that could not make sense when presented in a graph; otherwise a graph was used.

## 6.4 EVALUATION OBJECTIVES

The objectives for conducting the user testing and evaluation were:

- *To evaluate the completeness of customer profile components in SOA*: This was measured by the analysis of responses to the questionnaire that participants answered after they had finished interacting with the first system (Section 6.3.1.2, Appendix G);
- *To determine the differences between the SOA back-end system and CS back-end system*: This was measured by analysing the responses to the questionnaire the participants answered upon completion of interaction with the second system

(Section 6.3.1.3, Appendix H). Furthermore, eye tracking data were used to provide additional evaluation data for this objective (Section 6.3.2); and

- *To evaluate the SOA implementation:* This was done by evaluating the implemented SOA by using SOA design principles (Section 4.8).

The evaluation objectives were in line with fulfilling some of the research objectives, as outlined in Section 1.5. The next section will discuss the pilot study and the main end-user evaluation processes conducted, focusing on the objectives and using the instruments discussed in Section 6.3. Section 6.8 will discuss the SOA implementation evaluation.

## 6.5 EVALUATION

The project underwent a number of evaluations, as discussed in Section 6.1. This section discusses the evaluations that were conducted on the system, including the pilot study and the end-user evaluations. The SOA implementation evaluation will be covered in Section 6.8.

### 6.5.1 Pilot study evaluation

The pilot evaluation was conducted on the online e-commerce web application with the JWS-based customer profile back-end. A pilot study evaluation is defined as a small trial run in preparation for the main evaluation (Sharp, *et al.*, 2007). The evaluation was conducted before the main evaluation and its objectives were:

- To identify and fix bugs and other problems with the integrated SOA-based customer profiling system;
- To identify any usability issues that could affect the results of the main evaluation; and
- To ensure the system's readiness for the main end-user and the SOA implementation evaluation.

This section discusses how this evaluation was conducted by explaining the sample selection, the evaluation procedure that was followed, data collection and the analysis of the results. The analysis of the results provided valuable information regarding design and implementation changes made before the main evaluation.

### 6.5.1.1 Sampling

Purposive sampling was used in this evaluation phase. A purposive sample is a sample in which subjects are selected based on certain criteria that will have an influence on the results of the research (Research methods, 2009; Flick, 2009). Six Masters and two PhD students from the Department of Computing Sciences at NMMU were selected as participants for this evaluation. The motive for selecting these users was to find participants experienced in computing, Internet browsing and online shopping.

Table 6.1 shows the background experience and Table 6.2 shows the online shopping and profiling experience of these participants.

The two tables (Tables 6.1 and 6.2) show that the selected participants for this evaluation were expert computer and Internet users. The participants' average computer and Internet browsing experience amounted to more than six years. These participants were perceived by the researcher to be experts in testing the system's interface and functionalities prior to the main evaluation. The participants had experience in both online shopping and online registration, as is evident from Table 6.2.

Numerical values were assigned to the perceived Internet experience levels the participants reported; for example, 1 for expert, 2 for intermediate and 3 for novice. This was done to enable calculation of the standard deviation. This was found to be 0.35 (see Table 6.1).

Metric	Mean	Mode	Std. Dev
Computer Experience (Years)	6+	6+	0
Internet Experience (Years)	6+	6+	0
Internet Access Places	2.25	2	0.89
Hours on Internet / Day	3.17	6+	1.44
Perceived Internet Experience Level	Expert	Expert	0.35
n=8			

Table 6.1: Computer and Internet experience in years

Metric	Mean	Mode	Std. Dev
Number of Online Purchases	2	0	1.07
E-commerce Websites Visited This Year	1	1	0.53
Number of Online Registration Conducted	3.38	0	1.41
n=8			

Table 6.2: Online shopping and registration experience



### 6.5.1.2 Evaluation procedure

The users interacted with the end-user e-commerce web application using the desktop computers at their convenience, as the system was web-based. Several tasks (Appendix A) were set up to meet the objectives of the evaluation discussed in Section 6.5.1. Users performed the tasks and completed a background and post-task evaluation questionnaire (Appendix B).

The pilot study post-task evaluation questionnaire was divided into three sections, A, B and C. Sections A and B collected the participants' computer, Internet, online shopping and registration details similar to the details collected by the background questionnaire used in the main evaluation (Section 6.3.1.1). Section C of this questionnaire collected the participants' opinions on the overall reaction to the system, learnability, web-page design and navigation.

Additionally, section C of the questionnaire collected the general comments on the system. These were very useful in determining those areas of the system that needed attention before the main evaluation. Appendix B shows the questionnaire that was used.

Questionnaires were collected and the data were entered and analysed using Microsoft Excel 2007 (Section 6.3.3). The next section will discuss the results of this evaluation.

### 6.5.1.3 Pilot study results

Both qualitative and quantitative results were collected in this evaluation (Section 6.5.1.2, Appendix B). This section discusses these two types of results.

Figure 6.3 summarises the quantitative results obtained from the pilot study. The website learnability, web-page design and overall reaction to the system scored a mode of 4.00. The mean score for learnability was 3.88; the mean score for navigation was 3.13; the mean score for web-page design was 3.38; and the mean score for overall reaction to the system was 3.75.

The pilot study's quantitative results indicated that the participants had found problems with the systems navigation because of the lower rating obtained for both the mean and the mode. These were both below 3.5. This was confirmed with the qualitative results obtained from the general comments the participants had made. Design and implementation changes were made to the interface to correct this problem before the main evaluation was conducted (Section

6.5.1.4). The quantitative results also demonstrated the areas of the system that were almost perfect and ready for the main evaluation, for example, learnability and web-page design.

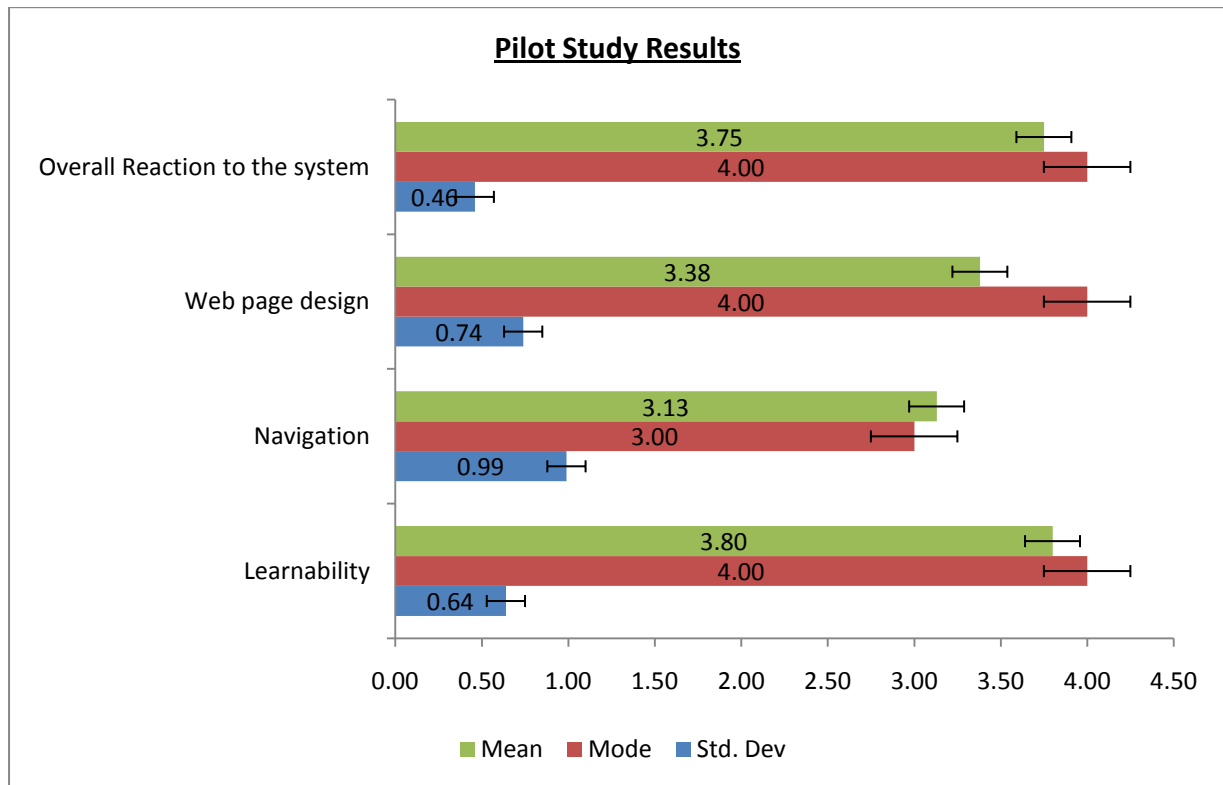


Figure 6.3: Pilot study evaluation system usability results [n=8]

Qualitative results were collected from the participants' comments that were made after interacting with the system. Some of the comments made were as follows:

- Make some interface pages independent of the master page, for example, the login page;
- Misspelt words were identified;
- There is a need for a clear link between the customer's profile and customer experience; and
- More or less product information placed far from the product pictures and the icons used were confusing.

The qualitative results provided important feedback on the system's functionality that led to some design and implementation changes (Section 6.5.1.4). This assisted in making the system more user-friendly and ready for the end-user evaluation. The main evaluation tasks and procedures (Appendices F and C) were set up taking into consideration the comments made in the pilot study. For example, an explanation of the more or less product information

buttons was given to the participant after it was noted in the pilot study that the buttons were confusing and were placed far from the product pictures.

#### 6.5.1.4 Design and implementation changes

In complying with the recommendations made by the experts who conducted the pilot study, the following changes were made to the system in order to get it ready for the final evaluation:

- An additional sub-master page was designed and implemented into the system. This additional master page catered for the pages that needed a different layout, for example, the login page; and
- More work was done on the product information display section of the product page (Figure 5.8). For example, the more or less product information buttons were made to be more obvious, and during the briefing session of the main evaluation emphasis was placed on the presence and purpose of these buttons.

Once the changes suggested by the pilot study evaluation results had been made, the system was ready for the main evaluation. The next section will discuss the main study evaluation.

### 6.5.2 Main study evaluation

The main end-user evaluation involved users interacting with two systems by performing two sets of similar tasks (Appendix F) on both the SOA-based and CS-based systems. As discussed in Chapter 5, the two end-user interfaces had the same interface, but were running on different customer profile back-ends. The tasks were set up in such a way that users could fully explore and interact with the customer profiling components of the systems, as well as the end-user e-commerce web-application functionalities.

This section discusses the main end-user evaluation procedure by explaining the sample selection, evaluation procedure, and eye tracking and data collection methods. This evaluation was conducted after the relevant research ethics clearance was granted by the Faculty RTI committee.

#### 6.5.2.1 Sampling

Tullis and Albert (2008) stated that a representative sample has to be used in the evaluation of computer systems. A representative sample is a sample which has characteristics similar to

the actual targeted population of the study (Steinberg, 2008). A convenience sample, in which available and volunteer participants are recruited (Tullis and Albert, 2008), was used in this evaluation. E-mails were sent to other Departments at NMMU, for example, Business Management, asking volunteers to participate in this evaluation. Furthermore, word-of-mouth was passed to students during lectures and at many other opportunities to recruit participants for the evaluation.

The targeted participants were postgraduate students at NMMU, lecturers and undergraduate students who had an average of four or more years of computer use and Internet experience. A total of 30 participants conducted the evaluation, as advised by Mr Venter of the USC at NMMU; however, one participant's data were taken out of the analysis because the data were found to be corrupt. This section discusses the sample in terms of the biographical details and the background experience, including computer and Internet-browsing experience, and online shopping and registration experience.

Figure 6.4 shows the age range of the sample. The figure indicates that 83% [n=24] of the participants were in the age range of 21-30. No participant below the age of 21 was included in the sample, and this was done according to the requirements of the research ethics committee of the NMMU. Figure 6.5 shows the composition of the sample in terms of position and year of study.

The figure indicates that the sample consisted of 39% [n=11] Masters' students and 25% [n=7] Honours' students from various departments at NMMU. One lecturer, two DTech students and two PhD students were also part of the sample. Five undergraduate students with a minimum of three years of computer and Internet experience were also included in the sample, as can be seen in Figure 6.5.

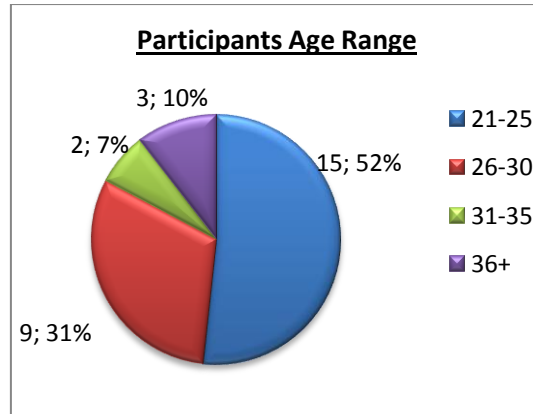


Figure 6.4: Participants' age range [n=29]

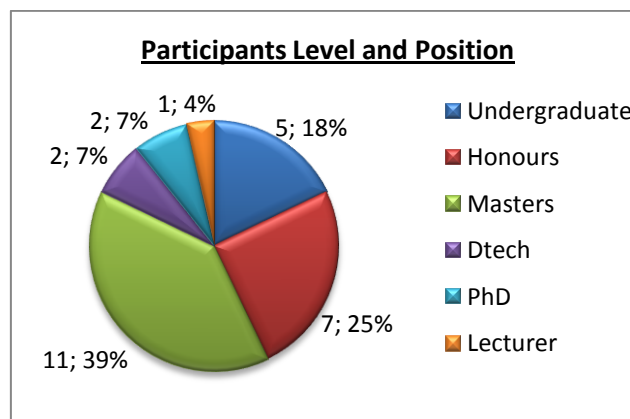


Figure 6.5: Participants' position [n=29]

Seventy-six percent [n=22] of the participants were male and 24% [n=7] were female. Fifty-nine percent [n=17] of the participants were black, while 35% [n=10] were white, and Indians and coloureds were 3% [n=1] each. Figure 6.6 shows a summary of the participants' ethnicity and Figure 6.7 shows the gender of the participants.

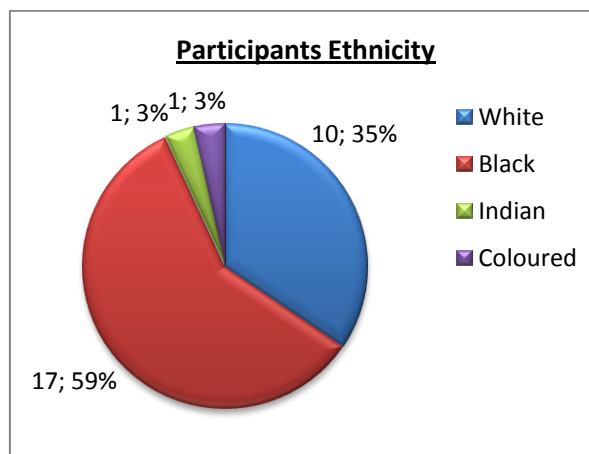


Figure 6.6: Participants' ethnicity

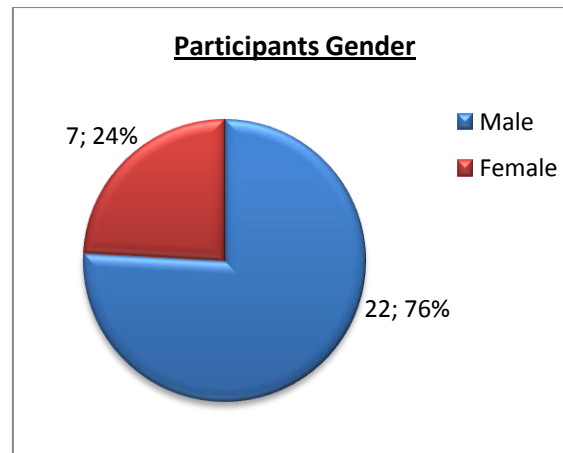


Figure 6.7: Participants' gender

Table 6.3 shows the computer and Internet experience of the participants. The table shows that the mean computer experience for the sample was more than six years with a standard deviation of 0; the mean Internet experience was more than six years with a standard deviation of 0.40, and the mean Internet access places amounted to two with a standard deviation of 0.74. The mode for all the parameters in the table was the same as the means of the parameters. The results in Table 6.3 indicate that the sample consisted of experts in computing and Internet browsing. Although a convenience sample was used, computer and Internet experience were important criteria for the participants to successfully complete the tasks and the post-task questionnaires.

As was discussed in Section 6.5.1.1, a numerical value was assigned to the participants self-reported Internet experience and the value was used to calculate the standard deviation that was found to be 0.55 (Table 6.3).

Metric	Mean	Mode	Std Dev
Computer Experience (Years)	6+	6+	0.00
Internet Experience (Years)	6+	6+	0.40
Internet Access Places	2	2	0.74
Self-rated Internet Experience	Expert	Expert	0.55
n=29			

Table 6.3: Computer and Internet experience in years

Ninety-seven percent [n=28] of the participants had browsed through an online e-commerce website this year and conducted online registration, and 34% [n=10] of the participants had conducted online shopping before. Appendix I outlines the detailed data and results.

