A MIDDLEWARE INTEGRATING ERP, CRM
AND SUPPLY CHAIN MANAGEMENT SYSTEM
USING SERVICE ORIENTED ARCHITECTURE

CHONG KWONG CHEN

FACULTY OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

MAY 2011
A MIDDLEWARE INTEGRATING ERP, CRM AND SUPPLY CHAIN MANAGEMENT SYSTEM USING SERVICE ORIENTED ARCHITECTURE

CHONG KWONG CHEN

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA KUALA LUMPUR

MAY 2011
ABSTRACT

In the era of competitive business world, ERP, CRM and SCM system has become among the most influential enterprise systems in term of improving competitive advantages of an organization. However, most of the systems running in companies are continue to exist in isolation. They become less relevant in today business context due to lack of integrated information achievable through each of the system respectively. This dissertation introduces an idea of service-sharing system integration approach named SCAIM that consists of a conceptual model and a connector to integrate the participated systems at conceptual level. Through the development of the conceptual model and the connector, any representational and services participated are organized in a center component and integrity these systems functionalities is mapped conceptually. Hence, the capability of SCAIM to integrate ERP, CRM and SCM is tested by using different standard testing approaches. Basically, the research methodology chosen for this dissertation is survey research; hence survey questionnaire and interview are two modes of survey research used to conduct the survey and data analysis. Of 40 survey responses, majority agreed that ERP, CRM and SCM are most suitable among the business applications to be integrated to improve competitive advantages of an organization. Moreover, this research supports the idea that SCAIM increases the integration level of ERP, CRM and SCM effectively.
ABSTRAK

ACKNOWLEDGEMENT

I would like to take this opportunity to convey my deepest gratitude to all who have rendered assistance to me, either directly or indirectly, in completing this dissertation.

Firstly I would like to express my appreciation to my current supervisor, Dr. Norizan Mohd Yasin, who has supervised me throughout this dissertation. I would like to thank her for his continuing guidance, encouragement and advises, especially for her trusty and having faith in my capabilities of designing and developing the system.

Apart from this, I would like to convey my special thanks to all postgraduates from FCSIT University of Malaya, and respondents towards the research survey, who have spending valuable time for providing some helpful information during interview and questionnaire.

Finally, the unwavering moral, emotional and financial support from family and friends is highly appreciated. I thank you all for the opportunity to fulfill this research.
# TABLE OF CONTENT

ABSTRACT .................................................................................................................. I

ABSTRAK .................................................................................................................. II

ACKNOWLEDGEMENT ............................................................................................ III

TABLE OF CONTENT ............................................................................................... IV

LIST OF FIGURES .................................................................................................... XI

LIST OF TABLES ........................................................................................................ XV

CHAPTER 1: INTRODUCTION ................................................................................... 1

1.1 Introduction ......................................................................................................... 1

1.2 Background of the Study .................................................................................. 2

1.3 Problem Statement .......................................................................................... 4

1.4 Objective of the Dissertation ......................................................................... 7

1.5 Expected Outcome .......................................................................................... 8

1.6 Scope of the Dissertation ............................................................................... 9

1.7 Significance of the Dissertation ................................................................... 10

1.8 Brief Research Methodology ....................................................................... 10

1.9 Organization of the Dissertation .................................................................. 12

1.10 Summary ....................................................................................................... 13

CHAPTER 2: LITERATURE REVIEW .......................................................................... 14

2.1 Introduction ..................................................................................................... 14

2.2 Review on ERP, CRM, and SCM ................................................................. 15

2.2.1 Enterprise Resource Planning (ERP) .................................................... 15

2.2.1.1 ERP Objectives ............................................................................. 15

2.2.1.2 ERP System Architecture ............................................................. 16

2.2.2 Customer Relationship Management (CRM) ....................................... 18
2.5.2.2 Advantages of MOM ................................................................. 42
2.5.2.3 MOM Limitations ................................................................. 43
2.5.3 Integration Broker ................................................................. 43
2.5.3.1 Integration Broker Architecture ........................................... 44
2.5.3.2 Integration Broker Advantages ............................................. 45
2.5.3.3 Integration Broker Disadvantages ........................................ 45
2.5.4 Comparison of ESB, MOM, and Integration Broker .................... 45

2.6 Conceptual Framework of Integration Review ........................................ 46

2.7 Conclusion .................................................................................. 48

CHAPTER 3: RESEARCH METHODOLOGY .................................................. 49
3.1 Introduction ................................................................................. 49
3.2 Survey Questionnaire ................................................................. 49
3.3 Questionnaire Design ................................................................. 50
3.4 Data Collection ........................................................................... 52
3.5 Pilot Test ..................................................................................... 53
3.5.1 Pilot Test Result Analysis .......................................................... 55
3.6 Administration of Questionnaire ................................................... 56
3.7 The Interview ............................................................................. 57
3.8 Population and Sampling Techniques ........................................... 58
3.9 Justification of Research Sampling Methods .................................... 60
3.10 Conclusion ............................................................................... 61

CHAPTER 4: DATA ANALYSIS AND FINDINGS ......................................... 63
4.1 Introduction ................................................................................ 63
4.2 Data Analysis for Judgment and Random Sampling ....................... 64
4.2.1 General Information about the Respondents and Companies .......... 64
4.2.1.1 Respondents’ Role in the Company ....................................... 64
4.2.1.2 Size and Business Area of Respondents’ Company ......................... 65
4.2.1.3 ERP, SCM and CRM Related Systems in the Company .................... 66
4.2.1.4 ERP, CRM and SCM Integration in the Companies .......................... 67
4.2.2 Level of System Integration in the Companies ...................................... 68
4.2.2.1 Current Systems Implemented in the Companies ............................. 68
4.2.2.2 Users of the Systems .......................................................................... 70
4.2.2.3 Systems Developed by the Same Vendor .......................................... 71
4.2.2.4 Information Sharing among the Systems ............................................ 71
4.2.2.5 Maintaining Systems that are Not Initially Work Together ............... 72
4.2.2.6 Integrating Different Systems to Fulfill Business Requirements ........ 72
4.2.2.7 Integration Level of the Systems ...................................................... 73
4.2.2.8 Middleware Suites Implemented in the Companies ......................... 74
4.2.2.9 Electronic Data Interchange (EDI) Implemented in the Companies ...... 75
4.2.3 Perceptions of Respondents on System Integration ............................... 76
4.2.3.1 Importance of System Integration ...................................................... 76
4.2.3.2 Current Levels of System Integration ............................................... 78
4.2.3.3 Preferred Methods of System Integration .......................................... 78
4.2.3.4 Barriers to System Integration Implementation ................................. 79
4.2.3.5 Potential Systems to Participate in the Integration .............................. 80
4.2.4 Promotion of System Integration using SOA ........................................ 81
4.2.4.1 Participants of the System Integration .............................................. 81
4.2.4.2 Service Oriented Methodology in the Integration .............................. 81
4.2.4.3 Functionalities of the Integration ...................................................... 82
4.2.5 Other Opinions ...................................................................................... 83
4.2.5.1 Beneficial of System Integration ....................................................... 83
4.2.5.2 Limitations and Weaknesses of Current Systems .............................. 84
CHAPTER 5: SYSTEM DEVELOPMENT ................................................................. 91

5.1 Introduction ..................................................................................................... 91

5.2 System Requirement Analysis ......................................................................... 91

5.2.1 Functional Requirements ........................................................................... 92

5.2.1.1 Use Case Diagrams ............................................................................... 94

5.2.2 Non-Functional Requirements .................................................................... 95

5.3 System Design ................................................................................................. 97

5.3.1 SCAIM Architecture Design ...................................................................... 97

5.3.1.1 Conceptual Model Module ................................................................. 98

5.3.1.2 Adaptive Service Connector Module ............................................... 99

5.3.1.3 ERP, CRM and SCM System Module ............................................. 100

5.3.1.4 ERP, CRM and SCM Integration through the Architecture .......... 101

5.3.2 SCAIM Functionality Design ................................................................... 103

5.3.3 Integration Success through Conceptual and Service Mapping ............ 112

5.3.4 SCAIM Component Design .................................................................... 115

5.3.5 Database Design ....................................................................................... 119

5.3.5.1 Database Structure ............................................................................. 119

5.3.5.2 Data Dictionary .................................................................................. 121

5.3.6 SCAIM Interface Design ........................................................................ 121

5.3.6.1 Input Design ....................................................................................... 121

5.3.6.2 Output Design .................................................................................... 122

5.4 System Implementation ............................................................................... 123

5.4.1 Conceptual Model Implementation ......................................................... 123
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.2 Adaptive Service Connector Implementation</td>
<td>127</td>
</tr>
<tr>
<td>5.4.3 Service Dictionary Implementation</td>
<td>128</td>
</tr>
<tr>
<td>5.4.4 ERP, CRM and SCM Systems Implementation</td>
<td>130</td>
</tr>
<tr>
<td>5.4.5 Systems Database Implementation</td>
<td>132</td>
</tr>
<tr>
<td>5.4.6 Conceptual Mapping and Service Binding Implementation</td>
<td>134</td>
</tr>
<tr>
<td>5.4.7 System Web Services Implementation</td>
<td>141</td>
</tr>
<tr>
<td>5.4.7.1 Web Services Enhancement 3.0</td>
<td>143</td>
</tr>
<tr>
<td>5.4.7.2 Security Sub-Component</td>
<td>145</td>
</tr>
<tr>
<td>5.4.7.3 Invocation Sub-Component</td>
<td>148</td>
</tr>
<tr>
<td>5.4.7.4 Addressing Sub-Component</td>
<td>150</td>
</tr>
<tr>
<td>5.4.7.5 Messaging Sub-Component</td>
<td>152</td>
</tr>
<tr>
<td>5.4.7.6 Routing Sub-Component</td>
<td>155</td>
</tr>
<tr>
<td>5.4.8 Business Components Implementation</td>
<td>156</td>
</tr>
<tr>
<td>5.4.9 Data Access Implementation</td>
<td>159</td>
</tr>
<tr>
<td>5.4.10 Integration Output Implementation</td>
<td>159</td>
</tr>
<tr>
<td>5.5 System Testing</td>
<td>162</td>
</tr>
<tr>
<td>5.5.1 Unit Testing</td>
<td>163</td>
</tr>
<tr>
<td>5.5.1.1 Program Coding Correction</td>
<td>163</td>
</tr>
<tr>
<td>5.5.1.2 Debugging Strategy</td>
<td>164</td>
</tr>
<tr>
<td>5.5.1.3 Black Box Testing</td>
<td>165</td>
</tr>
<tr>
<td>5.5.1.4 White Box Testing</td>
<td>166</td>
</tr>
<tr>
<td>5.5.2 Module Testing</td>
<td>169</td>
</tr>
<tr>
<td>5.5.3 Integration Testing</td>
<td>170</td>
</tr>
<tr>
<td>5.5.4 System Testing</td>
<td>170</td>
</tr>
<tr>
<td>5.5.5 Non-Functional Requirement Testing</td>
<td>171</td>
</tr>
<tr>
<td>5.6 System User Evaluation</td>
<td>174</td>
</tr>
</tbody>
</table>
5.7 Chapter Summary ........................................................................................................ 176

CHAPTER 6: CONCLUSION AND IMPLICATION ........................................... 177

6.1 Introduction ........................................................................................................ 177

6.2 Problems Encountered ....................................................................................... 177

6.3 System Strength and Contribution ....................................................................... 179

6.4 Limitations and Weaknesses .............................................................................. 182

6.5 Future Enhancement ......................................................................................... 183

6.6 Conclusion ........................................................................................................ 184

REFERENCES ......................................................................................................... 186

Appendix A: Survey Questionnaire ........................................................................... 192

Appendix B: SCAIM Use Case Tables ...................................................................... 202

Appendix C: SCAIM Data Dictionary ....................................................................... 208

Appendix D: SCAIM User Manual ........................................................................... 209

Appendix E: SCAIM Component Source Codes ...................................................... 231

Appendix F: SCAIM System Evaluation Form .......................................................... 243
LIST OF FIGURES

Figure 2.1: Common ERP Modules ................................................................. 16
Figure 2.2: Conceptual Pre-requisites for Integration Success ......................... 22
Figure 2.3: Extended Enterprise .................................................................. 27
Figure 2.4: Functional Interoperability Stack ................................................. 30
Figure 2.5: ESB Five Tiers Topology .............................................................. 40
Figure 2.6: A Typical MOM Architecture ....................................................... 42
Figure 2.7: Extended Enterprise Integration with Integration Broker .............. 44
Figure 4.1: Size of Respondents’ Company ................................................... 66
Figure 4.2: Business Area of Respondents’ Company ..................................... 66
Figure 4.3: User of the Existing Systems ....................................................... 70
Figure 4.4: Information Sharing among Existing Systems ............................... 71
Figure 4.5: Integration Level among Existing Systems .................................... 73
Figure 4.6: Companies Implemented EDI ..................................................... 75
Figure 4.7: Importance of System Integration by benefits ............................... 77
Figure 5.1: SCAIM Integration System Use Case Diagram ............................ 94
Figure 5.2: Proposed Architecture Design of SCAIM Integration System ........ 98
Figure 5.3: Maintain Systems List Sequence Diagram ................................... 104
Figure 5.4: Maintain Global Services Sequence Diagram ................................ 105
Figure 5.5: Maintain Local Concepts Sequence Diagram .............................. 107
Figure 5.6: Map Concepts Sequence Diagram .............................................. 108
Figure 5.7: Check Outstanding PO Sequence Diagram .................................. 109
Figure 5.8: List Product Satisfaction Sequence Diagram ............................... 111
Figure 5.9: Request for Quotation Sequence Diagram .................................... 112
Figure 5.10: Conceptual Mapping for Outstanding PO Global Service .......... 114
Figure 5.11: Class Package for Conceptual Model ......................................... 117
Figure 5.12: Class Package for Adaptive Service Connector ................................. 117
Figure 5.13: Class Package for Business Component........................................... 118
Figure 5.14: Class Package for Conceptual Mapping and Service Binding .......... 119
Figure 5.15: Entity Relationship Diagram for Service Dictionary....................... 120
Figure 5.16: Sample Input Screen - Register New Local Service of ERP .......... 122
Figure 5.17: Sample output - GRN Details Requested by SCM User ................. 123
Figure 5.18: CM Web Application in view of VS Solution Explorer .................. 124
Figure 5.19: ASC Web Application in view of VS Solution Explorer ................. 127
Figure 5.20: Create Tables and Attributes through SQL Server ....................... 129
Figure 5.21: Service Dictionary ER Diagram in SQL Server Database Diagram . 129
Figure 5.22: Interface of SCM to Invoke a Service of ERP .............................. 130
Figure 5.23: ERP Database in MS Access Environment ..................................... 132
Figure 5.24: CRM Database in SQL Server Management Studio ..................... 133
Figure 5.25: SCM Database in MySQL IDE Environment ................................. 133
Figure 5.26: Interface of ASC for Service Requesting Based on Configuration ... 134
Figure 5.27: Current User and Local Computer Certificate Stores ..................... 146
Figure 5.28: Service Consumer Security Policy Summary ................................. 148
Figure 5.29: Solution Explorer of ERP System Business Component ............... 157
Figure 5.30: UI of CM to Maintain Service Integration Output ......................... 160
Figure 5.31: Integration Output in EDI Format (RECADV)............................... 162
Figure 5.32: Input for Black-Box Testing of a Shared Service ......................... 165
Figure 5.33: Correct Output of Black-Box Testing ........................................... 166
Figure 5.34 (a): Units in SCAIM Participated System Module ......................... 169
Figure 5.34 (b): Module Testing on User Interface ........................................... 169
Figure 5.34 (c): Module Testing on Service Interface, Business and Data
Component ........................................................................................................ 170
Figure 5.35: System Performance on Bar Chart View ........................................... 172
Figure 5.36: Database Performance on Bar Chart View ........................................ 173
Figure 5.37: SQL Response Time Testing Using SQL Profiler .............................. 173
Figure 5.38: Window Server Utility Tool to Perform CPU and RAM Testing .... 174
Figure D.1: Login Screen of CM Module Interface ................................................ 211
Figure D.2: Main Screen of CM Module - System List .......................................... 212
Figure D.3: Register a New Participate System Module ......................................... 212
Figure D.4: Edit a Registered Participated System ................................................ 213
Figure D.5: Remove a Registered System Module ............................................. 213
Figure D.6: Maintain Global Concepts through CM ............................................. 214
Figure D.7: Register a New Global Concept ....................................................... 214
Figure D.8: Edit a Registered Global Concept .................................................... 215
Figure D.9: Remove Registered Global Concepts ............................................... 215
Figure D.10: Maintain Global Services through CM ............................................ 216
Figure D.11: Register a New Global Service ....................................................... 216
Figure D.12: Edit a Registered Global Service .................................................... 217
Figure D.13: Edit WSDL of a Registered Global Service ..................................... 217
Figure D.14: Remove Registered Global Services .............................................. 218
Figure D.15: Login Screen of ASC Module Interface ............................................. 219
Figure D.16: Main Screen of ASC Module - Local Concepts ................................. 220
Figure D.17: Register a New Local System Concept ............................................. 220
Figure D.18: Edit a Registered Local Concept .................................................... 221
Figure D.19: Remove a Registered Local Concept .............................................. 222
Figure D.20: Maintain Local Service through ASC of ERP System ....................... 222
Figure D.21: Register a New Local Service to ASC of ERP System ..................... 223
Figure D.22: Edit a Registered Local Service .................................................... 223
Figure D.23: Remove Registered Local Services .................................................... 224
Figure D.24: Maintain Input and Output Concepts of a Particular Local Service 225
Figure D.25: Local and Global Concept Mapping .................................................. 226
Figure D.26: Local and Global Service Mapping ................................................... 227
Figure D.27: Login Screen of a SCM System.......................................................... 228
Figure D.28: Main Screen of SCM Interface – External Service ......................... 229
Figure D.29: SCM User Invoking an ERP Service - Outstanding PO ............... 230
Figure D.30: Message Box Prompted for Saving Output EDI ............................. 230
Figure D.31: EDI Output File Provided by ERP, Invoked by SCM User ........... 230
LIST OF TABLES