### 6.5.2.2 Evaluation procedure

The evaluation was conducted in the NMMU Department of Computing Sciences Usability Laboratory and the aims for conducting the evaluation in a controlled environment have been outlined in Section 6.2. Each participant conducted the evaluation, one at a time, in the participant room of the Usability Laboratory (Figure 6.1) and each evaluation lasted an average of 25 minutes.

The participants were first welcomed into the laboratory with the aim that the participant should feel relaxed. Participants were briefed on the background to the project; a description of the intended evaluation and the objectives of the evaluation was also given. An overview of the equipment to be used, including the eye tracker, was given, and a consent form (Appendix D) was handed to the participant. The consent form stated the objectives of the evaluation, what was required of the participant and an explanation of the participant's rights.

In order to participate in the study, the participant had to sign the consent form as an agreement with the contents, and then proceeded to complete a background questionnaire (Section 6.3.1, Appendix E).

Upon completing the biographical details questionnaire, an overview of the tasks to be performed was explained to the participant before calibration of the eye tracking equipment (Section 6.3.2) was done in preparation for the task completion. Each participant performed the same set of tasks and the tasks were read out by the observer, who was sitting in the observer room, to the participants as they were conducting the tasks on the system (Appendix F). The reason for reading out the tasks to the participants was to avoid distracting eye tracker recording, as users may have been required to look at the task sheet during evaluation (Pretorius, *et al.*, 2005).

The participants were not told which system was running in the background, as this was part of the evaluation. The systems were swapped with each subsequent participant, and this resulted in 55% [n=16] of the participants interacting with the CS back-end system first and 45% [n=13] of the participants interacting with the SOA back-end system first. At the end of each task session, the participants were asked to complete a post-task evaluation questionnaire, as discussed in Section 6.3.1 (Appendices G and H).

### 6.5.2.3 Eye tracking and questionnaire data collection

As discussed in Section 6.3.2, eye tracking was used in the evaluation of this research to capture additional evaluation metrics that could be used to validate the responses that participants gave during the completion of the paper-based post-task evaluation questionnaires. All the participants' eye tracking recordings were merged into one project in the Tobii Studio eye tracking software (Section 6.3.2). The project had two tests, one test for SOA and the other for CS. Each participant's eye tracking recordings were collected into either a CS test or an SOA test, depending on the current system with which the participant was interacting.

When participants completed the evaluation tasks, the eye tracker recording was stopped and participants were asked to complete a post-task evaluation questionnaire. This implied that eye tracking data were being collected, whilst the participant was conducting the evaluation tasks. The eye tracking recordings were stored on the evaluation computer and a back-up was made at the end of every evaluation day and stored on a different computer and DVD.

Questionnaires were collected immediately after the debriefing session (Appendix C), and the observer recorded the order of the type of the system with which the participant had interacted and a unique participant ID was recorded at the top of each of the three questionnaires at the end of each evaluation session. This was done to match the results of the eye tracking, the type of system used and the participants' background details collected by the background questionnaire. Appendix C gives an outline of the step-by-step evaluation procedure that was modified from the eye tracking usability evaluation procedure, as discussed by Pretorius, *et al.* (2005).

The background and post-task evaluation questionnaire data were entered into Microsoft Excel 2007 worksheets and a special code book was created for each response in the questionnaires discussed in Section 6.3.1. The aims of creating the code book were:

- To ensure data standard and removing data redundancies during data entry, for example, the response of yes should have the same value wherever it appears in the data; and
- To improve understanding between the researcher and the statistical consultant, Mr Venter, during analysis.



The questionnaires were collected and stored safely awaiting data analysis. The following section discusses how the data analysis was conducted.

## 6.6 DATA ANALYSIS

Once all the 30 participants' data were collected, Mr Venter of the USC was consulted for assistance in the statistical and data analyses. The data collected by questionnaires and eye-tracking (Sections 6.3.1 and 6.3.2) were entered into the Microsoft Excel 2007 sheets (Section 6.5.2.3). The data analysis process started with cleaning the data, in which inconsistencies and errors were identified, fixed or removed. This process resulted in one participant's data being removed from the dataset because they were found to be corrupt.

A Microsoft Excel data analysis application developed by Mr Venter was used to analyse the *clean* data. The application had formulas for the required descriptive and inferential statistical tests (Section 6.3.3). Descriptive statistics formulas in the application were made from normal Microsoft Excel functions, such as, average, standard deviation and countif.

The inferential statistics formulas were implemented using normal formulas. However, they worked only when the data or the mean scores being tested showed statistical significance. That is, when the p-value was found to be within the acceptable range, a specific formula was used to calculate the practical significance of the results, for example, Cohen's d (Section 6.7.2). Once the correct data were entered into the application, the results were obtained. The next section will discuss the main evaluation result analysis.

## 6.7 ANALYSIS OF RESULTS

The evaluation had several objectives (Section 6.4). Different data collection instruments, for example, questionnaires and eye tracking recoding, provided data which could be used to achieve the evaluation objectives. This section discusses the data analysis results for the particular research objectives, as discussed in Section 6.4. The section will report on the two statistical analyses that were conducted on the data by means of descriptive and inferential statistics (Section 6.3.3).

### 6.7.1 Descriptive statistical results

This section discusses the descriptive statistical results of the data collected for measuring the objectives of the evaluation (Section 6.4).

### 6.7.1.1 Completeness of customer profile components in SOA

The goal here was to determine whether the identified customer profile components, implemented as individual services (Section 4.3.1), were sufficient and useful for performing the customer profiling functionality. Figure 6.6 shows the results obtained when participants were asked to rate various customer profiling components. Section D of the post-task evaluation first system questionnaire was used to collect this information from the participants (Section 6.3.1.2, Appendix G).

Figure 6.8 indicates the overall satisfaction with the customer profiling components' functionality that was calculated by taking the average rating for each semantic differential response. The figure shows that the create customer profile component had a mean score of 4.38 that was equal to the mode score with a standard deviation of 0.44. The view-profile component had a mean score of 4.22, a mode score of 4.67 and standard deviation of 0.50. The update profile component had a mean score of 4.06, a mode score of 4.67 and a standard deviation of 0.67. Deleted profile component scores had a mean of 4.12, a mode of 5.00 and a standard deviation of 0.96.

The overall profiling component's satisfaction was calculated by taking the average of all the scores of the individual components. Results show that the overall mean was 4.19, the overall mode was 4.67, and the overall standard deviation was 0.47 (Figure 6.8).

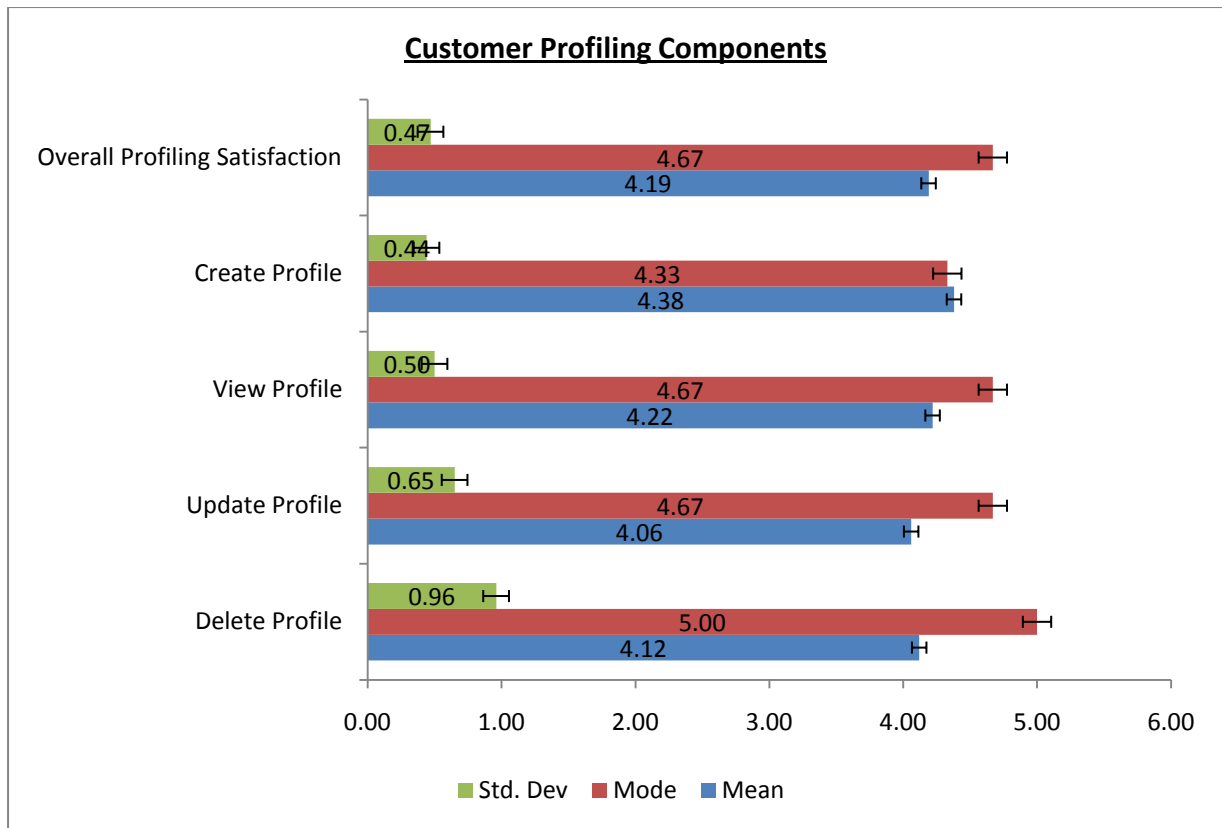


Figure 6.8: Customer profile components [n=29]

Section E of the post-task evaluation questionnaire first system (Section 6.3.1.2, Appendix G) used an ASQ to collect the overall post-task rating of the system (Tullis and Albert, 2008). Figure 6.9 indicates the results obtained from the analysis of these data. The figure shows that the mean score for ease of completing tasks was 3.98, the mode was 4.67 and the standard deviation was 1.15. Task completion times scored as follows: the mean was 3.67; the mode was also 3.67, while the standard deviation was 1.08. The systems functionality's mean was rated as 4.04, the mode was 5.00, and the standard deviation was 1.26. The calculated overall scores were as follows: the mean was 4.52, the mode was 4.67, and the standard deviation was 0.49.

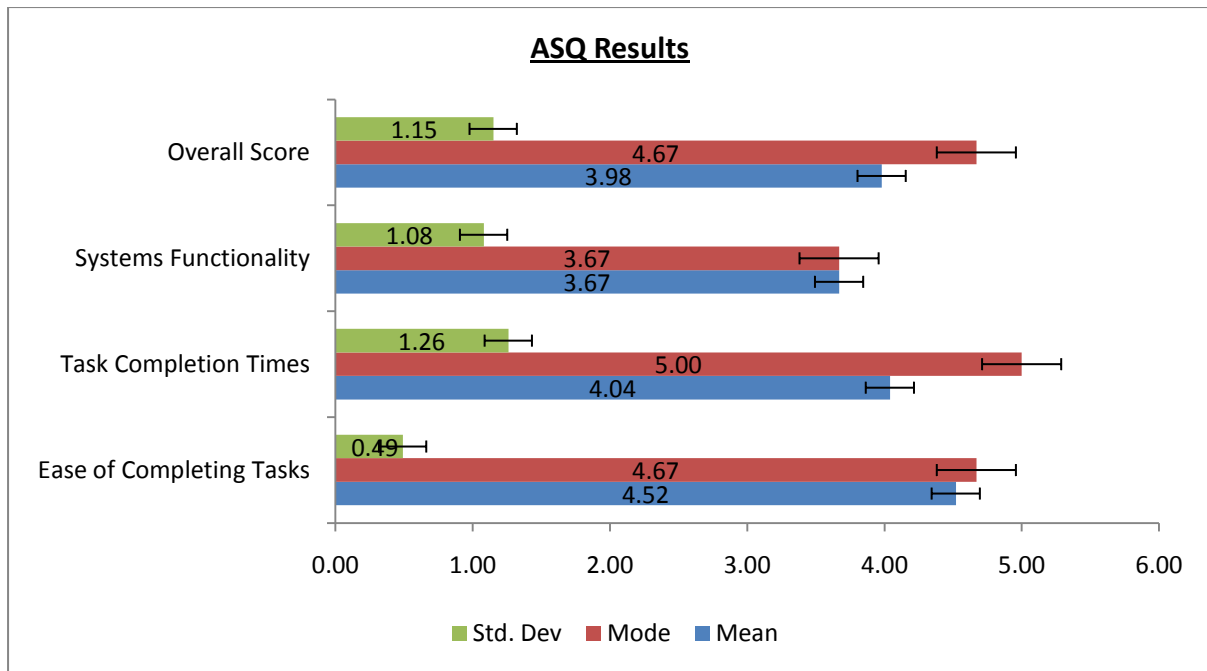


Figure 6.9: After scenario questionnaire results [n=29]

The results for the customer profiling components in SOA indicated that the participants found the customer profiling components implemented as being loosely coupled services (Section 4.3.1) useful and complete for conducting the customer profiling functionalities (Figures 6.8 and 6.9). Despite the participants rating the task completion to be tedious and time consuming, ease of completing tasks was rated high (more easily completed).

This meant that although the tasks were rated as having taken longer to complete than the participants expected, they were quite easy to accomplish (Figure 6.9). The task completion times can also be confirmed to be time consuming by looking at the average time it took for one evaluation to be completed (Section 6.5.2.2).

The next section will examine the results of the data collected to evaluate the comparison between the SOA and CS systems.

#### 6.7.1.2 Comparison between SOA and CS back-end systems

The aim was to determine whether the participants had observed any differences between the two systems used in this evaluation. Post-task evaluation questionnaire from the second system and eye tracking data were used to collect data to evaluate this objective (Sections 6.3.1.3 and 6.3.2, Appendix H). This section discusses the results obtained from this analysis.

Figure 6.10 shows the results of the comparison in terms of a system's response and task completion. The comparison shown in Figure 6.10 was between the mean, mode and standard deviation. The overall score indicates that the mean was 4.52; the same in both systems, the mode was 5.00 in SOA and 4.80 in CS, while the standard deviation was 0.60 in SOA and 0.58 in CS. The main differences can be seen on the ease-of-task completion and task-completion times, in which different values for the mode and the mean, as well as for the standard deviation and the mean were observed respectively.

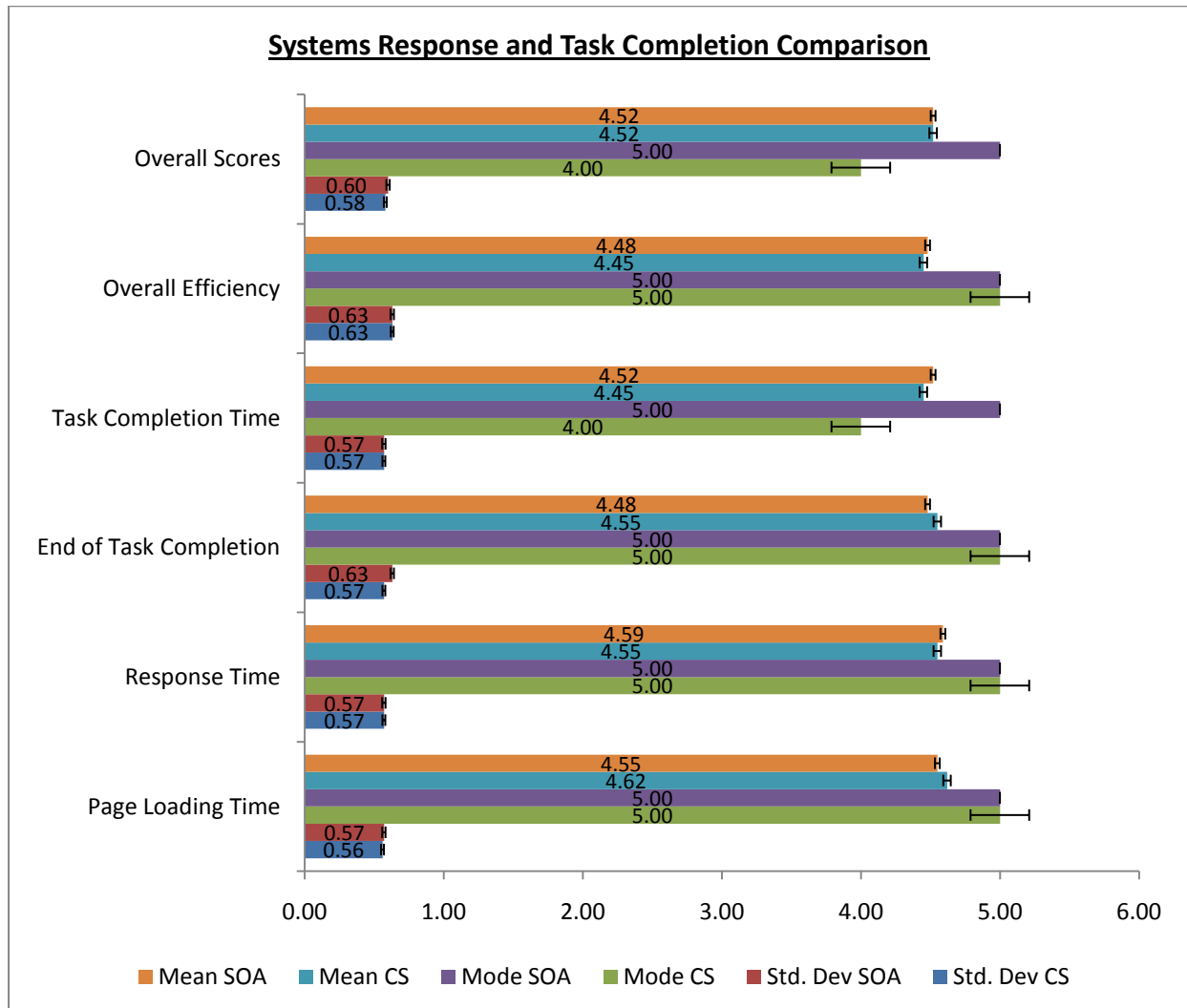


Figure 6.10: Systems' response and task completion comparison [n=29]

Figure 6.11 extends the information shown in Figure 6.10 by indicating the calculated differences between the mean, mode and standard deviation of the ratings for the two systems. The values were obtained by subtracting the CS rating from the SOA rating. The mode was removed from Figure 6.11 because it had a difference of 0 as can be seen from Figure 6.10.