Table 2.1: Hub/Spoke and Bus Architecture Comparison ................................. 38
Table 2.2: Basic Elements of the Conceptual Framework ................................. 47
Table 3.1: Title and Objectives of the Survey Questionnaire ............................. 50
Table 3.2: Sub-Title and Objective of Sections in the Questionnaire ................ 51
Table 3.3: Distribution of Question Type in the Questionnaire ......................... 52
Table 3.4: Criteria and Description for Pilot Test the Questionnaire .................. 54
Table 3.5: Pilot Test for a Multiple Choice Question ........................................ 55
Table 3.6: Most Favorable Answers to Five Criteria by Respondents .................. 56
Table 4.1: Role of Judgment Respondents in Company ....................................... 65
Table 4.2: Implemented Systems in the Companies .......................................... 69
Table 4.3: Current Level of System Integration among the Companies ............... 78
Table 4.4: Preferred Methods of System Integration ......................................... 79
Table 5.1: Sub-Components of Conceptual Model ............................................. 99
Table 5.2: Sub-Component of Adaptive Service Connector ............................... 100
Table 5.3: Sub-Component of System Module ............................................... 101
Table 5.4: Projects to Simulate SCAIM Integration System .............................. 116
Table 5.5: Codes for CM Form Based Authentication ....................................... 125
Table 5.6: Setting web.config File to Support Windows Authentication Mode ....... 126
Table 5.7: SCM System Configuration Setting ............................................... 131
Table 5.8: ERP System Configuration Setting ............................................... 131
Table 5.9: Source Codes to Implement Object Oriented Input Parameter .......... 135
Table 5.10: Code Snippets for Calling Dynamic Proxy Class ............................ 137
Table 5.11: Code Snippets on Building the Proxy Class ................................... 138
Table 5.12: Source Codes on Generating Codes for the Proxy Class ................. 139
Table 5.13: Code Snippets for Compiling Codes Stored in the StringWriters...... 139
Table 5.14: Source Code for Discovering Web Methods........................................ 140
Table 5.15: Source Codes for Invoking Web Method by the Proxy Class............. 141
Table 5.16: Source Codes for Service Interface Object........................................ 142
Table 5.17: Source Codes of Web Service for Outstanding PO Global Service .... 143
Table 5.18: Registered Configuration and SOAP Extension Classes for WSE ..... 144
Table 5.19: Policy File for Service Consumer....................................................... 147
Table 5.20: Code Snippets for Asynchronous Calling Dynamic Proxy Class....... 149
Table 5.21: Partial Codes for Asynchronous Service Proxy Class....................... 150
Table 5.22: Service Message with Message Information header.......................... 151
Table 5.23: Endpoint Reference for Outstanding PO Service ......................... 152
Table 5.24: The TCP Service Requester Code Snippets.................................... 153
Table 5.25: The TCP Service Provider Code Snippets....................................... 154
Table 5.26: The Referral Cache of the Message Router..................................... 156
Table 5.27: Code Snippets of ERP Business Component................................. 158
Table 5.28: Data Access Component to Access ERP Data .................................. 159
Table 5.29: Convert to CSV Output Format ....................................................... 161
Table 5.30 (a): White-Box Testing on Button Click Events ............................. 167
Table 5.30 (b): White-Box Testing on Invoking Dynamic Web Service .......... 167
Table 5.30 (c): White-Box Testing on Calling the Business Component .......... 167
Table 5.30 (d): White-Box Testing on Data Access Component ...................... 168
Table 5.30 (e): White-Box Testing on Auto-Generated SQL Strings............... 168
Table 5.30 (f): White-Box Testing on Retrieving Result Data Set ................... 168
Table 5.31: Integration Testing and Evaluation............................................... 170
Table 5.32: Performance Testing Results ....................................................... 172
Table B.1: Maintain System List Use Case ..................................................... 202
Table B.2: Maintain Local Concepts Use Case .......................................................... 203
Table B.3: Map Concepts Use Case ........................................................................ 204
Table B.4: Check Outstanding PO Use Case ............................................................ 205
Table B.5: List Product Satisfaction Use Case ....................................................... 206
Table B.6: Request for Quotation Use Case ............................................................ 207
Table C.1: SCAIM Tables and Descriptions ............................................................ 208
Table D.1: Minimum Hardware Requirement of User Environment ....................... 210
Table D.2: Minimum Software Requirement of User Environment ....................... 210
Table E.1: Source Code for EDI Output Generation ............................................. 231
Table E.2: Class of Web Service Dynamic Proxy ................................................... 237
# ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMQP</td>
<td>Advance Message Queuing Protocol</td>
</tr>
<tr>
<td>ASC</td>
<td>Adaptive Service Connector</td>
</tr>
<tr>
<td>B2B</td>
<td>Business to business</td>
</tr>
<tr>
<td>BOM</td>
<td>Bill of Material</td>
</tr>
<tr>
<td>CAS</td>
<td>Computer-Aided Selling</td>
</tr>
<tr>
<td>CIO</td>
<td>Chief Information Officer</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Integration</td>
</tr>
<tr>
<td>CPFR</td>
<td>Collaboration planning, forecasting and replenishment</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>CM</td>
<td>Conceptual Model</td>
</tr>
<tr>
<td>CSS</td>
<td>Customer Services and Support</td>
</tr>
<tr>
<td>CTO</td>
<td>Chief Technical Officer</td>
</tr>
<tr>
<td>EAI</td>
<td>Enterprise Application Integration</td>
</tr>
<tr>
<td>EDI</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>EMA</td>
<td>Enterprise Marketing Automation</td>
</tr>
<tr>
<td>EPS</td>
<td>E-Procurement System</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>ESB</td>
<td>Enterprise Service Bus</td>
</tr>
<tr>
<td>ETL</td>
<td>Extract Transform Load</td>
</tr>
<tr>
<td>FFA</td>
<td>Field Force Automation</td>
</tr>
<tr>
<td>IFS</td>
<td>Internet Fulfillment Serve</td>
</tr>
<tr>
<td>JMS</td>
<td>Java Message Service</td>
</tr>
<tr>
<td>LCIM</td>
<td>Level of Conceptual Interoperability Model</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MEPs</td>
<td>Message Exchange Patterns</td>
</tr>
<tr>
<td>MRP</td>
<td>Manufacturing Resource Planning</td>
</tr>
<tr>
<td>MOM</td>
<td>Message-Oriented Middleware</td>
</tr>
<tr>
<td>P&amp;G</td>
<td>Procter &amp; Gamble</td>
</tr>
<tr>
<td>POS</td>
<td>Point of Sales</td>
</tr>
<tr>
<td>QFD</td>
<td>Quality Function Deployment</td>
</tr>
<tr>
<td>SCC</td>
<td>Supply-Chain Council</td>
</tr>
<tr>
<td>SCM</td>
<td>Supply Chain Management</td>
</tr>
<tr>
<td>SCOR</td>
<td>Supply Chain Operations Reference</td>
</tr>
<tr>
<td>SDLC</td>
<td>Software Development Life Cycle</td>
</tr>
<tr>
<td>SDM</td>
<td>Software Development Methodology</td>
</tr>
<tr>
<td>SDS</td>
<td>Sales and Distribution System</td>
</tr>
<tr>
<td>SFA</td>
<td>Sales Force Automation</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SCAIM</td>
<td>Service-Oriented Conceptual Adaptive Integration Methodology</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>WSCI</td>
<td>Web Service Choreography Interface</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Definition Language</td>
</tr>
<tr>
<td>WSMF</td>
<td>Web Service Modeling Framework</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XP</td>
<td>Extreme Programming</td>
</tr>
<tr>
<td>xPATH</td>
<td>XML Path Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>Stylesheet Language Transformations</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
</tbody>
</table>
CHAPTER 1: INTRODUCTION

1.1 Introduction

During the past decades, it has been seen an increasing competitiveness of business environment. Enterprise systems such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Supply Chain Management (SCM) have been constructed as critical success factor for organizations to maintain their competitive advantage. However, each of the system exists in isolation, lack of means for information sharing among each other.

It is very difficult to integrate existing systems components that are not originally designed to be work together. According to Arshah, Desa, and Hussin (2008), stakeholders do not have the knowledge of accessing to the internal workings of the system due to lack of support from experienced colleagues. Hence, many of the legacy systems were developed by using proprietary technologies and most of them are lack of standards like XML (Kuchibhotla, Dunn & Brown, 2009; Cruz & Rajendran, 2003). Thus most of the integration is time consuming that causes prohibitively high development cost for small and medium businesses.

In order to achieve effective interoperability, a flexible approach that denies traditional data-centric integration technologies is selected. This thesis will focus on the Information Integration by sharing functionality among the systems rather than simply sharing data among them. Hence, the premise of this dissertation is that a proposed system integration approach named Service-Oriented Conceptual Adaptive Integration Methodology (SCAIM), which is capable of enabling different systems work cooperatively to share services at the interoperability basis.
The challenge now is to simulate the design of the SCAIM that enables integrating among ERP, CRM and SCM system components effectively. ERP, CRM and SCM are among the enterprise applications that have similar application domains and have the most significant market demand on their integration for the small and medium business environment.

This dissertation focuses on the development and implementation of the SCAIM system to integrate components in the selected systems (ERP, CRM and SCM) that are conceptual equivalence. The concept of conceptual equivalence will be discussed later in chapter two. Hence, the concepts of ERP, CRM and SCM in this thesis are treated separately, while a particular component of these systems are selected respectively in order to implement or simulate a particular integration purpose.

1.2 Background of the Study

ERP was originally derived from Manufacturing Resource Planning (MRP II), which also the extension of Material Requirements Planning (MRP). The paradigm of MRP has been introduced since the last four decades, where the main idea focused on the inventory management of material, plant maintenance, personnel and distribution planning, production and logistic process supports. Then MRP incorporated with finance, accounting, distribution management and human resource functions that contributed to the concept of MRP II (Sheikh, 2003).

The current ERP enhancement is focusing on the web-enabled ERP by using technologies like WAP and World Wide Web. Many of the on-running researches are recently focusing on the extended ERP that beyond the enterprise domain. Furthermore, more and more companies have begin to realize the important of CRM and SCM
package that could be integrated with ERP in order to increase the company productivity and also to increase the collaboration between the company, customers, suppliers and their partners not only internally and externally.

CRM was originally derived from Customer Database Marketing that was introduced in 1980’s. The practices were focused on dealing with company’s customers individually by the customer service group. In 1990’s, Customer Relationship Marketing was first introduced. The methodology was customer loyalty programs, where the companies reward the customers by giving them the incentives, gifts or even money. The methods were good enough for the companies to enhance the customer experiences rather than simply collect data for their own use in previous database marketing (The History of CRM, 2006).

The real Customer Relationship Management concept began in early 2000’s. Unlike the database marketing and relationship marketing, CRM provides new business strategies for companies to increase customer satisfaction from customer order to end product receiving, at the same time to increase their profits. Furthermore, the ways of information processing become more dynamic on the customer reactions and motivations in order to cater the constant change of customers needs in the real world.

Also known as Logistics Network Management, SCM is a set of approaches to enable suppliers to obtain information about the manufacturers, warehouses, stores, wholesale, retailers and consumers. By implements SCM, an organization is able to deliver right quantities services or end products at the right locations, right time, and less costs. (Simchi-Levi, Kaminsky, & Simchi-Levi, 2000).
Many of the experts like Council of Logistic Management argue that Logistic Network different from Supply Chain while others feel the terms can be overlapped. From the enterprise perspectives, SCM is all about the procurement and purchasing while Logistic Network Management focuses on the coordination of information flows and how the software package could help companies in achieving their business strategies. However, the software giants like SAP and Oracle focusing more on supply chain rather than logistics.

It is said that there exist similarities among ERP, CRM and SCM from representational and operational perspective because they are typically fall under very similar application domain or categories. The similarities are prerequisites for system integration success. These systems are designed to aim for increasing competitive advantages of an enterprise.

1.3 Problem Statement

The main problems concerned by this research are related to the solution on system integration issues, and how it helps in improving competitive advantages of the organizations. The core common integration problems are includes:

- **Lack of Semantic Interoperability among Existing Systems**

Several literatures have mentioned that performance and functionality of ERP, CRM and SCM systems has been improved but most of them continue to exist in isolation (Sheikh, 2003; Schellong, 2005; Yawei & Laixi, 2007). Hence, most of the companies had chosen to construct the systems sequentially in a long term period of time, causes the problem on lack of means to share information among each of the system (Arshah, Desa, & Hussin, 2008). In other words, each of the system is
unable to interpret the receiving information accurately from other system by its original meaning.

- **Lack of Functional Interoperability among Existing Systems**

According to Arshah, Desa, and Hussin (2008), the lack of integration between the software systems has caused most of the business processes tend to be done manually. For instance, a Sales and Distribution component of SCM consists of functions like inventory control, sales order entry, purchasing, and supplier scheduling, while a Manufacturing component of ERP include functions such as production control, material requirement planning, bill of material, shop floor control and so on. It is very difficult for production control of ERP to retrieve the quantity of customer purchase order entered through sales order of SCM automatically if the systems are not originally designed to be work together. In other words, work orders have to be created manually in order to proceed with the production and a lot of time has been wasted through the manual process (Sheikh, 2003; Leon, 2000).

- **Insufficiency of ERP in Collaboration with Customer and Supplier**

According to Chang and Makatsoris (2001), “It is very hard for ERP systems to perform real time simulation of adjustments due to constraints in the manufacturing, finance, distribution, warehouse or other operations because the system is mainly concerning on the business transaction processing”. Meanwhile, Robinson and Wilson (2001) stated that customer services must be directly connected to ERP systems in order to react rapidly to customer orders. It is important to generate a platform for direct communication between actual demand to production and planning (Robinson & Wilson, 2001). Consider these researches, it is obvious that
ERP system is not good enough to improve the competitive advantages due to lack of integration between the company, customers and suppliers in the business context. ERP needs to be integrated with CRM and SCM in order to cater with continuous changes on customer demand and the way enterprise doing business.

- **No Open Standard Methodology for Information Exchange**

Existing integration approaches such as MOM and Integration Broker do not support the open standard of exchange data format between ERP, CRM and SCM systems. There must be no misunderstanding for sender and receiver in using any electronic business documents. Even though some of the current approaches are able to map the different elements in the documents by applying business ontology, but in certain level it may limits the extensibility and flexibility in using standards (Guo, 2006). According to Breissinger *et al.* (2000), it is necessary for a company to implement a standard electronic data exchange with its trading partners in order to compete with global business world. Besides, Arshah, Desa, and Hussin (2008) defines one of the main constraints for system integration is the lack of standards such as Web Services and XML that is pre-defined in the integration domain. Considering these opinions, it is necessary to have an open standard integration methodology that compatible with operating system, programming language and hardware.

- **Lack of Interface Tool to Cater System Change**

From technical perspectives, any system integration approach has to cater system change management. According to Arshah, Desa, and Hussin (2008), one of the most common factors to the failure in information system integration is the late response to the changes in user’s requirements. Hence, it is always the factor in
causing the failure of ERP implementation while the developers fail to manage the changes in employees’ needs, or the staff themselves resist to system change (Stefanou, 2000; Themistocleous, Irani, O’Keefe, & Paul, 2001). The research works show that the systems are constantly changing. This is because systems integration involves dynamic process that require dynamic tool or interface which able to handle change in integration participants. Meanwhile, Kuchibhotla, Dunn and Brown (2009) and Cruz and Rajendran (2003) mentioned that most of the companies are still using different data format while performing any data transmission or information sharing. The practices have increased the difficulty to bind together heterogeneous applications to perform business functions. In short, it is necessary to develop a single consensus on the final outcome of interface design to map the data between different systems.

1.4 Objective of the Dissertation

The motivation behind the idea of developing and implementing the Service-Oriented Conceptual Adaptive Integration Methodology (SCAIM) that integrate ERP, CRM and SCM comes from knowledge that the non-prescriptive or service oriented integration mechanism is still lacking in the market. The naming of SCAIM is due to the approach consists of a conceptual model and a connector to integrate the participated systems at conceptual level. The main objective of this research is to design and simulate a service based system that enables interoperability among ERP, CRM and SCM within a company network. Therefore, many of the current commercial integration tools like SAP Exchange Infrastructure, Oracle Enterprise Service Bus and Microsoft BizTalk Server provide less focuses on the service oriented architecture of integration methodology.
The following are the primary objectives of the dissertation:

- Investigate and understand the basic idea of service oriented based methodology that enables interoperability among software system.

- Identify system components of ERP, CRM and SCM that have similar application domain, and define the components to be participated in the research development. The representational and operational characteristic of those ERP, CRM and SCM system components are analyzed.

- Develop a Conceptual Model (CM) and three connectors named Adaptive Service Connector (ASC). CM and ASC represent of basic components to prove the concept of SCAIM. CM organizes all representational and service related of each system component participant in the integration. Each of three ASC component making use of CM in order to facilitate the integrative functionality provided by the ERP, CRM and SCM system respectively.

- Evaluate the capability of the SCAIM methodology implementation in the ERP, CRM and SCM system components integration.

1.5 Expected Outcome

The expected outcome of proposed integration system methodology consists of the following criteria which are mainly used to solve the problems as mentioned in Section 1.3:

- Support for conceptual level capability for attributes mapping between independent applications, in order to solve the semantic interoperability problems among the systems.
• Provide service sharing capability in the sense that every participated system is able to consume any shared service provided by each other, in order to improve the functional interoperability that are lacking among the existing systems.

• Enable unlimited involvements of the company staffs, suppliers and customers in invoking every single resource shared out by any participated enterprise systems. This attribute allows all business process to be integrated and perform real time solution to improve collaboration between business partners.

• Apply independent open standard technologies to support different exchange data format between participated systems. Hence, the methodology is compatible with heterogeneous operating systems, programming languages and hardware implemented by existing enterprise systems.

• Provide single consensus interface for information mapping. Local concepts of each participated system are able to be mapped to the global concepts to increase the integrity of all information as a whole. Besides, the proposed integration methodology supports for every single functionality or requirement change in a participated system locally, with minimum changes are required to the proposed system.

1.6 Scope of the Dissertation

The scope of the research outlines design and implementation of a service oriented system that enables interoperability among ERP, CRM and SCM systems components. A component from each of three systems was chosen to be the participants of the simulation of systems integration, which are Sales and Distribution module of ERP, Customer Services and Support module of CRM, and E-Procurement module of SCM.
Despite other integration mechanism like Message Oriented Middleware (MOM), Extract Transform Load (ETL), and Enterprise Service Bus (ESB), this research was targeted on the Service Oriented Integration methodology in creating two types of research components, namely Conceptual Model (CM) and Adaptive Service Connector (ASC). The details of these two types of research component will be discussed in Chapter 5 System Development.

1.7 Significance of the Dissertation

The dissertation simulates the capability of the service oriented integration methodology, SCAIM in addressing the integration needs in commercial ERP, CRM and SCM systems components. The result of the dissertation can be used as guidelines to the software maintainers, who are forced to work independently with each system. Hence, software integration project managers, systems maintainers, system administrators and system developers are the target audience of the dissertation.

1.8 Brief Research Methodology

The main concern of the study is to gather the facts and information regarding to the needs of system integration and roles of middleware software being applied or applying in the marketplace. In order to achieve the purposes, the study will be comprise of the stages such as detailed literature reviews, planning, scheduling and conducting the study, administering and analyzing the collected data from different resources.

The methods of administration of the study include online and mail survey, and phone and face-to-face interview. Online survey is conducted electronically through internet survey to gather opinions from different kind of stakeholders around different locations.
Hence, the mail survey allows more specific responses in advanced, hence it make follow up become more convenience. Research methods like Phone and interview are used to target on getting responses regarding more complex questions, thus guarantee more truthfully answers from the respondents.

The study attempts to cover those industrial application stakeholders from different companies in Peninsular Malaysia and Singapore for convenience of survey administration. More than 50 companies are invited to participate in the study, and the size of the companies is divided into small (less than 50 employees), medium (50 to 500), and large (more than 500 employees). To ensure the heterogeneity meaning of the study outcome, the invited companies will be selected from multiple types of industry areas such as educational, government, health care, health care, finance, and retail. Meanwhile, 150 questionnaires consist of 35 questions each are distributed to employees of the companies. The target respondents will be consists of heterogeneity professions such as IT consultant, business manager, system engineer, business development, finance, system developer and maintainer, and others.

The estimated period for the study is approximately 3 months to complete, which will be started from November 2008 and ended with January 2009. After the survey is completed, the tasks will involve administering and analyzing gather data. Particular sampling techniques will be applied when analyzing for result of the study, such as Judgment and Random Sampling technique. Details discussion regarding sampling technique will be discussed in chapter 3.
1.9 Organization of the Dissertation

This dissertation comprises of six chapters. The chapters include introduction, literature review, research methodology, system development, and conclusion and implication.

The research starts by discussing the problem domains, objectives, expected outcome, scope, and significances of the research. A brief description regarding the organization of the research is provided. This material is presented in **Chapter 1** Introduction.

**Chapter 2** is focus on the literature review on the generic information of ERP, CRM and SCM, their similarities and relationships. Then, the chapter will also discuss the concepts of similarity and interoperability for integration success. Review will also focus on the proposed system integration approaches, EAI, and the current platform for EAI.

**Chapter 3** explains the methods involved in the research. This chapter will defines what method to be used to capture necessary research data, and what data should be collected. Hence, a system methodology is chosen to define development procedures, techniques, tools, and document aids.

In **Chapter 4**, detailed analysis about the result of data collection is presented in table forms as well as in graphical diagrams. The results captured from the survey questionnaire are analyzed and summary of the findings is discussed from the perspective of Random Sampling and Judgment Sampling techniques.

In **Chapter 5**, focuses are on the integration system design and development activities. The services offered by the system components are identified, while architecture of Conceptual Model and Adaptive Service Connector are designed by using web services
technology. The chapter also describes the system development that involves system
coding activities. The chapter also highlighted the system integration implementation
environments including the software and hardware involved. Test plans are generated in
different test method to ensure error free of the final result of the dissertation.