The line graph (Figure 6.11) shows the differences between the two systems. A positive difference indicates that SOA was rated higher for that particular metric and a negative difference indicates that CS was rated higher for that particular measure. Figure 6.11 shows that the SOA standard deviation was found to be rated slightly higher in page loading times and ease of task completion (0.01 and 0.06 respectively), while the mean for page loading time and ease-of-task completion was rated higher in CS, with a difference of -0.07 for both metrics. The standard deviation difference for response time, task completion time and overall efficiency was found to be 0.

The results in Figures 6.10 and 6.11 shows that the participants observed some small differences in the metrics that were used to measure the systems' response to the two systems. This is more evident in Figure 6.11 which shows the different values to be within the range of -0.07 to 0.07. The statistical significance of these observed differences will be discussed in Section 6.7.2 (Inferential statistics).

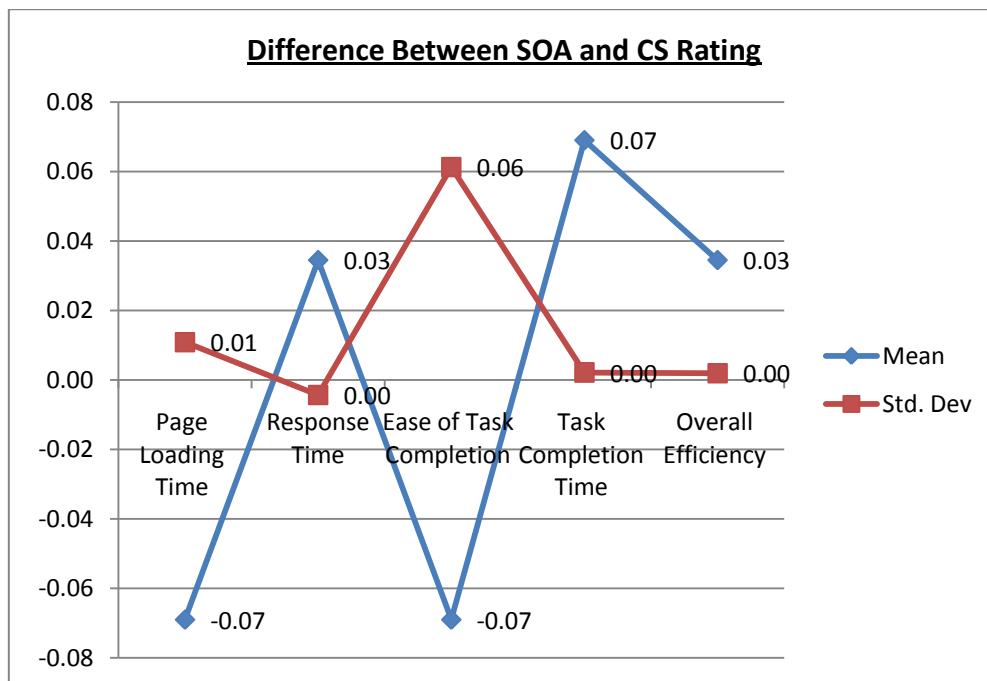


Figure 6.11: Difference between SOA and CS rating [n=29]

An overall assessment of the SOA and CS systems was also conducted, in which participants answered questions with semantic differentials with regards to the general feeling about the two systems. Figure 6.12 shows the results obtained from this analysis. The figure indicates that the overall standard deviation was 0.78 in SOA and 0.73 in CS; the mode was 4.20, the same in both systems; and the mean was 4.28 in CS and 4.27 in SOA.

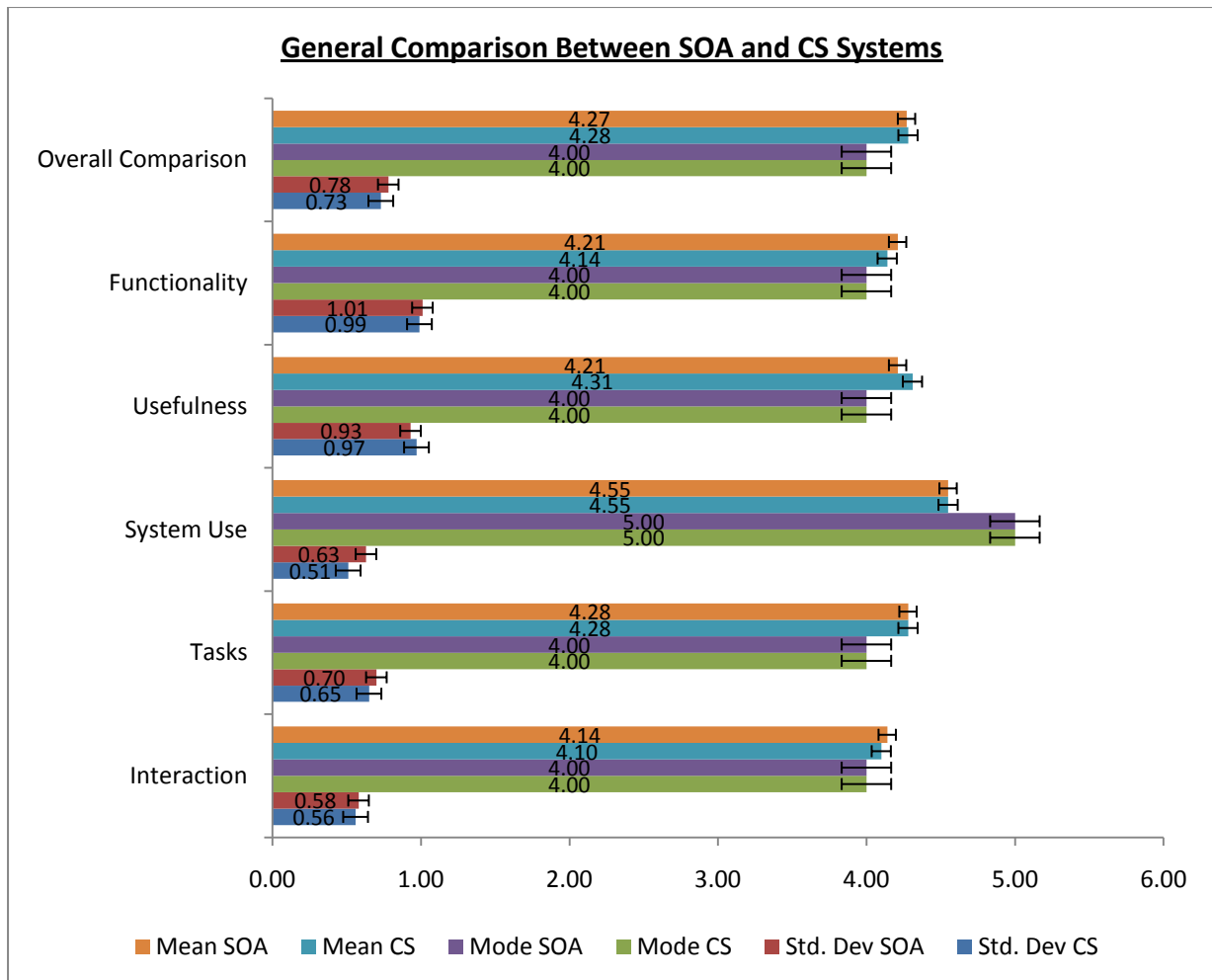


Figure 6.12: General comparison between SOA and CS systems [n=29]

Figure 6.13 shows the summary of the overall differences found between the SOA and CS systems. The values were found by subtracting the CS rating from the SOA rating. A positive difference means SOA system was rated higher than the CS system and vice versa. The figure indicates that the standard deviation for SOA was higher than for CS, while the mean for easiness and usefulness was rated higher in CS.

The results in this section indicated that the participants rated the systems slightly differently. For example, Figure 6.12 shows that the overall comparison resulted in the SOA having a slightly smaller mean value (4.27) as compared with CS systems mean overall comparison (4.28). The mode for comparison was the same for all the metrics in Figure 6.12, which means that for most responses the participants reported no difference. In general it can be concluded that the participants' responses indicated some differences between the two systems. However, these responses were perceived by the researcher to be insignificant.

The results confirm that SOA as a paradigm has nothing to do with the end-product as discussed in Section 3.2. In SOA different systems can inter-operate in the background and present the same interface to the end-user (Jammes, *et al.*, 2005). Results obtained from the end-users in this evaluation indicated that despite the two systems interacting with different customer profile back-ends, the participants found the output presented to be similar.

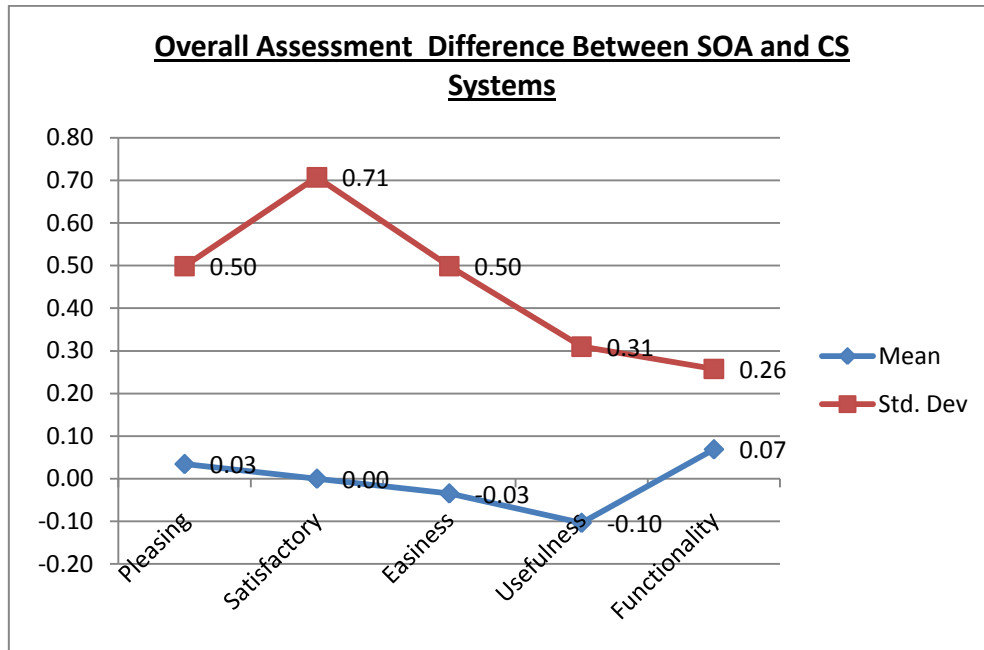


Figure 6.13: Overall assessment difference between SOA and CS systems [n=29]

### 6.7.1.3 Eye tracking results

Eye tracking was used to collect additional evaluation information that was not possible to collect by using questionnaires (Section 6.3.2). Whilst using the eye tracker, the systems' page loading times were recorded to establish which systems' pages were loading faster than the other. The target was to record the times for pages that were interacting with the back-end customer profiling components in both systems (Section 5.4, Table 5.1). These results will be compared to what the participants said during completion of the post-task evaluation questionnaire second system.

The differences in this section were calculated by subtracting the CS value from the SOA value. This means that a positive difference indicates that SOA system recorded a longer time to load the page than the CS system and vice versa. The comparison was done only on mean and standard deviation because the mode could not be found in the dataset. This was because



each value was different from the other by a matter of milliseconds and for precision purposes every millisecond was accounted for.

Despite a number of pages interacting with the customer profile back-end, the data of only two page loading times were analysed. The other pages' loading times were found to have inconsistent data and were removed during the data cleaning (Section 6.6). The pages' loading times that were used in the analysis were the registration and create-profile page. These pages interacted with the register customer and add-customer services in SOA (Section 4.3.1) and register customer and add-customer classes in CS (Section 4.6).

Figure 6.14 shows the results of the comparison between these two page loading times. The result indicates that the CS system was quicker, that is, it completed the background processing more quickly than the SOA system. Figure 6.14 shows that both the mean and the standard deviation for SOA were slightly higher than the mean and standard deviation for CS. This means that the SOA system took more time to complete the background interaction with the customer profiling web services than did the CS system.

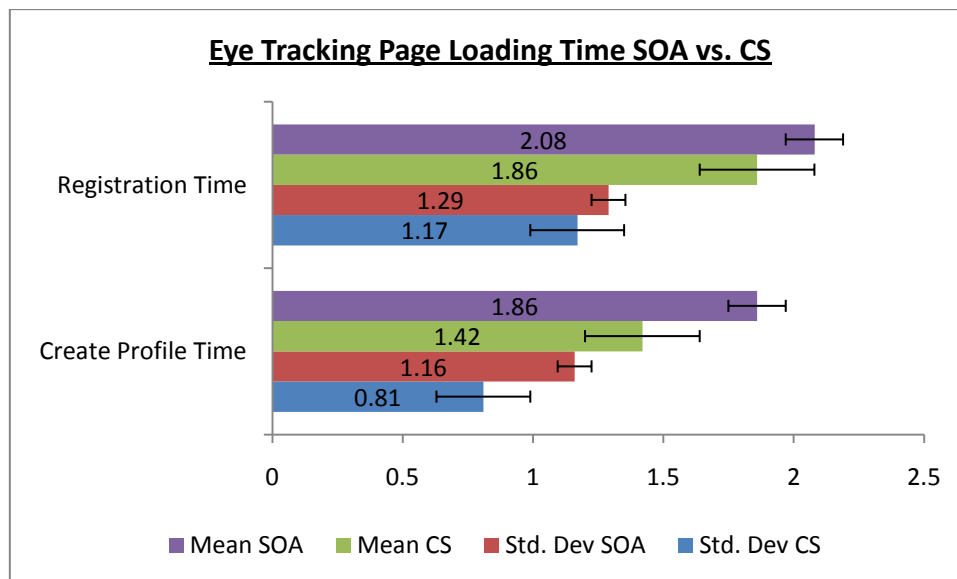


Figure 6.14: Eye tracking page loading times SOA vs. CS [n=29]

Figure 6.15 extends the description in Figure 6.14 by showing a line graph for the mean and standard deviation. As discussed, the difference was calculated by subtracting the CS value from the SOA value, and in both instances, a positive value was observed. This indicates that SOA was a bit slower (took longer) in completing the background processing than CS.

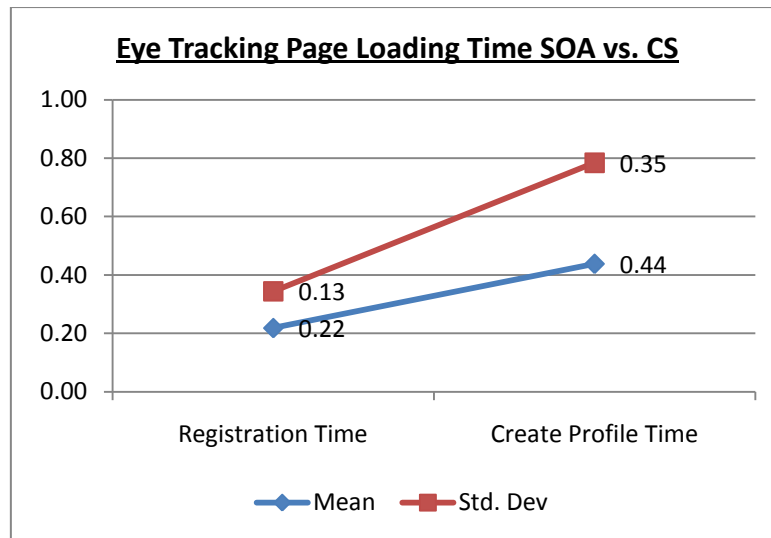


Figure 6.15: Eye tracking page loading times SOA vs. CS line graph [n=29]

Comparison between the page loading times for the participants that interacted with the SOA system first and the CS second, and then vice-versa was also conducted. Figure 6.16 shows the page loading times for the participants that interacted with the SOA system first and the CS system second. Thirteen participants interacted with the SOA system first, and then proceeded to interact with the CS system secondly.