Conclusions of the implemented research and implication of the research to the
integration stakeholders is discussed in Chapter 6. The chapter ends with a discussion
of the future enhancements issues to the system integration.

1.10 Summary

This chapter has outlined the basic structure of this dissertation which focuses on the
proposed SCAIM system to integrate ERP, CRM and SCM enterprise systems. The
main objectives of the research are to enable enterprise system users to be able to
process their daily business process through the proposed integrated modules. The main
scopes of this research are to create a service oriented methodology system integration,
which called Service-Oriented Conceptual Adaptive Integration Methodology. Sales
and Distribution module of ERP, Customer Services and Support module of CRM, and
E-Procurement module of SCM are chosen as the integration participants, and two main
modules of the developing system integration are Conceptual Model and Adaptive
Services Connector.

This research serves as the integration needs in commercial ERP, CRM and SCM
system components. Hence, target audience of the system integration will be software
integration project managers, systems maintainers, system administrators, system
developers, and business process planners.
CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Literature review is carried to review the critical points of current knowledge regarding the research area. It explains the justification of the future works in the research area based on the finding of the literature review. According to Cooper (1998), a literature review seeks to describe, summarize, evaluate, clarify and/or integrate the content of primary reports or original scholarship.

The main focus of this chapter is to justify the prerequisites in integrating three core business systems - ERP, CRM and SCM. At first, an illustration on the general review of three participated system of the integration approach - ERP, CRM and SCM respectively, and the similarities between the systems that support the integration. The interoperability among the participants is also discussed.

Next, the discussion of this chapter focuses on the understanding and critical points of current knowledge regarding the selected approach - Enterprise Application Integration (EAI). EAI acts as a preference and basic architecture of the proposed integration system, the Service-Oriented Conceptual Adaptive Integration Methodology (SCAIM). Finally, the current integration platforms for EAI are also described.
2.2 Review on ERP, CRM, and SCM

The main focus of this section is on major aspect of three most common enterprise applications - enterprise resource planning (ERP), customer relationship management (CRM) and supply chain management (SCM) that companies use to gain competitive advantages. In addition to emphasize why enterprise applications are chosen to simulate the integration approach, the research will also present the similarities that exist among the applications in terms of their architecture and basic concepts.

2.2.1 Enterprise Resource Planning (ERP)

According to APICS Dictionary, Enterprise Resource Planning (ERP) is defined as accounting based information systems that mainly use for identifying and planning the resources within a manufacturing, distribution, or service organization, which needed to take, make, ship, and account for customer orders (Leon, 2000).

According a study done by AMR Research (2006), top five ERP vendors in year 2005 are SAP, Oracle, Sage Group, Microsoft and SSA Global, who contributed 72 percents of total revenue worldwide.

2.2.1.1 ERP Objectives

ERP systems are in a great demand during this decade. According to Bakry and Bakry (2005), the objective of an ERP system is to automate the business process of the enterprise, in order to lead better enterprise performance by supporting e-business implementation concepts. The philosophy behind the reasons is the ERP objectives itself, where these objectives are the reasons why ERP is chosen for the integration. Based on explanations by Sheikh (2003), Leon (2000), Bakry and Bakry (2005),
Tianshe *et al.* (2006), Jen-Her and Yuh-Min (2003), and, Chia-Chia and Dong-Her (2009), the most common purposes of ERP implementation include the following:

- To overcome the pitfalls in legacy systems (Leon, 2000; Tianshe *et al.*, 2006).
- To shorten accessing time to integrated internal information (Chia-Chia & Dong-Her, 2009; Jen-Her & Yuh-Min, 2003).
- To support business growth through merging and acquisition (Chia-Chia & Dong-Her, 2009; Leon, 2000).

### 2.2.1.2 ERP System Architecture

The ERP architecture system is of ‘modular’ construction and dynamic structure, which provides scalability for itself to be modified and expanded easily (Bakry & Bakry, 2005). From operational perspective, all ERP modules are working as an integrated information system that delivers a single database. However, some of common modules are selected to be discussed here in order to have better understanding on the architecture of ERP. The general modules of ERP are depicted in Figure 2.1.

![Figure 2.1: Common ERP Modules (Leon, 2000)](image-url)
Manufacturing
Previous section has discussed about ERP was originally derived from MRP system. The main idea of MRP attempted was to create in-house integrated systems in manufacturing industry. Until today, the Manufacturing modules are still the main focuses of most ERP packages for increasing the effectiveness of the internal enterprise operations.

Sales and Distribution
In order to increase collaboration between the company with the customer and supplier, and increase efficiency of entire supply chain, the Sales and Distribution play important roles. Typical business process that requires Sales and Distribution modules include Sales, Shipping and Billing. Hence, Inventory Control, Purchase Order, and Sales Order Systems are among the common sub-system that support Sales and Distribution process.

Financial
Financial Accounting modules of ERP system provide functionalities to centrally tracking the financial information within an internal framework of multiple companies, language, currencies, and charts of accounts (Leon, 2000). Generally, General Ledger, Accounts Payable, and Accounts Receivable sub-systems construct a financial module in ERP system.

Other Supportive Systems
Apart from Manufacturing, Financial, and Sales and Distribution modules, typical ERP systems include Human Resource Management, Quality Management, and Plant Maintenance.
2.2.2 Customer Relationship Management (CRM)

Customer Relationship Management (CRM) is an industry term used for software and methodology to manage customers interests, behaviors, and needs systematically in order to development of strong relationships between the company and their customers. In other words, “CRM is defined as an approach used by the companies through communication with clients in order to retain existing customer, increase customer loyalty, and to cater their benefits” (Fan et al., 2004). Customer information is personalized and stored into the system for marketing purposes.

Currently in the global CRM marketplace, the market leaders of Customer Management applications are SAP, Oracle, and salesforce.com (AMR Research, 2006).

2.2.2.1 CRM Objectives

In short, CRM system connects different parts of a company through the management of customer relationship. According to Fjermestad and Romano (2003), Kim, Lee, Lee and Chun (2003), Schellong (2005), and Fan et al. (2004), the common core purposes of CRM software include the following:

- Discover new customers, better understanding of existing customers’ needs, and improve the royalty of customers to the company or products (Fan et al., 2004; Fjermestad & Romano, 2003; Schellong, 2005).

- To optimize customer life cycle management and improve relationship between customer and the company (Fan et al., 2004; Schellong, 2005).

- To increase organization revenue (Kim, Lee, Lee & Chun, 2003; Fan et al., 2004).
2.2.2.2 CRM System Architecture

Unlike ERP, the architecture of CRM does not concern about the Operation or purely integrated information computer system, but it concerns about Analytic, and Collaborative perspective on the relationship between customers and the company (Che-Wei, Cheng-Ru, & Chia-Chun, 2009; Schellong, 2005). This is because CRM is defined as philosophy, company’s policy and business strategic to maximizing revenue and customer satisfaction.

Operational

As according to Schellong (2005), Operation CRM streamlined the processes in the back and front offices business (sales, marketing and services) of the company. Several common CRM modules in the marketplace include Computer-Aided Selling (CAS), Sales Force Automation (SFA), Customer Services and Support (CSS), Enterprise Marketing Automation (EMA), and Field Force Automation (FFA).

CAS provides source of information for all areas of sales and distribution process, and acts as a tool to generate, qualify and track every lead to closure. Company’s critical sales and sales force management tasks are automated under the SFA module. CSS provides functionalities to handle services request, customer complaints, and enquiries. On the other hand, EMA is used to accumulate information regarding trends, business environment, competitors and potential customers for analysis to improve marketing efficiency. FFA is aimed to optimize the process and information provided to the company in assignment of labor to individual work order, at the same time to improve visibility and service to customers at low service cost.
Collaborative

Collaborative CRM is related to the combinations of different communication channels or the ways to keep a single view on the customer and offer consistent customer experience across channels (Schellong, 2005). The communication channels mentioned here could be online services, call center, Point of Sales (POS), counter, or sales force. Channel such as Online Services are very efficient and attractive in managing the potential offer and reduce the enterprise’s expenses of sales, service, and marketing. According to Che-Wei, Cheng-Ru, and Chia-Chun (2009), the target of collaborative CRM is to reduce company operational cost, hence maintain good relationship with the customers.

Analytical

The Analytical CRM emphasizes on the using of OLAP or data mining technology to interpret the accumulated customer data means to enhance the relationship between customer and the company. Hence, the core concepts of Analytical CRM are Acquisition, Retention, and Information. Acquisition means to encourage cross-selling and up-selling, Retention to retain existing customer, while Information providing up-to-dated and accurate information regarding product and services to the customers.

2.2.3 Supply Chain Management (SCM)

Supply chain is known as business processes of producing products or services, which involves supplier and customer. The components of the processes are including supply, manufacturing, sales and distribution (Leon, 2000, p.474). Inherit from the definition, the purpose of having SCM is to satisfy customer demands by planning, designing, and managing the operation of supply chain. It provides the functions of coordinating the processes of creating raw material, work in progress and finish goods.
In the SCM marketplace, Oracle still maintain the top SCM vendor in terms of revenue, while SAP, Accruent, American Express, Dun & Bradstreet (D&B) round out the second to five place respectively in terms of their revenue in year 2006 (AMR Research, 2006).

2.2.3.1 SCM Objectives

The key advantage of SCM is to create value for customer and increase profitability for the supply chain company. According to researchers, the main reasons for companies to choose implementing SCM include:

- To increase inventory turnover of a company (Yawei & Laixi, 2007).
- To increase the productivity and efficiency of manufacturing (Pattnaik, Sutar, & Govindan, 2009).
- To encourage collaboration and build relationships between suppliers, enterprise, partners, and the customers (Bo & Lei, 2008).

2.2.3.2 SCM System Architecture

While discussing about the operational architecture of SCM, concept of Supply Chain Operations Reference (SCOR) model is referred. SCOR introduced by Supply-Chain Council (SCC), which include five basic components of SCM are Plan, Source, Make, Delivery, and Return (Francis, 2007).

From the operational perspective, general functionalities of SCM are Demand Planning, Procurement, Manufacturing, and Distribution, and Performance Measurement. These five functionalities are directly related to the concepts of Plan, Source, Make, Delivery, and Return respectively.
2.3 Integration Prerequisite - Similarity

The selected enterprise application - ERP, CRM and SCM for the integration typically fall under very similar application categories or application domain. At a conceptual level, they overlap in some level in terms of their operational and representational characteristic. Although these three systems may be implemented in different ways and it may operate under different technologies, but they representing the same concept (conceptual equivalence). This is the main factor to make sure the integration success.

The objectives, architectures and definition of ERP, CRM and SCM are discussed. The following sections are focuses on two concepts - Representation and Operational Perspective. These concepts demonstrate the prerequisite for the integration success. Hence, the similarities between ERP, CRM and SCM will be discussed in the Section 2.3.4 and 2.3.8 as supportive to the necessity for them to be integrated in this research.

![Figure 2.2: Conceptual Pre-requisites for Integration Success](image-url)
2.3.1 Representation Perspective

The discussions of the three systems objectives depict the representation perspective to the integration success. The main concern is to find out the similar domain that exists among ERP, CRM and SCM systems that can prove the systems are conceptual equivalence from the representation perspective. Conceptual equivalence among the enterprise systems means systems operate at the same or very similar application domain although they have different functionality and vocabulary. For those systems to be worked as conceptual equivalence or domain application equivalence, Semantic Interoperability among the systems to be integrated could be achieved.

2.3.2 Conceptual Equivalence

The concepts of ERP, CRM and SCM represented here are important for semantic mapping. Semantic mapping is the process of translating two or more systems that seem to have different objectives but conceptually they represent similar concept. Referring to Park (2002), it is possible for two different systems to have the same concept or very similar concepts. Semantic ambiguity between the source and target system objectives can be reduced by utilizing information stored in a database that link to a Conceptual Model and tools like syntax rules. In later chapter, the proposed conceptual-based model will be demonstrated to facilitate the purpose of semantic interoperability among the three enterprise applications.
2.3.3 Semantic Interoperability

Semantic interoperability is the ability for two or more computer systems hardware or software to exchange information whiles each of the system is able to interpret the receiving information accurately by its original meaning. In this research, this is the target to enable ERP, CRM and SCM to sharing their data, information and service to each other at a common understanding vocabulary, and all request information will be interpreted correctly in order to achieve perfect semantic interoperability.

To analyze the philosophy of Semantic Interoperability in details, Level of Conceptual Interoperability Model (LCIM) is concerned (Tolk & Muguira, 2003). Of seven levels of Extended LCIM model, Semantic Interoperability is located at the forth level (Level 3) of capability for interoperation. LCIM was first introduced by (Tolk & Muguira, 2003). There are five levels of interoperability, which are System Specific Data (Level 0), Documented Data (Level 1), Aligned Static Data (Level 2), Aligned Dynamical Data (Level 3), and finally is Harmonized Data (Level 4).

Level 2 of traditional LCIM, Aligned Static Data, is the subset concept of Semantic Interoperability. In this level, data (does not include information and services) of the systems are documented using a common reference model based on a common ontology. Hence, the meaning of the data is unambiguous described.
However, Turnitsa (2005) improved this original LCIM and came out with current version of LCIM - Extended LCIM. This version of LCIM emphasizes on the interoperability issue rather than simply data interchange in original LCIM. The seven levels of Extended LCIM are No Interoperability (Level 0), Technical Interoperability (Level 1), Syntactic Interoperability (Level 2), Semantic Interoperability (Level 3), Pragmatic Interoperability (Level 4), Dynamic Interoperability (Level 5), and Conceptual Interoperability (Level 6).

Obviously, level 3 of this model is the target of the integration. According to Turnitsa (2005), the level of Semantic Interoperability is achieved when a common information exchange reference model is applied. Turnitsa and Tolk (2008) related integrated information (data with semantic indicator) among systems as requirements to achieve semantic interoperability level of extended LCIM. This research has used term of Conceptual Model (CM) to demonstrate the reference model.

However, not all application domains require the same degree of alignment of the extended LCIM model. The higher level of extended LCIM like Conceptual Interoperability (Level 6) is not the target of this research. The reason is because there are no necessity for the systems participates in the integration of this research to be documented based on engineering methods, which is pre-requisite of the Conceptual Interoperability (Turnitsa & Tolk, 2008).
2.3.4 Representation Concerns for ERP, CRM and SCM Systems

Integration

Three of them are common enterprise applications in the marketplace, and obviously, they are created to aim for increasing the competitive advantages of a company. From this point of view, there must be similarities exist between them in terms of the concepts they represented.

ERP and SCM applications have been known as the back office systems to support the daily processing and supply chain within the enterprise. From the objectives of ERP that discussed earlier, ERP focuses on managing the operation of an enterprise and provide short-term planning. The enterprise long-term planning relies on the SCM system which is able to derive the raw data recorded by ERP to make forecast on the company internal processing.

Since ERP is aimed to decrease the accessing time to the internal integrated information of enterprises, it does support the effective SCM by sharing the information across the supply chain members in order to help management make right decision in the emerging market scenario.

ERP supports the SCM systems’ objective to satisfy customer requirements and this advantage of ERP has become more obvious when it is web-enabled. For instance, customers are able to access the suppliers’ ERP systems via internet browser interface to check the real-time information regarding the shipping status, inventory, and pricing. The integration between ERP and SCM enables supply chain member to provide fast response to the customer orders and satisfies the customers simultaneously.
In common, the integration between ERP and CRM can be defined as a medium to collaborate front-end and back office operation of an organization that applied them. They enable customers and business partner to be included into value chain inward to and outward from the organization, and encourage the collaboration between companies. Extended Enterprise is a concept to integrate and consolidate all business resources for improved management and operations within a global competitive environment. Figure 2.3 illustrate the mentioned concept of Extended Enterprise.

![Extended Enterprise Diagram](image-url)

**Figure 2.3: Extended Enterprise (Bąkowski, 2002)**

Another reason for integrating CRM and ERP is because it can support the aims of CRM, which are to optimize customer life cycle management. In the computer assembly business world, for example, work order is generated by ERP system after a customer orders for certain specification (through CRM at the front end), and then ERP processes the bill of material with specific routing process. Without the intelligent processing of ERP systems from sales and marketing, purchasing, manufacturing, accounting and finally to distribution, time to delivery for the build-to-order product will be increased. CRM and SCM help company in managing daily business planning. CRM system deals with customer requirements while SCM supports the company sales and demand.
planning effectively according to collected data from CRM. The integration between CRM and SCM has become more obvious when some companies use CRM to support multi-channels customer support with integrated supply chain planning.

From early discussion, CRM and SCM system both aim to deliver revenue to the company. This vision can be achieved by having integration between these two enterprise systems. CRM collects the demand chain and opportunities in the market and SCM uses the collected data and translate it into a real-time supply chain planning. By the capabilities of CRM to collect customer requirements and SCM supply planning capabilities, the integration has further helped the company to manage customer expectations effectively. It has also improved the capabilities of supply network of the company.

2.3.5 Operational Perspective

The factor of integration success is throughout the Operation Perspective. In this section, ERP, CRM and SCM system architecture concept that discussed in the previous section, will be inherited to prove that there are possibilities of functional interoperability among three systems.

2.3.6 Representation & Execution Knowledge

Representation knowledge is regarding the knowledge of accessing the internal working of the systems. One of the barriers to integrate enterprise systems is that many of them can be viewed as a black box where the systems vendor tend to keep the internal operations of system without the knowledge the company. Representation knowledge provides perception into the perspectives and goals of other participants involved in a
development process (Booch, 1994). This means that all system maintainers participant in the integration should have knowledge of each other system’s operations.

Besides the representation knowledge, stakeholders and maintainers are also required to have the execution knowledge. According to von Mayrhauser and Vans (1995), most often the system maintainers focus only on understanding the system portion that is relevant to the change being introduced. Consequently, the maintainers will never understand the overall architecture of the systems that is being integrated. For the reason, any integration method should be able to provide maintainers with the mechanisms that control the execution of ERP, CRM and SCM systems, where the information is gathered from the systems stakeholders.

2.3.7 Functional Interoperability

Mastering representation and execution knowledge are the main factors to achieve functional interoperability. Functional Interoperability involves the ways of finding shared services of different systems in a collaborative environment. In this research, a conceptual-based model will be introduced to enable ERP, CRM and SCM components achieve functional interoperability without hindered by the discrepancy of language, interface and execution platform between three enterprise systems.

Researchers divided functional interoperability into different levels, such as Signature, Protocol, Semantic, Context and Quality Level (Fang, Hu, & Han, 2004; Strang & Linnhoff-Popien, 2003; Vallecillo, Hernandez, & Troya, 2004). As prerequisite to the functional interoperability, execution and representation knowledge of stakeholder and IT maintainer of the systems to be integrated could be improved by applying proper Business IS strategy, IS/IT Management strategy, and IT strategy during the integrating
process. Hence, the five levels of functional interoperability mentioned are close related to these interoperability strategies. Figure 2.4 represents the relationship between five levels of functional interoperability and three strategies with some examples technology or constraint of each interoperability level.

![Figure 2.4: Functional Interoperability Stack (Fang et al., 2004; Strang & Linnhoff-Popien, 2003; Vallecillo et al., 2004)](image)

Business IS strategy means interoperability between two or more system units or functions in achieving its business objective. IS/IT Management strategy focuses on the common IS/IT elements used in the extended enterprise (customers, suppliers, company, partners and so on) to apply standard and consistent policies. Thus, IT strategy emphasize on the ability of integration participants to be interoperates underlying certain compatible technology, such as execution platform, interface specification, and languages. This is the basic requirement of interoperability as stated by Wegner (1996). By applying these elements, the integration developers will be able to use compatible data structure and functionality to make ERP, CRM and SCM systems to be more interoperability from the operation perspective.
Out of the five levels of Functional Interoperability mentioned, Signature, Semantic, and Quality level are the main concern of the integration considerations. **Signature level** is about the conformance of interface definition, registration, and execution among the shared services in the integration (Fang *et al.*, 2004). This research uses Web Service Definition Language (WSDL) as an effort to resolve the functional interoperability problem at Signature level in the enterprise systems integration.