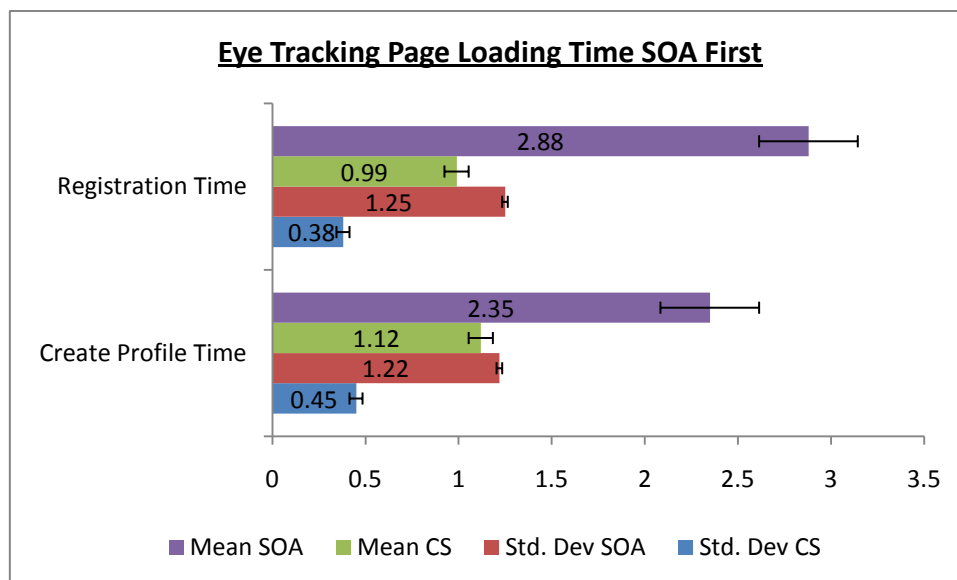


Figure 6.16: Eye tracking page loading time SOA first [n=13]

Figure 6.17 shows the difference in page loading times between the SOA and CS for the participants that interacted with the SOA system first. The same calculations for finding the difference discussed were used and Figure 6.17 shows only positive differences which

indicate that the SOA system took longer to load pages than did the CS system, that is, SOA took more time to complete the background processing than did the CS system.

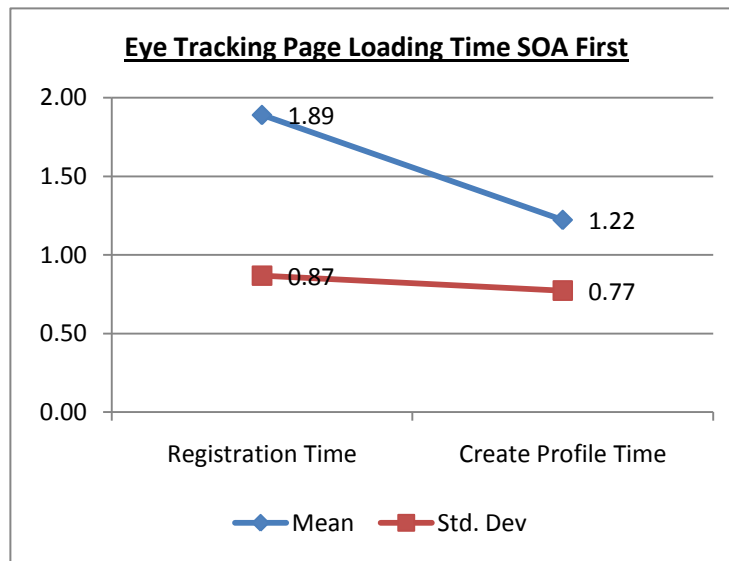


Figure 6.17: Difference in eye tracking page loading time SOA first [n=13]

The results of the comparison of the page loading times between the participants who interacted with the CS system first and went on to interact with the SOA system, further indicate that the SOA system took longer to load pages than did the CS system. Figure 6.18 shows the values of the mean and standard deviation for the two systems.

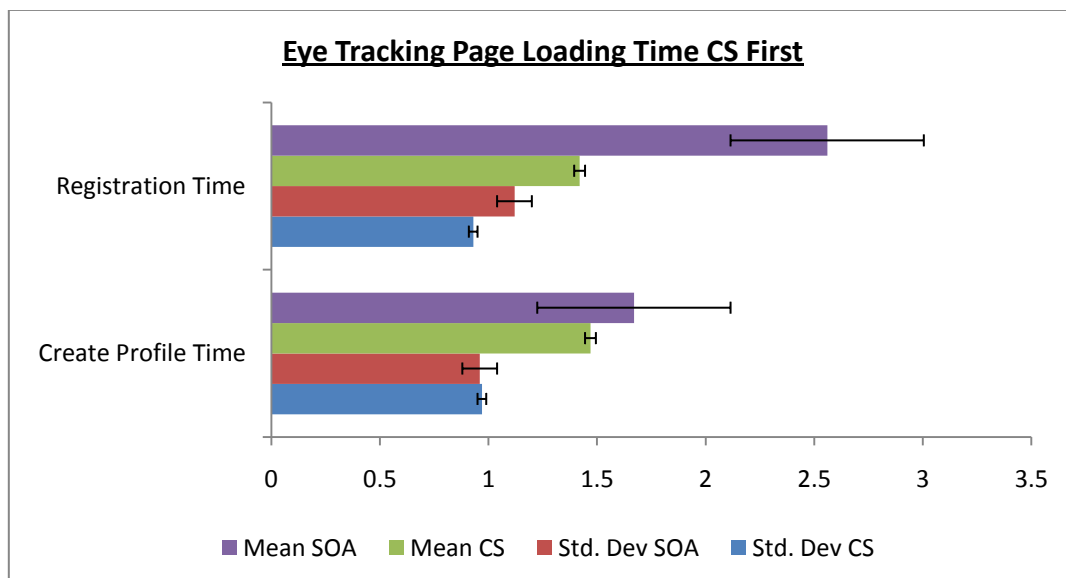


Figure 6.18: Eye tracking page loading times CS first [n=16]

Figure 6.19 shows a line graph that further explains the results displayed in Figure 6.18. The figure shows the difference between CS loading times and SOA loading times for the participants that interacted with the CS system first. After the calculations, positive differences were found, meaning that the SOA system took longer than the CS system to complete the back-end customer profile interactions.

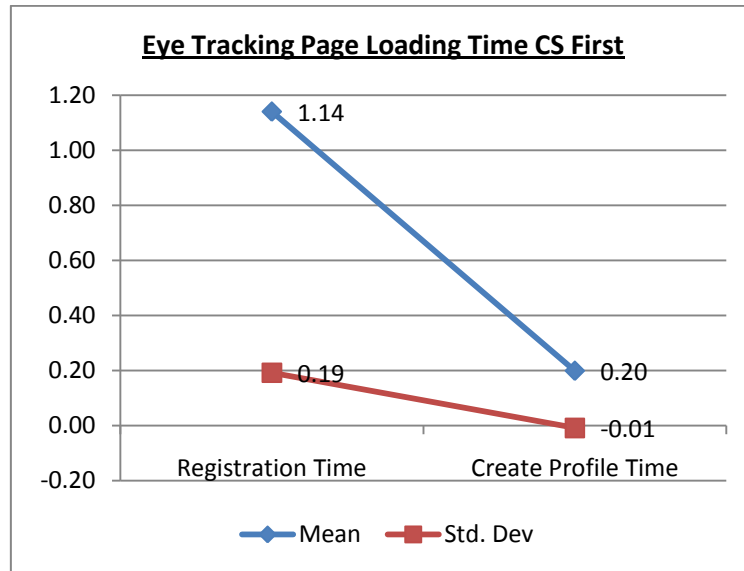


Figure 6.19: Difference in eye tracking page loading times CS first [n=16]

In conclusion, it can be said that the SOA system took longer to complete the background interactions with the customer profile web services. This trend has been observed, despite the type of system the participants interacted with the first time. Comparing with the participants questionnaire responses it can be observed that the eye tracking information is very clear and consistent; it was clear that the SOA system took longer to load pages than did the CS system, while in the questionnaire responses the differences seem not to be very clear (Figure 6.12).

This differences observed between page loading times of the two systems can be due to the fact that in an SOA environment the parameters were passed as standard messages using SOAP and the services were implemented as loosely coupled. Therefore to find an appropriate service and the passing of parameters took more time. While in CS the parameters were just sent to the called class to perform the execution and this resulted in CS background processes completed faster and consequently loading the pages quicker than SOA.

Further to page loading time comparison, a number of participants' eye tracking heat maps were compared. Eye tracking heat maps show how much time the user spent looking at a particular item of the system's interface by using different colour codes (Section 6.3.2). Heat maps are calculated from eye-gaze statistics of where on the screen the participants focused most of the time, but the heat maps are presented qualitatively and therefore give a quick overview or summary of the participants' eye-gaze patterns. Heat maps are easier to read and interpret than other eye tracking reports, for example, gaze plots (Pernice and Nielsen, 2009).

The first heat maps' comparison conducted was between the participants who interacted with the CS system first and the SOA system second. Figure 6.20 shows the heat maps obtained from participants after interacting with the CS system first and Figure 6.21 shows the heat maps of the same sample after interacting with the SOA system. During the first interaction (Figure 6.20), the participants spent most of their time looking at the product-display section of the product page (Figure 5.8). Figure 6.20 also shows that during the interaction with the CS system, the participants spent more time reading the product information on the right-hand side of the product page than during the second interaction with the SOA system (Figure 6.21).



Figure 6.20: Heat map for participants CS first [n=16]



Figure 6.21: Heat map for participants SOA second [n=16]

The second heat maps' comparison conducted was between the participants who interacted with the SOA system first and the CS system second. Figure 6.22 and 6.23 shows the heat maps obtained. Figure 6.22 shows that the pattern of the heat maps is very similar to the one observed in the participants who interacted with the CS system first and the SOA system second (Figures 6.20 and 6.21). In Figure 6.22 the participants also spent time looking at the product display section of the product page (Figure 5.8) in both systems; however, they spent more time looking at this section in the second interaction with the CS system (Figure 6.23).



Figure 6.22: Heat map for participants SOA first [n=13]



Figure 6.23: Heat map for participants CS second [n=13]

In summary, the observed heat maps provide less significant differences when compared with the page loading times. The heat maps confirm the participants' responses given during the completion of the post-task questionnaire, the second system (Section 6.3.1.3). Participants commented that they did not notice much difference between the two systems, as summarised in Figure 6.10 and Figure 6.12. Appendix I provides a summarised heat map of the two systems, SOA vs. CS.

In addition, the heat maps show that the system's learnability was good, as was found in the pilot study (Figure 6.3), ASQ results (Figure 6.9) and the general comments made about the system (Qualitative results, Section 6.7.3). This section has outlined the descriptive statistical results of the evaluation, including the differences between the two systems both from the eye tracking and the participants' perspective. The next section will discuss the inferential statistical analysis that was conducted to explore the statistical and practical significances of the results.

### 6.7.2 Inferential statistical results

A *t*-test was used in this research (Section 6.3.3.2) to find the statistical significance of the difference between the two systems that was observed in the descriptive statistics discussed in Sections 6.7.1.2 and 6.7.1.3. Statistical significance shows whether the results observed reflect the true representation of the targeted population or not (Steinberg, 2008). P-values are used to determine the statistical significance of the results. Acceptable p-values are calculated based on the chosen alpha values (that is the confidence level accepted, or how much error is acceptable) (Tullis and Albert, 2008; Steinberg, 2008). For example, for 95% confidence interval, alpha is chosen as 0.05, which means that the acceptable p-values in order to reject the null hypothesis has to be less than 0.05 (Tullis and Albert, 2008).

In addition, Cohen's *d* was used to find the practical significance (effect size) of the differences observed (Venter, 2009; Steinberg, 2008). The practical significance of the differences further explain whether the observed differences can have an impact on the main target population or not (Steinberg, 2008). There are generally three guideline measures for Cohen's *d*, namely:

- *Smaller effect*: Cohen's *d* values below 0.5;
- *Medium effect*: Cohen's *d* values between 0.5 to 0.8; and
- *Large effect*: Cohen's *d* values larger than 0.8.



A paired *t*-test that measures the differences between two designs utilising the same sample was used (Tullis and Albert, 2008). In this research, the same sample was used to test two different systems, CS and SOA.

A null hypothesis is usually proposed to test the statistical and practical significance of the results when using a *t*-test and other inferential statistics tests. The null hypothesis that was proposed and tested in this research was:

*H<sub>0</sub>: There is no difference between the two systems, CS and SOA.*

A *t*-test analysis was conducted on the data collected, that is, the eye tracking data and the participants' perceived differences between the two systems. Table 6.3 shows the results of the *t*-test when conducted on the participants' responses to the perceived differences between the two systems (Section 6.7.1.2).

Table 6.4 shows the only p-value that can be used to reject the null hypothesis is the value for the SOA system response, otherwise the other p-value supports the hypothesis. The p-value for the SOA system's response was 0.05. This is the minimum and looking at the Cohen's *d* value it can be seen that the value is within the range of medium effect size.

	Systems' Response		Overall Assessment	
	SOA	CS	SOA	CS
t-Statistic	2.08	-0.75	0.68	-0.56
d.f.	27	27	27	27
p-value	0.05*	0.46	0.50	0.58
<b>Practical</b>				
Cohen's <i>d</i>	0.78**	n.a.	n.a.	n.a.
* used p-value				
**used Cohen's <i>d</i> value				

Table 6.4: *t*-test results for participants' perceived differences between the SOA and CS systems

These results show that the difference was only observed by the participants in the SOA system, that is, the participants noted some differences with the SOA system. From the p-value and Cohen's *d* value it can be concluded that these values were statistically significant with a medium effect size.

Table 6.5 shows the inferential statistics obtained on the eye tracking page loading times of the two systems. The table indicates that p-values for the registration times for both the SOA and the CS systems were less than 0.0005; consequently the hypothesis was rejected. The SOA create-profile time p-value indicates that it rejected the hypothesis, while the CS system create-profile time supported the hypothesis.

Cohen's d values for all the parameters that rejected the hypothesis were in the large effect range which infers a very strong and practical significance.

	Registration Time		Create Profile Time	
	SOA	CS	SOA	CS
t-Statistic	-4.59	6.18	-2.58	2.01
d.f.	25	26	25	26
p-value	<.0005*	<.0005*	.016*	.055
Practical				
Cohen's d	1.78**	2.36**	1.00**	n.a.
* used p-value				
**used Cohen's d value				

Table 6.5: t-test results for eye tracking page loading times of the two systems SOA and CS systems

These eye tracking *t*-test results indicate that the null hypothesis proposed was rejected, and therefore it can be concluded that:

*There were differences between the two systems, CS and SOA.*

From these results, it can be concluded that the practical and statistical significance of the differences observed between the two systems, SOA and CS, were more apparent when eye-tracking was used (Table 6.5). This has been demonstrated by a large effect size and strong rejection of the hypothesis shown by the large Cohen's d value (effect size) and very small p-values (<0.0005). The participants' responses to the differences between the two systems resulted in large p-values (0.05), that is, values closer or equal to 0.05 and the effect size was medium (Table 6.4).

The statistical and practical significance found from the eye tracking data shows that in reality, performance differences do exist between the two systems, CS and SOA. CS system was found to be responding quicker than the SOA. However, to the participants it was not easy to notice the differences and they perceived the two systems to behave the same. This

has been confirmed with the inferential statistics discussed in this section. The next section discusses the qualitative results obtained from the evaluation.

### 6.7.3 Qualitative results

As discussed in Section 6.3.2, the questionnaires that were used in this evaluation collected both qualitative and quantitative data. The questionnaire for the quantitative data was collected by the weighted response questions using a Likert and semantic differential scale, while the qualitative data were collected by sections in the questionnaire that collected both open-ended answers and general comments. This section discusses the qualitative results obtained by examining the open-ended questions' responses, as well as the positive and negative comments that participants made during the evaluation.

#### 6.7.3.1 Positive comments

The positive comments that were received from the participants during the evaluation were divided into two groups: the first group contained comments collected after participants interacted with the first system, and the second group contained the comments collected after the participants had interacted with the second system. The open-ended questions' responses and general comments were used to seek further justification for the participants' responses.

For example, if a particular participant rated delete profile as very useful and in the general comments section made a good comment about delete profile, it was taken as a further justification for the high rating given on delete profile.

The positive comments regarding the goal of the research obtained from the first group were:

- The customer profile ratings were correct;
- The need to include a password and profile recovery as a customer profile component;
- Give customers control to edit their profile from a view profile;
- Delete profile is relevant, however, it had to be somewhere other than on the main menu; and
- The interface was easy and user-friendly.

The positive comments received from the second system post-task evaluation questionnaire were:

- The second system was faster than the first system;
- The second system asked less questions than the first system;
- The view profile was helpful;
- The different levels of the product information display were time saving;
- It was pleasant to see the full-time display of current basket items and totals; and
- There was no difference between the systems.

Upon analysis of the comments obtained from the second system, it was concluded that the system was easy to learn. Learnability was not part of the evaluation objectives (Section 6.4). However, by analysing the responses to the open-ended questions and general comments received after the participants had interacted with the second system, positive feedback on the learnability of the interface was observed.

Furthermore, the qualitative data confirmed that the participants found the systems' responses to be almost the same as shown in Section 6.7.2. Comments such as, "there was no difference between the systems" were received from the participants, even though the systems were being switched with each participant (Section 6.5.2.2).

#### 6.7.3.2 Negative comments

The negative comments that were received were mainly outside the scope of this research. However, a few relevant negative comments were received during the evaluation. There were no negative comments received from the participants after they had finished interacting with the first system and the following are the relevant negative comments received in the second system evaluation questionnaire:

- There were too many questions for the create-profile component;
- Some profile ratings were not correct, according to the participants;
- Some customer profiling functionalities were ambiguous, for example, updating a profile;
- During registration all fields should not be compulsory; and
- The user interface was not controllable.

The evaluation discussed thus far has not examined the SOA implementation of the customer profile back-end. There was a lack of formal SOA evaluation methods in literature at the time

of evaluation, despite the availability of web services-testing tools. The next section will discuss how the SOA evaluation was conducted in this research.

## 6.8 SOA IMPLEMENTATION EVALUATION

The goal of the SOA implementation evaluation was to determine whether the implemented customer profile was developed following SOA principles and exhibited SOA characteristics (Section 4.8). A number of ways exist in which SOA implementations can be evaluated (Salasin and Madni, 2007). In this research, the SOA implementation evaluation focused on how different SOA design principles were incorporated into the services implemented (Section 4.8).

Table 6.4 shows how each design principle was addressed during SOAD in this research. The results of this evaluation indicate that the customer profile JWS implemented exhibited SOA characteristics (Erl, 2008; Erl, 2005).

Metric	Customer profile web services design and implementation consideration
Standardised service contract	This was achieved by implementing the customer profile JWS as SOAP web services that were able to communicate using various standards (Sections 3.5 and 5.3.1). Service specification discussed in Section 4.3.2 provided further details on contracts required to communicate with customer profile web services.
Service loose coupling	This was achieved by implementing services as stand-alone components that were able to perform their functionalities without dependency on other services (Sections 4.3.1 and 5.3.1).
Service abstraction	This was achieved by having the services able to hide implementation details and only exposing their functions to service consumers (Section 4.7.3).
Service re-usability	This was achieved by implementing the services as loosely coupled components that were able to operate on the find-bind-execute paradigm by passing parameters between the service provider and the service consumer (Figure 3.5).
Service statelessness	This was achieved by including the @Stateless annotation in each customer profile JWS implementation (Section 5.3.1).
Service discoverability	This was achieved by having services deployed in a service registry (Figure 4.3).
Service composability	This was achieved by having the various services being able to work together and achieve a goal (Section 4.3.1).

Table 6.6: SOA implementation evaluation

## 6.9 SUMMARY

During the course of the research a number of research methodologies were used to achieve the objectives of the research, for example, a literature study, prototyping and an evaluation survey. This chapter has discussed the evaluation of the research, a vital tool in determining whether the research has met its goal and objectives, as outlined in Section 1.4. This chapter has discussed the instruments used in this evaluation, how the evaluations were conducted and how the results were obtained and analysed.

Three types of evaluations have been discussed, namely: pilot study, main end-user evaluation and SOA implementation evaluation. The major findings that transpired from this evaluation were that the customer profile components implemented were complete enough to perform the required customer profiling functionalities. Furthermore, the participants could not notice any significant differences between the SOA and CS systems, although the eye-tracking data revealed significant differences in terms of page loading times. Participants' eye-gaze heat maps indicated that the participants found the system easy to learn and the patterns observed were similar between the SOA and CS interactions.

Chapter 7 will conclude the research by discussing the research achievements, the contributions made and any possible recommendations. The chapter concludes with identification of the research limitations and possible future research avenues.

## CHAPTER 7: CONCLUSIONS

### 7.1 INTRODUCTION

Determining online customers' needs and preferences poses a challenge to many online businesses. Online businesses implement customer profiles to determine customer needs and preferences for recommendation and personalisation, among others. Recently businesses are increasingly adopting and implementing systems using cutting-edge technology in the form of SOA. The goal of this research has been to implement a customer profile in an SOA environment and to compare the implementation in the SOA environment with an implementation in the more traditional client-server environment.

The research achieved its goal by proposing, implementing and evaluating a customer profile model using SOA. The model was implemented using JWS in Netbeans IDE 6.5 and GlassFish ESB. An e-commerce website was developed using the Microsoft .NET framework technology to utilise the SOA-based customer profile implemented. End-user evaluation of the implemented customer profile model was conducted with 30 participants on two systems, one running on a client-server and another running on an SOA customer profile back-end. Eye tracking was used to obtain additional evaluation information.

The evaluation results revealed that the customer profile model implemented using SOA principles was found useful by the participants (Chapter 6). The results also indicated that the participants could not notice the differences between the client-server customer profile back-end system and the SOA customer profile back-end system. Further comparison between eye-tracking page loading time results indicated that the SOA system took significantly longer to complete the background interactions with the back-end customer profile web services than did the client-server system. This was proven by the inferential statistical evaluation that was conducted (Section 6.7.2).

In addition, the participants' eye-gaze heat maps obtained from the participants' interactions with the two systems produced similar patterns (Figures 6.20 and 6.21) when comparisons were drawn between the two systems. The heat maps' similar patterns suggested that:

*To the participants there was no difference, regardless of which background customer profiling system the interface was running.*

This chapter concludes the dissertation by revisiting the objectives of the research and showing how they have been met. This chapter also discusses the research achievements, the contributions that the research has made to the body of knowledge and suggesting possible recommendations for future research projects that could follow this research. The discussion in this chapter endeavours to answer research question 7 in Section 1.5.

## 7.2 RESEARCH OBJECTIVES REVISITED

The research objectives, as outlined in Section 1.4, were addressed as follows:

- *To understand customer profiling techniques and the various models used in businesses:* This objective was addressed in Chapter 2; and it explored the literature on e-commerce customer profiling;
- *To gain an understanding of current SOA models, systems and related technology, for example, web services and open source SOA implementation tools:* These objectives were addressed in Chapter 3. This chapter examined SOA and how it is used to implement systems;
- *To design and implement a suitable customer profile model in an SOA:* This was covered in Chapters 4 and 5. The analysis and design of the customer profiling system, using SOA and the implementation of the system respectively were investigated and discussed; and
- *To complete a comparative study between customer profile implementation, using an SOA and other distributed architecture, for example, client-server:* This was discussed in Chapter 6 and covered the evaluation and resultant analysis of the data collected.

The main research question that guided this research, as stated in Section 1.5, was:

*How can one implement a customer profile using SOA?*

A number of secondary research questions were formulated to assist in achieving the research objectives outlined (Sections 1.4 and 1.5). This section discusses how, during the course of the research, the research objectives were met by addressing the research questions, and by following the research methodology outlined in Figure 1.1 (Section 1.6).



### 7.2.1 Literature study

A literature study on how online businesses implement e-commerce and customer profiling systems was conducted. An investigation into the existing architectures, such as CORBA and DCOM was also performed. A number of uses of customer profiles in e-commerce, for example, personalisation, recommendation and customisation have been explored.

Furthermore, research into the recent paradigm of SOA was conducted. A definition of SOA for the purpose of this research was outlined. SOA-enabling technologies, such as web service, ESB and service-orientation were examined. BPM, notable SOA applications, for example, SaaS (software as a service) and cloud computing were also scrutinised. In addition, standard technologies that are commonly used in SOA, such as SOAP, WSDL and UDDI were explained. Contributions of organisations, such as OASIS, WS-I and W3C that focus on SOA standardisation, were additionally investigated. SOA benefits and challenges were also identified.

The aim of the literature study was to examine literature on e-commerce systems, customer profiling and SOA. The findings of the literature study were discussed in Chapter 2 and Chapter 3, and thereby the researcher managed to answer research questions 1 and 2 (Table 1.1). The literature study represents the first step of the research methodology discussed in Figure 1.1. The fundamental understanding obtained from the literature study provided an insight into customer profiling and SOA that was used to design and implement the proposed solution.

### 7.2.2 Prototyping

Prototyping the solution started with proposing a customer profile model using SOA that guided implementation. An investigation into how SOA services are derived, specified and realised was conducted following IBM's SOMA. Customer profile services were identified and specified that were used as input to the proposed customer profile model using SOA (Figure 4.3).

Evaluation of the possible open source implementation tools, such as SOAIF was conducted. This resulted in Netbeans IDE 6.5, GlassFish ESB and Microsoft SQL Server 2008 Express being selected for implementation avenues (Section 5.2). In addition, Microsoft Visual Studio 2008 was used to implement the e-commerce website that utilised the implemented customer profile web services.

The iterative prototyping methodology used enabled the incorporation of desirable changes during the analysis, design and implementation of the customer profile using SOA. The processes involved were discussed in Chapters 4 and 5, and provided answers to research questions 3, 4 and 5 (Table 1.1), and are represented by steps 2, 3, 4 and 5 of the research methodology discussed in Figure 1.1.

The next section will discuss the evaluation conducted on the prototype.

### 7.2.3 Evaluation

Three evaluations were conducted, namely: a pilot study, a main study and an SOA implementation evaluation. The pilot study and main study evaluation were conducted with samples of 8 and 30 participants respectively. The participants evaluated the system by conducting several tasks and completing post-task evaluation questionnaires (Appendices A, F, G and H). Eye tracking was used during the main study evaluation to capture additional evaluation metrics. Chapter 6 discussed the evaluation procedures and analysis of results that answered research question 6 (Table 1.1).

The main evaluation was conducted to test the usability of the customer profile implemented using SOA and to complete a comparative study between customer profile implementation methods, using two different architectures, namely SOA and client-server. The results suggested that:

*The participants did not notice any significant differences between the two systems; however, eye tracking results showed evidence that the SOA system was slower in performing back-end tasks than the client-server system (Section 6.7.2).*

The SOA system offered more benefits over implementation methods using client-server. For example, the JWS-based SOA system was interoperable with other platforms, such as, Microsoft .NET framework, services were standard-based and proved to be easy to implement and manage (Section 5.3). The evaluations conducted represented step 6 of the research methodology discussed in Figure 1.1.

## 7.3 RESEARCH ACHIEVEMENTS

This section will discuss the theoretical and practical achievements of the research.

### 7.3.1 Theoretical achievements

The theoretical achievement of the research focuses on the investigation done into customer profiling and SOA. In addition, the analysis and design of the proposed customer profile model, using SOA and the evaluation of the model, its usability, performance and SOA design principles are all notable achievements.

#### 7.3.1.1 Research conducted in customer profiling and SOA

An extensive literature study was conducted on customer profiling and SOA. The relevance, goals and objectives of the research were identified and a methodology was formulated for solving the problem.

#### 7.3.1.2 Customer profile model using SOA

The customer profile model was proposed to visually represent how customer profiling can be implemented in SOA and was used as a guide for its implementation. The customer profile model was proposed, based on concepts assembled from the literature review conducted on customer profiling and SOA. Various user-profile model designs were evaluated. However, no existing customer profile model could be found.

The proposed customer profile model outlined how the different technologies can seamlessly work together to implement a customer profile to be utilised as a service in an SOA environment (Figure 4.3).

### 7.3.2 Practical achievements

The practical achievement was demonstrated by the implementation of the customer profile model proposed in Section 4.7.3. Different technologies, such as Netbeans IDE 6.5, Microsoft Visual Studio 2008 and Microsoft SQL Server 2008 Express were used during the implementation. Implementation went further to demonstrate the interoperability of the customer profile between different technologies, for example, Microsoft .NET framework and Sun Java technologies.

An ESB and standard technologies play an important role in SOA implementation (Josuttis, 2007; Erl, 2005). This research identified and utilised the Open ESB and standard technologies such as SOAP, WSDL and UDDI to implement the web services identified (Sections 3.3, 3.5 and 5.3).

### 7.3.3 Major research findings

The major research findings of this research are:

- A practical understanding of how various open source SOA implementation tools can be used to implement an SOA-based system and how organisational goals can be aligned with IT;
- From the users' perspective an SOA-based system performed similarly to other systems implemented using other architectures, for example, the client-server implementation. This confirms that SOA does not affect the end product but rather the implementation style (Canfora, *et al.*, 2008; Erl, 2005);
- Eye tracking provided additional information that demonstrated the performance differences between the SOA and client-server systems. This had not been sufficiently evident from the participants' responses obtained; and
- SOA standards enabled different technologies to seamlessly interoperate and work together, for example, Microsoft .NET framework and Sun Java technologies.

## 7.4 RESEARCH CONTRIBUTIONS

This section discusses the contributions the research has made to the existing body of knowledge.

### 7.4.1 Theoretical contributions

Theoretically, the research contributed to an insight into SOA and how it can be implemented using a customer profile as a proof of concept. Section 1.2 discussed the problems being faced by businesses in trying to implement SOA. This research has investigated and reported the findings on SOA and how SOA-based systems can be implemented.

Figure 7.1 shows how a customer profile can be used as a service, serving thereby a number of e-commerce applications on the Internet. Using further SOA applications such as cloud computing and SaaS (Section 3.2.3), a customer profile can be implemented as a service and hosted in the cloud as shown in Figure 7.1.

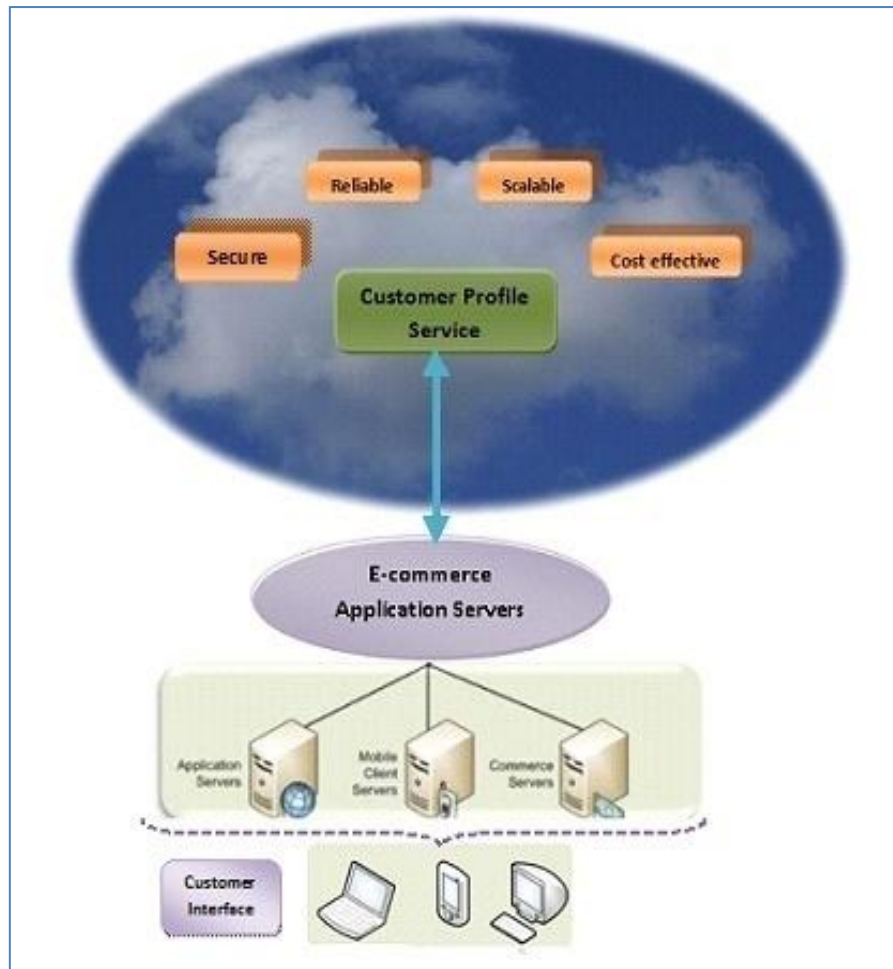


Figure 7.1: Customer profiling in an SOA environment

#### 7.4.2 Practical contributions

The practical contribution of this research focuses on the investigation of open source and other supporting SOA implementation tools that can be used to implement an SOA. Examples of such tools include: Netbeans IDE and GlassFish ESB, standards such as SOAP, WSDL and UDDI. A Java web services-based customer profile system was designed and implemented in SOA, using the identified SOA implementation tools. The successful implementation of the web services-based customer profile in SOA demonstrated how these tools can be practically used to implement a system in an SOA environment.

Furthermore, the evaluation that was conducted proved that the implementation was successful and had met its objectives.

## 7.5 RECOMMENDATIONS FOR FUTURE RESEARCH

A customer profile implemented in this research can be utilised by online businesses to provide personalised contents to customers. Businesses can adopt the model or its components to implement customer profiles for online customers. Different businesses require different information from customers to successfully provide personalised contents. Possible future research that can follow this research is investigating what aspects of a customer profile can be implemented as generic and what aspects can be implemented as private and specific for different businesses to meet individual organisations' strategic goals and objectives.

## 7.6 PROBLEMS ENCOUNTERED

A number of limitations were encountered during the course of the research. Literature on SOA was found to be irreconcilable and different SOA implementation techniques were found to be followed by major technology vendors, such as IBM and Microsoft. This was addressed by contacting and seeking assistance from various people who are actively and practically involved in SOA projects, for example, Ms Terlouw, PhD student at the University of Delft, Netherlands, and Enterprise Architect at ICRIS Consulting and Research, Netherlands and Mr Barry, principal of Barry & Associates Inc, USA.

The data collected during the evaluation required advanced statistical analysis that the researcher lacked. However, Mr Venter of the USC at NMMU, assisted with the data analysis and the statistical interpretation of the results.

Furthermore, SOA evaluation strategies have been found to be fairly developed as compared with evaluation strategies for other systems development techniques. Limited SOA evaluation guidelines exist and in this research guidelines for service-orientation were used to evaluate the SOA (Erl, 2008).

## 7.7 SUMMARY

Technology innovations are on the increase and digital convergence and seamless interoperability among software tenets is undoubtedly the most critical factor for businesses' success. Personalisation, based on customer profiles is shaping the future of e-commerce systems. The recent dawn of SOA, its related technologies and applications, such as SaaS,

cloud computing and web services are proving to be having a great impact on business success in today's challenging and ever-changing operating environments.