“**Semantic level** of functional interoperability addresses the issues of different semantic interpretation between service provider and consumer” (Fang *et al.*, 2004, p.154). For instance, Customer Service and Support module of CRM system request for the customer’s purchase status from Sales and Distribution module of ERP. Without a semantic service description framework, ERP and CRM are unable to interoperate to satisfy customer requests. Besides, only functional aspects of semantic level will be focused in this research. Web Service Modeling Framework (WSMF) is a modeling framework of different language variants to describe semantic web services (Lausen, Bruijn, Polleres, & Fensel, 2005).

**Quality level** emphasizes on the consistency between the quality properties provided by a service provider and the quality requirements requested by a service requester. Quality constraints could be the service response time, cost of service, and functional constraints. This level will be taken into consideration of system analysis of the integration model.
Protocol level emphasizes on the relative sequences of service operations when two services interact. To integrate the systems using SOA, Web Service Choreography Interface (WSCI) is chosen as the technology related to this level of interoperability. “From the perspective of message flow, WSCI is used to describe the attribute of web services” (Arkin et al., 2002). Context level will not be considered in this dissertation due to its focuses particularly in ubiquitous computing environments.

2.3.8 Operational Concerns for ERP, CRM and SCM Systems

Integration

After having done a deep inspection of architecture of ERP, CRM and SCM systems in the previous sections, it can be concluded that there are similarities among three systems from the operational perspective, which are possible to make them integrated and work as integrated automation processes.

From the system operation perspective, ERP and SCM systems are able to increase the capability of supply chain collaboration. As refer to a case study about information partnership between a manufacturer, Procter & Gamble (P&G), and a retailer, Wal-Mart. According to Grean and Shaw (2003), two related giant industries implemented software to sharing data across their mutual supply chains. Whenever the P&G products are running low at the distribution centre, the system will automatically send alerts to P&G in order to ship more products (Grean & Shaw, 2003). As an implication, the Demand Planning module of SCM could be integrated perfectly with Sales and Distribution module of ERP in terms of their supply chain efficiency.
For CRM and ERP, integration between the systems provides valuable information to the company units (manufacturing, designing) regarding the customer specified needs. One example tool to capture customer’s requirements and convert them into information called Quality Function Deployment (QFD), which is aimed to enable manufacturing product or service quality fulfill the actual user requirements (Chaudhuri & Bhattacharyya, 2005). From functionality point of view, the customer demands information will be input through Customer Service and Support module of CRM system and passing into manufacturing module of ERP system.

SCM and CRM operational functions are also able to interact among each as a competitive advantage to the company. The multi-channel capability of mySAP CRM allows customers to configure their orders through multiple interaction channels in order to contribute to the effectiveness of supply chain planning process. This example provides that Sales Force Automation module of CRM is able to drive company sales forecasting into Demand Planning module in SCM system. The operational functions mentioned above are just some of the similarities that exist among three systems. More collaboration operation services between ERP, CRM and SCM will be explained in System Development (Chapter 5) of this research.

2.4 Chosen Approach - Enterprise Application Integration

Enterprise Application Integration (EAI) approach was first introduced in the mid-1990s. According to Gartner Group, EAI is the unrestricted sharing of data and business process among any connected applications and data sources in the enterprise (Worldwide Industry Study, 2001). Gleghorn (2005) and Lee, Siau, and Hong (2003) define EAI as a business computing term for plans, method, and tools that are aimed at modernizing, consolidating, and coordinating the overall computer functionality in an
enterprise. From these definitions, it can be determined that EAI is the key approach to incorporate existing systems of a single organization into a broader application context.

ERP, CRM and SCM usually cannot communicate to one and another in terms of their data sharing and the business logic. Among the reasons include:

- Heterogeneity of the programming language
- Diversity of application platform
- Complexity of legacy system in terms of their business logic and requirements
- Incompatible applications interface

For these reasons, EAI plays an important role to link the applications within the organization in order to achieve objectives such as:

- Data and information integration within a company network
- To provide single consistent integration interfaces for applications interaction purposes
- To enable business logic sharing among each application to another
- To centralize business policies and rules to encourage vendor independent among each of the integration participants

In this research, the proposed service oriented methodology, SCAIM, which is a promising distributed computing paradigm that will be an enhanced architecture for EAI approach. In addition, Service Oriented Architecture (SOA) is chosen for SCAIM methodology to implement concepts of EAI approach because of its suitability to integrate enterprise application.
2.4.1 Architecture of EAI

During integration processes, the source and the target system in the traditional EAI architecture are required to be modified, where all these modification normally involve a lot of cost and time. For instance to integrate a custom Logistic and Distribute system of SCM (source system) running on a Window Vista with a custom Inventory Control system of ERP (target system) running on a mainframe, a point-to-point message-queuing middleware software is selected by the integration developers in order to share information between two systems. Due to the point-to-point middleware layer that only consists as a program interface, both systems are required to be modified on the system communication interfaces in order to accommodate the middleware.

Because of the limitations of traditional EAI, the following and current version of EAI architecture represents a different model of enterprise integration than the traditional version in terms of their flexibility. According to Gorton, Thurman, and Thomson (2003), the generic EAI architecture consists of technological components like User Application, Orchestration Layer, Routing/Transformation Layer, Transport Layer, Adapters, and the Data.

After having done a deep inspection on the current research works on EAI, many researches proposed another two basic architectures of EAI, there are Hub/Spoke Architecture and Bus Architecture (Gleghorn, 2005; Joseph, 1999).
Hub/Spoke Architecture

Basically this kind of architecture consists of a centralized broker or Hub and adapters or Spoke. Each participant application is connected to an adapter and the adapter is connecting to the centralized broker. As similar to the Generic EAI architecture, adapters are used to convert the data format to the format acceptable by the centralized broker and the application. On the other hand, the centralized broker is responsible for the transformation and routing the incoming and outgoing message for a particular purpose.

Bus Architecture

Compare to the Hub/Spoke architecture, bus architecture also has a centralized component called central messaging backbone which is mainly used for message propagation. However, unlike the Hub/Spoke architecture, the integration engine is not centralized, but embedded at each adapter of each application in the integration. Same as Hub/Spoke, the application sends the request message through adapter and transform within the integration engine. Each integration engine in the adapter is responsible only for the message from and to its own connected application.

2.4.2 Advantages of EAI

The most obvious advantage of current version of EAI is that it focuses on the integration of business processes, not only data integration like the traditional version. EAI has the capability to streamline the different business processes within the organization throughout the new and existing applications to increase the business process efficiency. With the ability to reuse the resources between each other applications in the integration, both architectures of EAI maintain the integrity of information across the applications. On the other hand, EAI allows non-experts to
develop and maintain the integration of the applications in order to achieve business purposes (Chen, Yin, Jin, Li, & Dong, 2007).

2.4.3 Challenges and Pitfalls of EAI

For Hub/Spoke architecture, it is more difficult for maintainers to expand the functionality of the integration engine in terms of the transformation and translation of messages. It is because the architecture emphasizes on the centralized hub to control the main process of integration. Meanwhile, the complexity of applications in the bus architecture affects the difficulty of development of integration. This is because the architecture itself emphasizes on the independent of the operations of every application with its respective integration engine. Maintainers are required to have the representation and execution knowledge of each application of the integration while upgrading the integration engines.

Compare to Bus architecture, Hub/Spoke is more similar to the proposed architecture, SCAIM, due to supporting the concepts of conceptual equivalent and semantic interoperability as discussed in the previous section, which are the core objectives of the research integration. To compare SCAIM architecture to the Hub/Spoke, the proposed Conceptual Model (CM) is correspondent to the centralized hub, whereas the concept of Adaptive Service Connector (ASC) is closely similar to the adapter of each participant applications.
2.4.4 Comparison of Hub/Spoke and Bus Architecture

Table 2.1 depicts the summary on the comparison for both types of EAI architecture.

Table 2.1: Hub/Spoke and Bus Architecture Comparison (Gleghorn, 2005; Joseph, 1999; Lee, Siau, & Hong, 2003)

<table>
<thead>
<tr>
<th></th>
<th>Hub/Spoke Architecture</th>
<th>Bus Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Development and maintenance</strong></td>
<td>Ease to develop and maintain due to its centralized hub architecture.</td>
<td>Due to its distributed integration engine architecture, complexity of development and maintenance highly depending on the complexity of participant applications.</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>More limited compare to Bus architecture.</td>
<td>More flexible because of the integration engine and adapter was built on distributed system concept.</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td>Ease to implement.</td>
<td>Moderate efforts were needed depending on separate application complexity.</td>
</tr>
</tbody>
</table>

2.5 Current Integration Platforms for EAI

There are a lot of integration platforms in marketplace that implement the architecture of EAI as discussed in the previous section. Integration platform basically is a set of technologies or architecture that is used to implement complex, bi-direction and business to business integration. Among the common key integration platforms that facilitate EAI approach are such as Enterprise Service Bus (ESB), Message Oriented Middleware (MOM) and Integration Broker.

2.5.1 Enterprise Service Bus

Enterprise Service Bus (ESB) is a standard integration platform that exploits web services, messaging middleware, intelligent routing, protocol transformation, and service mapping in a distributed environment to connect and coordinate the interaction of diverse system applications (Luo, Goldshlager, & Zhang, 2005; Vallecillo et al., 2004). In an integration process, ESB supports service request and response communication between the service provider and the service consumer via a service bus
(messaging backbone). Hence, it also support more-complex message exchange patterns (MEPs) via the event-driven and standard based bus.

There are a lot of arguments on whether ESB is a software, architecture, or platform for integration, and how would it is related to EAI. However, Service Oriented Architecture (SOA) can be considered as the best term to answer these arguments. According to Chen, Yin, Jin, Li, and Dong (2007), SOA can be defined as a promising distributed computing paradigm to have independent software services discoverable to all in an enterprise or authorized network. According to the definition, SOA is supposed to be the best architecture for EAI solution, where ESB provides the features to SOA by supporting messaging, services and event-based interactions among the systems in a heterogeneous environment.