Using SOA and its related technologies, businesses can implement systems that can provide agility and promote interoperability. Customer profiles implemented using SOA principles can provide online businesses with a competitive advantage in offering timely and accurate customer information for personalisation. Research has shown that personalisation can increase customer loyalty, sales and consequently profits.

The goal of this research was to investigate how a customer profile can be implemented using SOA. This research has successfully implemented a customer profile model by using SOA. The research brought together different technologies, theories and practices that can be used to implement a customer profile using SOA for personalisation on the Internet. The research project has been completed and has achieved its goals and objectives. Limitations outlined in this research and an improved scope can be addressed by possible future research in customer profiles for personalisation on the web, using SOA.

This chapter concludes the research by summarising the work done, outlining the achievements and contributions made and by suggesting avenues for future research.

## REFERENCES

- Adomavicius, G. and Tuzhilin, A. (1999): User Profiling in Personalised Application through Rule Discovery and Validation. Proceedings of the 5<sup>th</sup> ACM SIGKDD international conference on knowledge discovery and data mining. San Diego, California, USA. ACM: pp. 377-381. August 1999.
- Adomavicius, G. and Tuzhilin, A. (2005): Personalisation Technologies: A process-oriented perspective. Communications of the ACM 48(10): pp. 83-90. October 2005.
- Amazon.com (2009): [www.amazon.com](http://www.amazon.com). [Accessed October 2009].
- Amazon Web Services (2009): <http://www.amazon.com/gp/browse.html?node=3435361>. [Accessed May 2009].
- Amsden, J. (2007): Modeling SOA: Part 2. Service Specification. IBM. Available online [www.ibm.com/developerworks/rational/library/07/](http://www.ibm.com/developerworks/rational/library/07/). [Accessed January 2009].
- Apache.org (2009): <http://www.apache.org>. [Accessed April 2009].
- Arsanjani, J. (2004): Service-oriented modeling and architecture (SOMA) How to identify, specify, and realize services for your SOA. Available online <http://www.ibm.com/developerworks/library/ws-soa-design1/>. [Accessed January 2009].
- Bakken, D. E. (2003): Middleware. Encyclopedia of Distributed Computing, Kluwer Academic Press, 2003. Available online <http://www.eecs.wsu.edu/~bakken/middleware-article-bakken.pdf>. [Accessed August 2008].
- Balke, W. and Wagner, M. (2003): Towards personalised selection of web services. WWW 2003, Budapest, Hungary. May 2003.
- Barry, D. K. (2003): Web services and service-oriented architecture. Morgan Kaufmann Publishers, San Francisco, CA, USA.
- Bayer, T. (2008): Open ESB and ServiceMix in comparison. Available online <http://www.predic8.com/opensb-servicemix-comparison.htm>. [Accessed May 2009].
- BBC (2008): [www.bbc.co.uk](http://www.bbc.co.uk). [Accessed October 2008].



- BBC (2009): UK government backs open source. Available online  
<http://news.bbc.co.uk/2/hi/technology/7910110.stm>. [Accessed March 2009].
- Bell, M. (2006): Service-Oriented Architecture: a planning and implementation guide for business and technology. John Wiley & Sons, Inc., New Jersey, USA.
- Bierberstein, N., Laird, R. G., Jones, K. and Mitra, T. (2008): Executing SOA: A Practical Guide for the Service-Oriented Architect. IBM Press.
- Birukou, A., Blanzieri, E., D'Andrea, V., Giorgini, P., Kokash, N. and Modena A. (2007): IC-service: a service-oriented approach to the development of recommendation systems. Proceedings of the 2007 ACM symposium on applied computing. Seoul, Korea. March 2007.
- Booch, G., Rumbaugh, J. and Jacobson, I. (2005): The unified modeling language user guide. 2<sup>nd</sup> Edition. Addison-Wesley, Upper Saddle River, USA.
- Brantner, M., Florescu, D., Graf, D., Kossmann, D. and Kraska, T. (2008): Building a database on S3. Proceedings of the SIGMOD'08. Vancouver, Canada. ACM: pp. 251-263. June 2008.
- Buyya, R., Yeo, C. S., Venegopal, S., Broberg, J. and Brandic, I. (2009): Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as 5<sup>th</sup> utility. Future Generation Computer Systems 25(2009). Available online:  
<http://www.sciencedirect.com>: pp. 599-616. [Accessed May 2009].
- Canfora, G., Fasolino, A. R., Frattolillo, G. and Tramontana, P. (2008): A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architecture. Journal of Systems and Software 81(2008): pp. 463-480.
- Chang, S. E., Changchien, S. W. and Huang, R. (2006): Assessing users' product-specific knowledge for personalization in e-commerce. Expert Systems with Applications 30: pp. 682-693.
- Choi, O. and Han, S. Y. (2008): Personalisation of rule-based web services. Sensors 2008, 8: pp. 2424-2435.

- Chung, R., Sundaram, D. and Srinivasan, A. (2007): Integrated personal recommender systems. Proceedings of the international conference on electronic commerce (ICEC). Minneapolis, Minnesota, USA. August 2007.
- Colgrave, J., Heritage, I., Marsh, K. and Skinner, E. (2007): Accessing the IBM WebSphere service registry and repository from .NET. IBM. Available online [http://www.ibm.com/developerworks/websphere/library/techarticles/0705\\_colgrave/0705\\_colgrave.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0705_colgrave/0705_colgrave.html). [Accessed September 2009].
- Computer Economics (2009): SOA Adoption Surges. Available online <http://www.computereconomics.com/article.cfm?id=1423&tag-rbspot>. [Accessed March 2009].
- Cotroneo, D., di Flora, C., Graziano, A. and Russo, S. (2008): Securing services in nomadic computing environments. Information and Software Technology 50(2008): pp. 924-947.
- Cugola, G. and Di Nitto, E. (2008): On adopting content-based routing in service-oriented architectures. Information and Software Technology 50 (2008): pp. 22-35.
- Cummins, F. A. (2009): Building the agile enterprise with SOA, BPM and MBM. Morgan Kaufmann OMG Press, Burlington, USA.
- Davis, A. and Zhang, D. (2002): A comparative study of DCOM and SOAP. Proceedings of the IEEE 4<sup>th</sup> international symposium on multimedia software engineering (MSE '02). IEEE: pp. 40-55. December 2002.
- Ebay.com (2008): <http://www.ebay.com>. [Accessed August 2008].
- Eichelberger, H. (2008): Automatic Layout of UML Use Case Diagrams. Proceedings of the ACM symposium on software visualization (SOFTVIS 2008). Herrsching, Germany. September 16-17.
- Eirinaki, M. and Vazirgiannis, M. (2003): Web mining for web Personalisation. ACM transactions on Internet technology 3(1): pp. 1-27. February 2003.
- Elemenope.org (2009): <http://www.elemenope.org>. [Accessed April 2009].

- Erl, T. (2005): Service-Oriented Architecture: concepts, technology and design. Prentice Hall, Upper Saddle River, NJ, USA.
- Erl, T. (2008): SOA-Principles of service design. Prentice Hall, Upper Saddle River, NJ, USA.
- Fiorano Software Inc. (2008): Service Oriented Architecture Implementation Frameworks: Understanding the Business Benefits of Services Oriented Architecture Implementation Framework (SOAIF). Available online <http://www.fiorano.com>. [Accessed September 2008].
- Flick, U. (2009): An introduction to qualitative research. 4<sup>th</sup> Edition. Sage publications Inc., London, United Kingdom.
- France, R., Evans, A., Lano, K. and Rumpe, B. (1998): The UML as a formal modeling notation. Computer Standards and Interfaces 19 (1998): pp. 352-334.
- Gretzel, U. and Fesenmaier, D. R. (2006): Persuasion in recommender systems. International Journal of electronic commerce 11 (2): pp. 81-100.
- Hammond, J. S. and Lo Giudice, D. (2008): Open Source Adoption: Notes from the Field. Forrester Research. Available online <http://www.forrester.com/Research/Document/Excerpt/0,7211,46279,00.html>. [Accessed March 2009].
- Hayes, B. (2008): Cloud Computing. Communications of the ACM: 51(7): pp. 9-11. July 2008.
- Henning, M. (2006): The rise and fall of CORBA. ACM Queue 4(5): pp. 28-34. June 2006.
- Hofgesang, P. I. (2007): Web personalisation through incremental individual profiling and support-based user segmentation. Proceedings of the IEEE/WIC/ACM international conference on web intelligence. Silicon Valley, USA. November 2007.
- Hubbers, J., Ligthart, A. and Terlouw, L. (2007): Ten Ways to Identify Services. Available online <http://www.soamag.com/I13/1207-1.asp>. [Accessed February 2009].

- Hynes, D. and Pradhan, S. (2006): Mastering SOA: Part 1: Build a portfolio of services.  
Available online <http://www.oracle.com/technology/tech/soa/mastering-soa-series/part1.html>. [Accessed May 2009].
- IBM (2007): [www.ibm.com/developerworks/library](http://www.ibm.com/developerworks/library). [Accessed June 2008].
- Inaganti, S. and Behara, G. K. (2007): Service Identification: BPM and SOA handshake.  
BPTrends. Available online <http://www.bptrends.com>. [Accessed May 2009].
- Ingesby, P. (2009): Telkom Roadshow Presentation. Nelson Mandela Metropolitan University. February, 2009.
- Jammes, F., Mensch, A. and Smit, H. (2005): Service-Oriented Device communications using the device profile for web services. Proceedings of the 3rd international workshop on middleware for pervasive and ad-hoc computing (MPAC) '05. Grenoble, France. November-December 2005.
- Java.sun.com (2005): <http://www.java.sun.com>. [Accessed August 2008].
- Java.sun.com (2009): Annotations. Available online  
<http://java.sun.com/docs/books/tutorial/java/javaOO/annotations.html>. [Accessed October 2009].
- Jokela, S., Turpeinen, M., Kurki, T., Savia, E. and Sulonen, R. (2001): The role of structured content in a personalised news service. Proceedings of the 34<sup>th</sup> annual Hawaii international conference on system science (HICSS-34). Washington DC. IEEE Computer Society: 7. January 2001.
- Josuttis, N. M. (2007): SOA in Practice: The art of distributed system design. O'Reilly Media, Inc., Sebastopol, CA, USA.
- Jtds.sourceforge.net (2009): <http://www.jtds.sourceforge.net>. [Accessed April 2009].
- Kalahari.net (2009): <http://www.kalahari.net>. [Accessed May 2009].
- Kim, J. and Lee, E. (2005): A Study of user Xquery method for personalization recommendation in semantic web. Transaction on engineering, computing and technology, ISSN 1305-5613: pp. 225-228.

- Kim, J. W., Lee, K. M., Shaw, M. J., Chang, H., Nelson, M. and Easley, R. F. (2006): A preference scoring technique for personalised advertisements on Internet storefronts. *Mathematical and Computer Modeling* 44(2006): pp. 3-15.
- Kulkarni, N. and Dwivedi, V. (2008): The role of service granularity in a successful SOA realization-a case study. 2008 IEEE congress on services (Services 2008): pp. 423-430.
- Kuno, H. and Sahai, A. (2002): My agent wants to talk to your service: Personalising web services through agents. Software Technology Laboratory, HP. Available online <http://www.hpl.hp.com/techreports/2002/HPL-2002-114.pdf>. [Accessed July 2008].
- Lee, T. Q., Park, Y. and Park, Y. (2008): A time based approach to effective recommender systems using implicit feedback. *Expert systems with applications* 34 (2008): pp. 3055-3062.
- Leedy, P. D. and Ormrod, J. E. (2001): *Practical Research: Planning and Design*. 7<sup>th</sup> Edition, Prentice Hall, Upper Saddle River, NJ.
- Levi, K. and Arsanjani, A. (2002): A goal-driven approach to enterprise component identification and specification. *Communications of the ACM*. 45(10): pp. 45-52. October 2002.
- Liu, D., Lin, Y., Chen, C. and Huang, Y. (2001): Deployment of personalised e-catalogues: An agent-based framework integrated with XML metadata and user models. *Journal of Network and Computer Applications* (24): pp. 201-228.
- Lopez-Sanz, M., Acuna, C. J., Cuesta, C. E. and Marcos, E. (2008): Modeling of service-oriented architecture with UML. *Electronic notes in theoretical computer science* 194 (2008): pp. 23-37.
- Mahmoud, Q. A. (2005): Service-oriented architecture (SOA) and web services: The road to enterprise application integration (EAI). Available online <http://www.java.sun.com>. [Accessed August 2008].
- Manifold Data mining inc. (2008): <http://www.manifolddatamining.com>. [Accessed April 2008].

- Martin-Guerrero, J. D., Palomares, A., Balaguer-Ballester, E., Soria-Olivas, E., Gomez-Sanchis, J. and Soriano-Asensi, A. (2006): Studying the feasibility of a recommender in a citizen web portal based on user modeling and clustering algorithms. *Expert systems with applications* 30 (2006): pp. 299-312.
- Maurizio, A., Sager, J., Jones, P., Corbitt, G. and Girolami, L. (2008): Proceedings of the 41<sup>st</sup> Hawaii international conference on systems sciences. Waikoloa, Big Island, USA. January 2008.
- Microsoft.com (2009): <http://www.microsoft.com/>. [Accessed July 2009].
- Miete, N. (2009): Configurability in SaaS Applications. Proceedings of the ISEC'09. Pune, India. pp. 19-26. February 2009.
- Millet, L. I., Friedman, B. and Felton, E. (2001): Cookies and Web browser design: Toward realising informed consent online. Proceedings of the SIGCHI'01. Seattle, Washington, USA. ACM 3(1): pp. 46-52. March-April 2001.
- Mohagheghi, P. and Aagedal, J. (2007): Evaluating quality in model-driven engineering. International workshop in modeling in software engineering (MISE '07). IEEE computer society.
- Murray, P. (2009): Enterprise grade cloud computing. Proceedings of the WDDDM'09. Nuremberg, Germany. ACM. March 2009.
- Nelte, M. and Saul, E. (2000): Cookies: Weaving the web into a state. The ACM Student Magazine. 7(1): pp. 10-13.
- Netbeans IDE (2009): Available online <http://www.netbeans.org/index.html>. [Accessed September 2009].
- Ntawanga, F., Calitz, A. P. and Barnard, L. (2008a): Dynamic customer profile for e-commerce websites. Proceedings of the Southern African institute for management scientists (SAIMS) 2008 conference. Misty Hill Hotel, South Africa. September 2008.
- Ntawanga, F., Calitz, A. P. and Barnard, L. (2008b): Maintaining customer profiles in an e-commerce environment. Proceedings of the Southern African institute of computer

scientists and information technologists (SAICSIT) 2008 conference. Wilderness Beach Hotel, South Africa. October 2008.

OASIS (2004): [www.oasis-open.org](http://www.oasis-open.org). [Accessed September 2008].

Oracle (2008): Business Process Management, Service-Oriented Architecture, and Web 2.0: Business Transformation or Train Wreck? Available online <http://www.oracle.com/technologies/bpm/docs/convergence-bpmsoa-whitepaper.pdf>. [Accessed April 2009].

Ordanini, A. and Pasini, P. (2008): Service co-production and value co-creation: The case for a service-oriented architecture (SOA). *European Management Journal* 26(5): pp. 289-297.

Papazoglou, M. P. (2003): Service-oriented computing: concepts, characteristics and directions. *Proceedings of the 4<sup>th</sup> international conference on web information systems engineering (WISE '03)*. Los Alamitos, CA, USA. IEEE 2003.

Papazoglou, M. P. and van den Heuvel, W. J. (2007): Service oriented architecture: approaches, technologies and research issues. *The VLDB Journal* 16 (2007): pp. 389-415.

Pastore, S. (2008): The service discovery methods issue: A web services UDDI specification framework integrated in grid environment. *Journal of Network and Computer Applications* 31 (2008): pp. 93-107.

Paterno, F. (2000): *Model-based design and evaluation of interactive applications*. Springer, London, United Kingdom.

Peterson, G. (2008): Security in SOA-It's the Car, Not the Garage. *The SOA Magazine*. Available online <http://www.soamag.com/I15/0208-2.asp>. [Accessed September 2009].

Pernice, K. and Nielsen, J. (2009): *Eye tracking Methodology: How to Conduct and Evaluate Usability Studies Using Eye tracking*. <http://www.useit.com/eyetracking/methodology>. [Accessed October 2009].

- Pretorius, M., Calitz, A. P. and van Greunen, D. (2005): The added value of eye tracking in the usability evaluation of a network management tool. Proceedings of the Southern African institute for computer scientists and information technologists (SAICSIT 05) conference. White River, South Africa.
- Qui, R., G. (2007): Enterprise Service Computing. Preview available online <http://books.google.co.za/books>. [Accessed May 2009].
- Raj, S. G. (1998): A detailed comparison of CORBA, DCOM and Java RMI. Available online <http://my.execpc.com/~gopalan/misc/compare.html>. [Accessed August 2008].
- Rajput, W. (2000): E-commerce systems, architecture and applications. Artech House Inc., Norwood, MA, USA.
- Research methods (2009): Non probability sampling. Available online <http://www.socialresearchmethods.net/kb/samprnon.php>. [Accessed October 2009].
- Rumbaugh, J., Jacobson, I. and Booch, G. (2004): The unified modeling language reference manual. 2<sup>nd</sup> Edition. Addison-Wesley, Boston, USA.
- Salasin, J. and Madni, A. M. (2007): Metrics for Service-Oriented Architecture (SOA) Systems: What developers should know. Transactions of the Society for Design and Process Science 11(2): pp. 55-77. June 2007.
- Salesforce.com (2009): <http://www.salesforce.com>. [Accessed May 2009].
- Sharp, H., Rodgers, Y. and Preece, J. (2007): Interaction Design, beyond human-computer interaction. 2<sup>nd</sup> Edition, John Wiley & Sons.
- Schneider, G. P. (2007): Electronic Commerce. 7<sup>th</sup> Annual Edition. Boston, Thomson.
- Shearin, S. and Lieberman, H. (2001): Intelligent profiling by example. Proceedings of the 6<sup>th</sup> international conference on intelligent user interfaces. Santa Fe, New Mexico, USA. ACM: pp. 145-151. January 2001.
- Sommerville, I. (2007): Software Engineering. 8<sup>th</sup> Edition. Pearson Education Limited, United Kingdom.
- SOA Patterns.org (2009): <http://www.soapatterns.org>. [Accessed June 2009].



- Srivihok, A. and Sukonmanee, P. (2005): Intelligent agent for e-tourism: Personalization travel support agent using reinforcement learning. WWW 2005, Chiba, Japan. May 2005.
- Steinberg, W. J. (2008): Statistics Alive! Sage publications Inc., Los Angeles, USA.
- Stevens, P. and Pooley, R. (2006): Using UML: Software engineering with objects and components. 2<sup>nd</sup> Edition. Pearson Education Limited, Harlow, England.
- Terlouw, L. (2009): Making a service catalog work: 3 do's and don'ts. Available online <http://www.servicespecification.com/?p=135>. [Accessed May 2009].
- Terlouw, L. and Maarse, K. E. (2009): A Service specification framework for developing component-based software: A case study at the Port of Rotterdam. Proceedings of the 5<sup>th</sup> international workshop on advances in enterprise engineering III. CAiSE 2009. Amsterdam, Netherlands. June 2009.
- Tobii.com (2009): Tobii T60 Eye Tracker. Available online <http://www.tobii.com>. [Accessed October 2009].
- Tullis, T. and Albert, B. (2008): Measuring the user experience: collecting, analysing and presenting usability metrics. Morgan Kaufmann, Burlington, MA, USA.
- Uddi.org (2004): [http://www.uddi.org/pubs/uddi\\_v3.htm#\\_Toc85907967](http://www.uddi.org/pubs/uddi_v3.htm#_Toc85907967). [Accessed August 2008].
- Uml.org (2009): <http://www.uml.org>. [Accessed June 2009].
- Vaquero, L. M., Rodero-Merino, L., Caceres, J. and Lindner, M. (2009): A break in the clouds: Towards a cloud definition. ACM SIGCOMM Computer Communication Review. ACM 39(1): pp. 50-55. January 2009.
- Veeraragadoo, M. (2008): SOA case study-converting KPI processes to SOA. 2008 IT Web Enterprise SOA Conference. Gallagher Estate, Johannesburg. October 2008.
- Venter, D. (2009) Personal Communication. Senior lecturer and statistical consultant at the Nelson Mandela Metropolitan University.

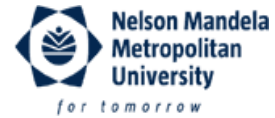
- Votis, K., Alexakos, C., Vassiliadis, B. and Likothanassis, S. (2008): An ontologically principled service-oriented architecture for managing distributed e-government nodes. *Journal of Network and Computer Applications* 31(2008): pp. 131-148.
- W3C (2004): Web Services Architecture. Available online <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. [Accessed June 2008].
- W3C (2007): SOAP Version 1.2. Available online <http://www.w3.org/TR/soap12>. [Accessed August 2008].
- W3C (2009): World Wide Web Consortium (W3C). <http://www.w3c.org>. [Accessed June 2009].
- Wagner, W. and Zubey, M. (2007): Customer relationship management: A people, process, and technology approach. Thomson Learning Inc., USA.
- Weightman, D. and McDonagh, D. (2003): People are doing it for themselves. DPPI '03. Carnegie Mellon University, Pittsburgh. June 2003.
- Woon, Y., Ng, W., Li, X. and Lu, W. (2003): Efficient Web Log Mining for Product Development. Proceedings of the 2003 international conference on cyberworlds (CW '03). IEEE Computer Society. December 2003.
- WS-I (2008): <http://www.ws-i.org>. [Accessed September 2008].
- XML.com (2008): <http://www.xml.com>. [Accessed August 2008].
- Zapthink (2009): SOA Strategy Comparison: IBM & Microsoft. Available online <http://www.zapthink.com/report.html?id=WP-0176>. [Accessed April 2009].
- Zhang, Y. and Jiao, J. R. (2007): An associative classification-based recommendation system for personalisation in B2C e-commerce applications. *Expert systems with applications* 33 (2007): pp. 357-367.
- Zimmermann, O., Krogdahl, P. and Gee, C. (2004): Elements of Service-Oriented Analysis and Design: An interdisciplinary modeling approach for SOA projects. IBM Developer works. Available online <http://www.ibm.com/developerworks/webservices/library/ws-soad1/> June 2009. [Accessed May 2009].
- Zoho.com (2009): <http://www.zoho.com>. [Accessed May 2009].

# APPENDICES

## Appendix A: Pilot Study Evaluation Tasks



**Department of Computer Science and  
Information Systems**  
**Nelson Mandela Metropolitan University**  
Tel: 041 504 2094/4168, Cell: 078 995 4545  
e-mail: Felix.Ntawanga@nmmu.ac.za



### Pilot study Evaluation Tasks

#### 1. Create your profile

- 1.1 Complete the biographical details form and click “*Next*” to proceed to the second step
- 1.2 Complete the customer profile questionnaire and click “*Submit*” to proceed
- 1.3 View your profile and record on the evaluation questionnaire in Section C, Question 1

#### 2. Purchase products: You can perform the following tasks in any order

**Note:** *Use the more or less product information option to provide sufficient information on products*

- 2.1 Purchase a vital brand health product containing 300 mg of concentrated garlic oil
- 2.2 Purchase a bottle of red wine made from vines of 8 years or older
- 2.3 Purchase a digital camera and DVD player with built-in internal HDD
- 2.4 Change the quantities of the health product and wine to 2
- 2.5 Submit your order

#### 3. Delete profile [optional], otherwise Logout

#### 4. Complete the rest of the evaluation questionnaire

Thank you for your participation!



## Pilot Study Background and Post-Task Questionnaire

### Section A: Computer and Internet Experience

1. How many years of computer experience do you have?

- ☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

2. How many years of Internet experience do you have?

- ☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

3. Where do you have Internet access? *[Mark all that apply]*

- ☐ Home      ☐ University      ☐ Work      ☐ Other(s) [specify] \_\_\_\_\_

4. What is the average number of hours you spend on the Internet per day?

- ☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

5. How would you rate your Internet experience?

- ☐ Expert      ☐ Intermediate      ☐ Novice

### Section B: Online Shopping and Profiling Experience

1. Have you ever purchased a product online?

- ☐ Yes      ☐ No      *[If No, go to question 4]*

2. What type of products have you purchased online?

- ☐ Books      ☐ Software      ☐ Music      ☐ Movies      ☐ Others [Specify] \_\_\_\_\_

3. From which online shopping website(s) have you purchased these items?

- ☐ Kalahari.net      ☐ Amazon.com      ☐ BidorBuy.co.za      ☐ Others [Specify] \_\_\_\_\_

4. Have you ever conducted an online registration before?

- ☐ Yes      ☐ No      *[If No, end of questionnaire]*

5. On which website(s) have you registered?

- ☐ Kalahari.net      ☐ Amazon.com      ☐ Facebook      ☐ Google Mail      ☐ Twitter  
☐ Others [Specify] \_\_\_\_\_

### Section C: Customer Experience

1. Record your initial customer profile.

Wine	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
Groceries	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
Electrical products	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced

2. How do you rate the following?					
2.1 Overall reaction to the system	Very frustrating			Very satisfying	
	1	2	3	4	5
2.2 Web pages design	Very unpleasant			Very pleasant	
	1	2	3	4	5
2.3 Navigation of the website	Very difficult			Very easy	
	1	2	3	4	5
2.4 Learnability	Very difficult			Very easy	
	1	2	3	4	5

3. Describe any observed negative aspects of the following:	
3.1 The web pages	
3.2 Website navigation	
3.3 Website Functionalities	

4. Any other comments you may have on the following:	
4.1 The web pages	
4.2 Website navigation	
4.3 Website Functionalities	

Thank you for your participation!



**Department of Computer Science and  
Information Systems**  
**Nelson Mandela Metropolitan University**  
Tel: 041 504 2094/4168, Cell: 078 995 4545  
e-mail: Felix.Ntawanga@nmmu.ac.za



### **Evaluation Procedure**

#### **1. Welcoming the participant into the usability lab**

- 1.1 An explanation of the environment
- 1.2 An explanation of the equipment to be used [eye tracker]
- 1.3 The purpose of the evaluation
- 1.4 An outline of the evaluation process

#### **2. Consent form**

- 2.1 Participant will be given consent form and time to read and understand the contents
- 2.2 Participant sign the consent form, if they agree with the contents
- 2.3 Participant completes background questionnaire

#### **3. Tasks**

- 3.1 Calibration of eye tracker
- 3.2 Participant perform task [Appendix F]

#### **4. Evaluation questionnaire**

- 4.1 Participant completes evaluation questionnaires
- 4.2 Repeating 3 and 4 for Second system

#### **5. Debriefing session**

- 5.1 Participant debriefed about the next step
- 5.2 Answer any question from the participant
- 5.3 Participant asked if they would like to view their recorded video
- 5.4 Thank the participant for their time

End of evaluation

## Appendix D: Consent Form



### Department of Computer Science and Information Systems Nelson Mandela Metropolitan University

Tel: 041 504 2094/4168, Cell: 078 995 4545

e-mail: Felix.Ntawanga@nmmu.ac.za



### Consent Form

Ref: \_\_\_\_\_

**Contact Person: Felix Ntawanga**

Dear Participant,

You have been selected to participate in this research study for the evaluation of Online Customer profiles. The researcher [Felix Ntawanga] will provide you with the necessary information to assist you in understanding the study and explaining what would be expected of you as a participant. Please feel free to ask the researcher to clarify anything that is unclear to you.

To participate, it will be required of you to complete the written consent section at the bottom of this form. Information required include; your surname and initials, signature, email address and date to verify that you understand and agree to the conditions. Participation in this research is completely voluntary. During the course of the evaluation you will be required to perform certain tasks with the system and your eye movements will be recorded by the eye tracker. You will also be required to complete questionnaires at the end of each of the two evaluation sessions. There are no known risks associated with this evaluation or the equipment to be used.

Please take note of the following:

- The evaluation is about the system and you are not being evaluated in any way and therefore be free and as honest as possible when interacting with the system and completing questionnaire;
- The data collected is solely going to be used by the researcher to evaluate the system and will not be viewed by anyone else;
- Your identity will never appear in any publication to be made from the results of this research;
- You have the right to withdraw from the evaluation at any point in time for any reason; and
- You may view your recorded data if you wish after the evaluation session;

Your time and effort for participating in this evaluation shall be greatly appreciated. Your signature below indicates that you have read and understood this consent form and that you voluntarily agree to participate.

Yours sincerely

**Felix Ntawanga**

**RESEARCHER**



**Department of Computer Science and  
Information Systems  
Nelson Mandela Metropolitan University**

Tel: 041 504 2094/4168, Cell: 078 995 4545

e-mail: Felix.Ntawanga@nmmu.ac.za



**Background Questionnaire**

Surname and initials		Contact Number	
Signature		Date	
Email address			

**Section A: Biographical details**

Surname and Initials		Gender	<input type="checkbox"/> Female	<input type="checkbox"/> Male
Age	<input type="checkbox"/> 17-20	<input type="checkbox"/> 21-25	<input type="checkbox"/> 26-30	<input type="checkbox"/> 31-35
Year of Study			Degree	
Ethnicity	<input type="checkbox"/> White	<input type="checkbox"/> Black	<input type="checkbox"/> Indian	<input type="checkbox"/> Coloured
Home Language	<input type="checkbox"/> English	<input type="checkbox"/> Afrikaans	<input type="checkbox"/> Xhosa	<input type="checkbox"/> Other _____

**Section B: Computer and Internet experience**

1. How many years of computer experience do you have?

☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

2. How many years of Internet experience do you have?

☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

3. Where do you have Internet access? *[Mark all that apply]*

☐ Home    ☐ University    ☐ Work    ☐ Other(s) [specify] \_\_\_\_\_

4. What is the average number of hours spent on Internet per day?

☐ <1      ☐ 1-2      ☐ 3-5      ☐ 6+

5. How would you describe your Internet experience?

☐ Expert    ☐ Intermediate    ☐ Novice

**Section C: Online Shopping and Profiling experience**

1. Have you browsed through an online shopping website before?

☐ Yes      ☐ No      *[If No, go to question 7]*

2. How many online shopping websites have you visited this year?

☐ 0      ☐ 1-2      ☐ 3-5      ☐ 6+

3. Have you ever purchased a product online?

☐ Yes      ☐ No      *[If No, go to question 7]*



4. How many items have you purchased online?

☐ 1-2      ☐ 3-5      ☐ 6+

5. What product categories have you purchased online?

☐ Books      ☐ Software      ☐ Music      ☐ Movies      ☐ Others [Specify] \_\_\_\_\_

---

6. From which online shopping website(s) have you purchased these items?

☐ Kalahari.net      ☐ Amazon.com      ☐ BidorBuy.co.za      ☐ Take2.co.za

☐ Others [Specify] \_\_\_\_\_

7. Have you ever conducted an online registration before?

☐ Yes      ☐ No      *[If No, end of questionnaire]*

8. On which website(s) have you registered?

☐ Kalahari.net      ☐ Amazon.com      ☐ Facebook      ☐ Google Mail      ☐ Twitter

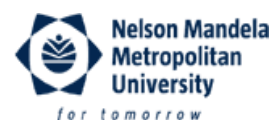
☐ Yahoo.com      ☐ Others [Specify] \_\_\_\_\_

## Appendix F: Main Study Evaluation Tasks



### Department of Computer Science and Information Systems Nelson Mandela Metropolitan University

Tel: 041 504 2094/4168, Cell: 078 995 4545  
e-mail: Felix.Ntawanga@nmmu.ac.za



### Evaluation Tasks

#### 1. Create your profile

- 1.1 Click on “*New customer register here*” link on the login page
- 1.2 Complete the biographical details form and click “*Next*” to proceed to the second step
- 1.3 Complete the customer profile questionnaire and click “*Submit*” to proceed

#### 2. Record initial customer profile

- 2.1 Click on the “*View Profile*” menu item
- 2.2 Record the profile ratings displayed on the evaluation questionnaire (Section C for 1<sup>st</sup> system and section G for 2<sup>nd</sup> system)

#### 3. Purchase products [Note: Change quantity before adding product to basket, click on the product picture to view its product information]

- 3.1 Click on “*Wine category*” and purchase a bottle of “*White Wine*” that is harvested from mid February between 24-26 Degrees Balling
- 3.2 Click on “*Grocery category*” and purchase a “*vital*” brand health product containing 300mg of garlic
- 3.3 Click on “*Electrical category*” and purchase a “*DVD player*” with a built-in Hard Disk Drive (HDD)

#### 4. Submitting order

- 4.1 Click on “*Basket*” menu item and ensure the purchases are correct
- 4.2 If purchases are correct, then click on “*CheckOut*” menu item [or follow link provided on basket] otherwise “*Update*” or “*Continue Shopping*” to make correct choices
- 4.3 Click “*Submit Order*” in “*Check out page*” to complete the purchase

#### 5. Record updated profile

- 5.1 Click on the “*View Profile*” menu item
- 5.2 Record the profile rating displayed on the evaluation questionnaire (Section C for 1<sup>st</sup> System and Section G for 2<sup>nd</sup> System)

#### 6. Delete profile [optional] Otherwise Logout

- 6.1 Click on the “*Delete Profile*” menu item [or “*Logout*”]
- 6.2 Read note and proceed accordingly
- 6.3 Press ESK key [to stop eye tracking recording]
- 6.4 Complete the rest of the questionnaire

## Appendix G: Post-Task Evaluation Questionnaire First System



**Department of Computer Science and  
Information Systems**  
**Nelson Mandela Metropolitan University**

Tel: 041 504 2094/4168, Cell: 078 995 4545  
e-mail: Felix.Ntawanga@nmmu.ac.za



### Post-task Evaluation Questionnaire First System

#### Section C: Customer profile

1. Record your initial customer profile.			
1.1 Wine	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
1.2 Groceries	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
1.3 Electrical products	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced

2. Do you agree with your profile?	Strongly Disagree					Strongly Agree
2.1 Wine	1	2	3	4	5	
2.2 Groceries	1	2	3	4	5	
2.3 Electrical products	1	2	3	4	5	

3. Record your updated customer profile.			
3.1 Wine	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
3.2 Groceries	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
3.3 Electrical products	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced

#### Section D: Customer profile Functionalities

How do you rate the following customer profiling functionalities?					
	Very frustrating				Very satisfying
1.1 Creating a profile	1	2	3	4	5
	Very unpleasant				Very pleasant
1.2 The process of creating a profile	1	2	3	4	5
	Very insufficient				Very sufficient
1.3 Profile information captured	1	2	3	4	5
	Not useful				Very useful
1.4 To display a profile	1	2	3	4	5
	Very insufficient				Very sufficient
1.5 Profile information displayed	1	2	3	4	5
	Dislike				Like
1.6 Viewing a profile	1	2	3	4	5
	Not useful				Very useful

1.7 Updating a profile	1	2	3	4	5
	Very unpleasant				Very pleasant
1.8 Having a profile updated	1	2	3	4	5
	Very ambiguous				Very clear
1.9 The process of updating a profile	1	2	3	4	5
	Not useful				Very useful
1.10 To delete a profile	1	2	3	4	5
	Very Unnecessary				Very Necessary
1.11 The process of deleting a profile	1	2	3	4	5

### Section E: Overall Task and Functionalities

1. Overall task satisfaction.	Strongly Disagree			Strongly Agree	
1.1 Overall, I am satisfied with the ease of completing the tasks	1	2	3	4	5
1.2 Overall, I am satisfied with the amount of time it took to complete the tasks	1	2	3	4	5
1.3 Overall, I am satisfied with the functionalities of the system	1	2	3	4	5

### Section F: Customer Profiling Components

<p>1. Customer profile contains the following components; <i>Create profile, identify customer, update profile, retrieve profile</i> and <i>delete profile</i>. Do you think any other component should be added? If so, list them.</p>     
<p>2. Do you think any component(s) should be deleted? [Mark all that apply]</p> <p> <input type="checkbox"/> Create profile    <input type="checkbox"/> Identify customer    <input type="checkbox"/> Update profile    <input type="checkbox"/> Retrieve profile    <input type="checkbox"/> Delete profile </p>

## Appendix H: Post-Task Evaluation Questionnaire Second System



### Department of Computer Science and Information Systems Nelson Mandela Metropolitan University

Tel: 041 504 2094/4168, Cell: 078 995 4545

e-mail: Felix.Ntawanga@nmmu.ac.za



### Post-task Evaluation Questionnaire Second System

#### Section G: Customer profile

##### 1. Record your initial customer profile

1.1 Wine	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
1.2 Groceries	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
1.3 Electrical products	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced

##### 2. Do you agree with your profile?

	Strongly Disagree				Strongly Agree
2.1 Wine	1	2	3	4	5
2.2 Groceries	1	2	3	4	5
2.3 Electrical products	1	2	3	4	5

##### 3. Record your updated customer profile

3.1 Wine	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
3.2 Groceries	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced
3.3 Electrical products	<input type="checkbox"/> Novice	<input type="checkbox"/> Intermediate	<input type="checkbox"/> Advanced

#### Section H: Systems Response

##### 1. Rate the two systems on the following criteria

		Very Slow					Very Fast
1.1 Page loading time	First System	1	2	3	4	5	
	Second System	1	2	3	4	5	
		Very Slow					Very Fast
1.2 Response time	First System	1	2	3	4	5	
	Second System	1	2	3	4	5	
		Very Difficult					Very Easy
1.3 Task completion	First System	1	2	3	4	5	
	Second System	1	2	3	4	5	
		Very Long/Slow					Very Short/Fast
1.4 Task completion times	First System	1	2	3	4	5	
	Second System	1	2	3	4	5	
		Very Inefficient					Very Efficient
1.5 Overall efficiency	First System	1	2	3	4	5	
	Second System	1	2	3	4	5	

## Section I: Overall Assessment of the Systems

		Very Terrible			Very Wonderful	
1	First System	1	2	3	4	5
	Second System	1	2	3	4	5
		Very Frustrating			Very Satisfying	
2	First System	1	2	3	4	5
	Second System	1	2	3	4	5
		Very Difficult			Very Easy	
3	First System	1	2	3	4	5
	Second System	1	2	3	4	5
		Not useful			Very Useful	
4	First System	1	2	3	4	5
	Second System	1	2	3	4	5
		Insufficient functions			Enough functionalities	
5	First System	1	2	3	4	5
	Second System	1	2	3	4	5

## Section J: General Comments

1. Did you notice any difference(s) between the first and the second system?

☐ Yes      ☐ No      *[If yes, specify]:*

2. What did you like most about the customer profiling component of the systems?

2.1 First system

2.2 Second system

3. What did you like least about the customer profiling component of the systems?

3.1 First system

3.2 Second system

4. Do you have any comment or suggestions about the following?

4.1 Initial Customer profile:
4.2 Updated customer profile:
4.3 System response:
4.4 First system:
4.5 Second system:
5. Any other general comments or suggestions?

## Appendix I: Main Evaluation Results

Metric	Yes	No
Browsed e-commerce website	28	1
Purchased products online	10	19
Conducted online registration	28	1

Table 1: Online shopping and registration summary [n=29]

Statistics	Registration Time		Create Profile Time	
	SOA	CS	SOA	CS
n	27	28	27	28
Mean	2.23	2.01	2.00	1.52
SD	1.20	1.07	1.07	0.72
Min	1.04	0.64	0.86	0.57
Quartile 1	1.25	1.10	1.22	1.05
Median	1.72	1.90	1.54	1.26
Quartile 3	2.71	2.90	2.71	1.70
Maximum	5.56	4.30	4.73	3.59

Table 2: Descriptive statistics summary of eye tracking page loading times

System Response (SR)		Overall Assessment (OA)		Difference SR	Difference OA
SOA	CS	SOA	CS	SOA-CS	SOA-CS
3.60	4.20	3.20	4.00	-0.60	-0.80
5.00	5.00	4.60	4.60	0.00	0.00
4.00	5.00	3.80	5.00	-1.00	-1.20
4.00	4.20	4.20	4.20	-0.20	0.00
4.20	4.60	4.40	4.40	-0.40	0.00
5.00	5.00	5.00	5.00	0.00	0.00
4.20	4.20	3.40	3.60	0.00	-0.20
5.00	4.40	4.80	4.20	0.60	0.60
3.60	4.20	4.40	4.20	-0.60	0.20
4.40	4.80	4.20	4.20	-0.40	0.00
4.40	5.00	4.00	4.60	-0.60	-0.60
4.20	4.20	4.00	4.00	0.00	0.00
4.80	4.80	4.40	4.40	0.00	0.00
5.00	4.80	4.40	4.40	0.20	0.00
4.80	4.80	4.20	4.00	0.00	0.20
4.40	4.20	4.00	4.00	0.20	0.00
4.40	4.40	4.20	4.20	0.00	0.00
5.00	5.00	5.00	5.00	0.00	0.00
4.80	5.00	4.60	4.60	-0.20	0.00
4.80	4.60	4.80	4.20	0.20	0.60
3.80	3.80	2.20	2.20	0.00	0.00
4.60	4.20	4.60	4.20	0.40	0.40
4.40	4.60	4.40	4.40	-0.20	0.00



5.00	3.60	4.40	4.20	1.40	0.20
4.00	4.00	4.00	4.00	0.00	0.00
5.00	4.60	5.00	4.60	0.40	0.40
5.00	4.60	4.60	4.60	0.40	0.00
5.00	4.60	5.00	5.00	0.40	0.00
4.80	4.80	4.00	4.00	0.00	0.00

Table 3: Summary of participants' responses to system response and overall assessment  
[n=29]

Registration Time		Create Profile Time	
SOA	CS	SOA	CS
3.283	0.638	3.520	1.300
2.601	1.182	2.626	1.172
2.887	1.107	2.829	0.883
2.175	1.069	1.736	0.913
2.518	0.743	1.006	0.979
3.861	1.313	3.863	1.887
2.572	1.167	2.243	1.474
2.627	0.954	1.184	1.020
2.629	0.785	1.408	1.056
5.563	1.449	4.065	1.610
2.782	1.277	3.229	1.099
3.951	1.175	2.789	1.204
1.346	2.419	1.365	1.323
1.717	2.598	1.543	3.127
1.084	0.720	1.580	1.428
1.430	3.054	1.016	2.857
1.514	0.885	1.072	0.567
1.132	2.718	1.226	0.932
1.549	2.881	1.896	1.643
1.141	3.708	0.863	1.894
1.166	2.741	1.096	1.041
1.041	2.566	1.461	1.078
1.059	3.226	1.216	3.592
1.208	2.958	1.527	2.611
1.294	3.053	1.242	1.217
1.465	3.162	1.708	1.692
4.588	4.301	4.728	1.722

Table 4: Summary of participants' responses to system response and overall assessment  
[n=27]

Create Profile	View Profile	Update Profile	Delete Profile	Profile satisfaction	Overall Satisfaction
3.67	4.00	3.33	3.00	3.50	3.67
5.00	5.00	4.00	5.00	4.75	4.67
3.33	4.00	4.67	3.50	3.88	4.67
4.33	4.33	3.67	4.00	4.08	4.33
4.00	4.67	3.67	5.00	4.33	5.00
4.67	5.00	4.33	5.00	4.75	4.67
4.00	3.67	4.00	4.00	3.92	4.67
4.67	4.67	4.67	5.00	4.75	4.67
5.00	5.00	3.67	2.50	4.04	4.33
3.67	4.67	3.67	3.50	3.88	4.67
4.33	3.33	2.67	4.50	3.71	3.00
4.00	3.67	3.67	4.00	3.83	3.67
4.00	4.00	4.67	5.00	4.42	4.67
4.67	4.33	4.67	4.50	4.54	4.33
4.67	4.67	4.33	3.00	4.17	4.67
4.33	4.33	4.67	4.00	4.33	4.00
4.00	3.67	3.00	4.00	3.67	4.33
5.00	4.67	5.00	4.00	4.67	5.00
4.33	3.67	4.67	4.00	4.17	4.67
4.33	3.67	4.33	4.00	4.08	5.00
4.33	3.33	2.67	1.00	2.83	3.67
4.33	4.00	4.33	4.00	4.17	4.67
4.67	4.00	5.00	5.00	4.67	5.00
4.33	4.33	4.33	5.00	4.50	4.67
4.00	3.67	3.33	3.00	3.50	4.67
5.00	4.67	4.00	5.00	4.67	5.00
4.67	4.33	4.33	5.00	4.58	4.67
5.00	4.67	3.67	5.00	4.58	5.00
4.67	4.33	4.67	5.00	4.67	5.00

Table 5: Summary of participants' responses to customer profiling components [n=29]



Figure 1: Client-server (left) and SOA (right) eye tracking heatmap

## Appendix J: Conference Proceedings Abstracts

Southern African Institute for Management Scientist (SAIMS) 2009 Conference.  
Summerstrand Hotel, Port Elizabeth, South Africa. September 2009.

**Title:** Customer Profiling as a Service on the Internet

**Authors and address:** Felix Ntawanga, André P. Calitz and Lynette Barnard, Department of  
Computing Sciences, NMMU, P.O. Box 77000, Port Elizabeth, South  
Africa.

### Abstract

E-commerce and Internet-based transactions are now integrated into business functions. Internet-based transactions are efficient, reliable and can lead to a reduction in transaction costs. For bigger companies, the trend is to implement and host e-commerce systems in-house on a client-server computer architecture. Internet technologies continue to evolve and IT businesses are now offering their products as services that can be accessed and utilised on demand, with customers only paying for actual service usage. The growth of Web 2.0 concepts has led to the development and evolution of web-based communities (such as Facebook), hosted services and applications. Hosted services and applications, for example Software as a Service (SaaS) where software is accessed on the Internet and utilised on demand, is being utilised by new business ventures.

This research investigates and proposes a model businesses can utilise to implement a customer profile as a service on the Internet. Online businesses incorporate customer profiles in E-commerce applications to assist with the implementation of one-to-one marketing strategies, such as personalisation. Research has proven that personalisation can lead to an increase in customer loyalty. Customer profiling as a service can be beneficial to an online business in that customer's needs and preferences can be accessed on demand, while data security, scalability, efficiency and reliability of the data are handled by a third party.

**Keywords:** E-Commerce, Customer profiling, Internet services, Service Oriented Architecture.

Southern African Telecommunications Networks and Application Conference (SATNAC) 2009. Royal Swazi Spa, Swaziland-August-September 2009.

**Title:** Customer profile model using Service-Oriented Architecture

**Authors and address:** Felix Ntawanga, André P. Calitz and Lynette Barnard, Department of Computing Sciences, NMMU, P.O. Box 77000, Port Elizabeth, South Africa.

### **Abstract**

Customer profiling has gained much recognition in the e-commerce domain because of the benefits it is capable of bringing to online businesses. Online businesses utilise customer profiles to assist implementing one-to-one marketing strategies in order to attract new customers and retain existing customers. Personalisation is one successful method used by online businesses to implement one-to-one marketing strategies and research has revealed that personalisation can improve customers' satisfaction levels, purchasing behaviour, loyalty and consequently increase sales. Organisations are increasingly adopting and implementing e-commerce systems using service-oriented architecture (SOA) principles. Incorporating customer profiling into SOA e-commerce systems is vital for survival and success on the competitive Internet environment. This research sets out to investigate how a customer profile can be developed and implemented using SOA and how customer profiles can be used to provide appropriate personalisation in an SOA environment. The research further endeavour to complete a comparative study on customer profile implementation in SOA and other existing architectures such as client-server.

11<sup>th</sup> Annual conference on World Wide Web applications (ZAWWW2009). Pine Lodge Hotel, Port Elizabeth, South Africa. September 2009.

**Title:** Customer evaluation of personal customer profiles

**Authors and address:** Lynette Barnard, André P. Calitz and Felix Ntawanga, Department of Computing Sciences, NMMU, P.O. Box 77000, Port Elizabeth, South Africa.

### **Abstract**

The Internet and its related technologies have significantly changed the way businesses operate. As a direct result of these technologies, on-line businesses now have access to customers on a global scale. The nature of the Internet prevents direct personal contact with a customer as an individual. In order to provide a customer with products and services that best suit the needs of each individual customer, the on-line company needs to know its customers individually. Customer profiling is a method used by on-line companies to collect information about customers and to get to know the individual customer's needs and preferences. Obtaining information regarding the needs and preferences of each individual customer is a major challenge facing on-line businesses. Questionnaires and various web-based technologies are used to collate this information. Once this information is obtained, initial customer profiles are created. The purpose of the customer profile is then to enable the on-line business to provide each individual customer with personalised products and services. The current research project investigated methods to create and maintain customer profiles. An e-commerce website was developed to cater for the creation and maintenance of customer profiles. The customer profiles were used to provide personalised product information. Customers evaluated their personal customer profiles created by the e-commerce system. The results indicated that Internet customers do prefer personalised product information based on an individual customer profile and customers appreciate to be familiar with their own customer profile.

**Keywords:** Customer profile, E-commerce, Internet marketing

Southern African Institute of Computer Science and Information Technology (SAICSIT) 2008 Conference. Wilderness Beach Hotel, South Africa. October 2008.

**Title:** Maintaining customer profiles in an e-commerce environment

**Authors and address:** Felix Ntawanga, André P. Calitz and Lynette Barnard, Department of Computing Sciences, NMMU, P.O. Box 77000, Port Elizabeth, South Africa.

### **Abstract**

Recent studies have indicated an increase in customer profiling techniques used by e-commerce businesses. E-commerce businesses are creating, maintaining and utilising customer profiles to assist in personalisation. Personalisation can help improve customers' satisfaction levels, purchasing behaviour, loyalty and subsequently improve sales. The continuously changing customer needs and preferences pose a challenge to e-commerce businesses on how to maintain and update individual customer profiles to reflect any changes in customers' needs and preferences.

This research set out to investigate how a dynamic customer profile for on-line customers can be updated and maintained, taking into consideration individual web visitors' activities. The research designed and implemented a decision model that analysed on-line customers' activities during interaction sessions and determined whether to update customers' profiles or not. Evaluation results indicated that the model was able to analyse the on-line customers' activities from a log file and successfully updated the customers' profiles, based on the customer activities undertaken during the interaction session.

Southern African Institute for Management Scientist (SAIMS) 2008 Conference. Misty Hills Hotel, South Africa. September 2008

**Title:** Dynamic customer profile for e-commerce websites.

**Authors and address:** Felix Ntawanga, André P. Calitz and Lynette Barnard, Department of Computing Sciences, NMMU, P.O. Box 77000, Port Elizabeth, South Africa.

### **Abstract**

On-line businesses take advantage of personalisation technologies to make their products and services unique and to tailor their product information for specific customers' needs and preferences. Research has shown that personalisation can improve customers' satisfaction levels, purchasing behaviour, loyalty and consequently increase sales. Knowledge about the customer is fundamental for the establishment of effective personalised services. On-line businesses are creating and maintaining customer profiles and recording the customers' product knowledge using questionnaires that ask customers to rate products and services. This method suffers a flaw of users assigning arbitrary ratings not representing their true opinions.

The aim of this research is to design an on-line system that can dynamically create and maintain customer profiles. The research evaluates the accuracy of the customer profile and the usefulness of providing personalised product information based on the customer's profile.

Results indicated that the on-line product knowledge questionnaire developed to determine the customers' initial product category knowledge was suitable. Customers were satisfied with the initial customer profiles the system created as well as with the updated profile. Profiles were updated based on customer behaviour and activities during an on-line product purchase session. Results also indicated that the personalised product information that was displayed, based on the customers' profiles, was useful. The customer profile can thus be used to provide personalised product information.