2.5.1.1 ESB Architecture

Another comparison between ESB and EAI is that the architecture of ESB architecture is similar to the Bus Architecture of EAI, but with more advanced enhancements. As compare to normal web application client server architecture, ESB is typically implemented in five tiers distributed architecture. The five tiers of ESB topology are application interface (web browsers for example), application servers (act as web service clients), enterprise service bus, web services providers, and database servers. However in physical, the web application server can be as the web service provider. Figure 2.5 depicts the five tiers topology of ESB architecture.
2.5.1.2 ESB Benefits

The key benefits to implement integration using ESB are:

- Avoiding failure caused by single centralized broker since ESB using distributed messaging services and this encourage zero downtime of service availability.

- Less development time and cost since more configuration than integration construction in ESB implementation.

- One time application integration configuration to provide ready for reuse service type, this enable same services used in different purposes.

- Decentralized architecture of ESB increase scalability of whole integration domain with adding new applications.

Figure 2.5: ESB Five Tiers Topology (Chen, Yin, Jin, Li, & Dong, 2007)
2.5.1.3 ESB Disadvantages

There are some other drawbacks for the bus, which are:

- Since ESB implementation requires more configuration than coding, it usually involves more hardware to handle processing.
- Learning curves of developers on ESB take more time than simple point-to-point messaging because new skill and knowledge required configuring ESB.
- Service Management component is able to control service version, but required ongoing management to avoid tight coupling among the services.

2.5.2 Message Oriented Middleware

Message Oriented Middleware (MOM) can be defined as client/server infrastructure that operates on the concepts of passing and queuing messages, where message-passing is non-blocking. In another words, MOM effectively increases interoperability, portability, flexibility and scalability between applications by allowing the applications to be distributed over heterogeneous network infrastructure, various implementation platform, and diverse programming languages (Bernstein, 1996). As compared to ESB, Hart (2003) defines MOM as software to be installed in both portion of client and server workstation, which provides transparent communication services through asynchronous calls or publish-subscribe between the client and the server systems.

2.5.2.1 Architecture of MOM

Typically, MOM is implemented in the client/server and peer-to-peer architecture with supporting asynchronous messaging calls throughout the applications. In an EAI environment, MOM architecture is similar to EAI bus architecture where an application could be service a consumer or service provider and MOM supports peer-to-peer communications between applications. Hence, MOM typically asynchronous because
message queues in MOM architecture provides temporary storage while the service provider is busy or not available.

From the operations perspectives, any passing-messages will be passed through a MOM message broker. The MOM messaging client and server are communicating by using operating system, transport and network layer in their respective application domain, but without being linked by a private, dedicated and logical connection (Jung, Paek, & Kim, 1999). Figure 2.6 depicts the typical MOM system architecture.

![Figure 2.6: A Typical MOM Architecture (Jung, Paek, & Kim, 1999)](image)

### 2.5.2.2 Advantages of MOM

The primary advantages of MOM are such as:

- The service consumer and provider are not required to connect to the network at the same time, and this increase the flexibility and reliability of the applications being integrated.
Typically MOM provides built-in transformation mechanism to transform the received messages to be the application server’s native formats. That means the message sender need not convert the message format before the transmission.

2.5.2.3 MOM Limitations

However, MOM has some critical limitations, which are:

- MOM may incompatible with other MOM implementation due to the use of application specific or proprietary messaging structure in a single MOM implementation. This could affect the flexibility, interoperability, and portability when a new application joins into the integration domain.

- The tendency asynchronous architecture of MOM may not fit some synchronous based inter-application communications. This is because some application requires immediate response from the opponents before the next process to be started.

2.5.3 Integration Broker

Integration Broker is an end-to-end messaging middleware integration platform built for complex EAI architecture. The main objective of Integration Broker is to enable business processes to be fully automated in between supplier, customer and the company (Apshankar, Clark, Hanson, Mittal, & Myerson, 2002). This definition can be proved when it has initially created to integrate various extended enterprise applications like CRM, human resource management system, supplier’s systems, and other internal legacy systems. The leading market Integration Broker packages include IBM MQSeries Integrator, webMethods, SeeBeyond, Microsoft BizTalk Server and Java Message Service (JMS).
2.5.3.1 Integration Broker Architecture

Integration Broker platform is categorized as type of Hub/Spoke architecture of EAI. An integration broker acts as the centralized hub in the architecture for complex component-to-component interactions. In other words, it basically extracts the data or services from the source component, transform and transfer it under standard transport protocols and through intelligent routes to the target component. As similar to MOM, communication between integration broker and the components are appeared to be in the form of messages. Figure 2.7 depicts the typical structure of Integration Broker that involved in the communication processes.

![Figure 2.7: Extended Enterprise Integration with Integration Broker (Apshankar, Clark, Hanson, Mittal, & Myerson, 2002)](image-url)
2.5.3.2 Integration Broker Advantages

The key benefits of Integration Broker over other integration platform include:

- The hub/spoke architecture of Integration Broker provide more manageable, cheaper infrastructure that eliminates the traditional time consuming point-to-point integration.
- Auditing, development and monitoring elements of Integration Brokers provides flexible and easy to manage configuration manual to the maintainers.

2.5.3.3 Integration Broker Disadvantages

However, there are some limitations of Integration Broker, such as:

- Implementation of multiple protocols supported integration broker may requires additional licensing of the integration software and hence additional hardware is needed to operate the integration server.
- Specialized knowledge is needed to deploy and monitor the integration server.

2.5.4 Comparison of ESB, MOM, and Integration Broker

The integration platforms as discussed in previous have certain level of suitability to meet the EAI requirements on:

- **Architecture** - ESB, MOM and Integration Broker are basically inheriting from the EAI architecture of Bus or Hub/Spoke, and most of them are implemented as middleware products.
- **Integration Connectivity** - No matter the platform is based on bus or hub/spoke architecture, the connectivity among the systems is throughout a set of connectors or adapters or connectors.
- **Standard** - According to EAI laws, there are no universal standards for software system integration. This concept is practically true when ESB, MOM and Integration Broker platforms are free to implement specific communication standards during integration processes. For example, all three platforms are supporting for multiple transport protocols like HTTP, FTP, SMTP, SOAP, WSDL and UDDI for communication between integration components.

- **Transformation** - Most of the EAI platforms are applying common data format that understandable by most of the participated system components, or the platforms provide transformation services to convert the sending data into the format that fit to the destination node. In this case, ESB provide dynamic transformation based on XSLT, service mapping, and data transformation like XML Path Language (xPATH), while MOM consists of built-in transformation mechanism to auto transform the received messages to be the application server’s native formats. Hence, Integration Broker provides service orchestration functions that support flexible component to component integration and allow coordination of multiple business processes implementation through intelligent routing and transformation features.

### 2.6 Conceptual Framework of Integration Review

As reviews of previous discussions, a conceptual framework introduced as ideas to solve the problem discussed in the previous chapter, and parallel with the ideas of discussed expected outcome (refer Section 1.5). Each of the concepts is based on the expected outcome against the integration problems. Basically the integration framework is a Hub/Spoke architecture, where all the participated application is connected to a centralized integration engine. The core benefit of the integration engine is to provide the capability for different independent applications to share resources (services). This
idea provides the solution to the problem of functionality interoperability lacking among systems.

The spoke or adapter is another component of the framework to overcome the integration problems. This connector is mainly used to facilitate the service sharing activities between participated systems. The core objective of having connector is to support conceptual level mapping of attributes among the participants. The capability encourages decrease of semantic interoperability issues. Table 2.2 depicts basic elements of the proposed conceptual framework of integration system.

**Table 2.2: Basic Elements of the Conceptual Framework**

<table>
<thead>
<tr>
<th>Component/Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Messaging Connector</strong></td>
<td>According to Apshankar, Clark, Hanson, Mittal, and Myerson (2002), an adapter should located in each of the participated system used to transfer messages by using standard middleware or transport protocols such as HTTP, FTP, SMTP, and web service protocols like UDDI, SOAP, and WSDL.</td>
</tr>
<tr>
<td><strong>Intelligent Routing</strong></td>
<td>As similar to service router in ESB, this element ensure the incoming and outgoing messages are delivered to the appropriate system components (Luo, Goldshlager, &amp; Zhang, 2005; Vallecillo et al., 2004). Routing of the framework should support both synchronous and asynchronous messaging services.</td>
</tr>
<tr>
<td><strong>Transformation</strong></td>
<td>One of the significant strength of ESB and MOM is its dynamic transformation architecture (Ji-chen &amp; Ming, 2006; Jung, Paek, &amp; Kim, 1999). As result, this element should responsible for messages content mapping and transforming between disparate systems with different platforms.</td>
</tr>
<tr>
<td><strong>Development and Monitoring Environment</strong></td>
<td>As refer to Kotsiopoulos, Keane, Turner, Layzell, and Zhu (2003), a good software middleware should have includes environment for developer to keep track all actions throughout any systems. Conceptually, Development environment provide features to deploy, publish, and binding services while Monitoring environment provides facilities to manage the flows of messages, message transmission status, cancel message transmission, and re-deploy the messages.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Provide features to ensure transferring messages over the framework are secured along all the related protocols. Security policies will be enforced by providing controls like encryption and decryption, digital signature, and appropriate authentication methods to the transferring messages.</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>Architecture of MOM, ESB and Integration Broker are free to implement specific communication standards for integration process (Jung, Paek, &amp; Kim, 1999; Ji-chen &amp; Ming, 2006; Kotsiopoulos, Keane, Turner, Layzell, &amp; Zhu, 2003). This element is mainly used to define open standard output for exchange data format between participated systems.</td>
</tr>
</tbody>
</table>
Besides that, only a single integration engine is required of whole architecture of the integration methodology. In other words, the framework provide single consensus interface for the pre-configuration to the service and attribute mapping activities.

2.7 Conclusion

The main focuses on this chapter is on the operation and representation perspective of three popular enterprise systems ERP, CRM and SCM with their respective objectives and architecture. The reality of the marketplace is that the maintenance of these enterprise systems independently is more costly and timely consuming than develop an integrated system. Hence, there are many similarities and possibilities for these systems to work together as a single extended enterprise system in order to fulfill a lot of business processes demands.

The pre-requisites of integration success are equivalence to the similarities among three types of the systems in terms of their operational and representational philosophy that closely related to the proposed service oriented methodology. In addition, the chapter has discussed the similarities that exist among ERP, CRM and SCM in the concept they represent and their architecture. The similarities between ERP, CRM and SCM have proven to be necessary for them to be integrated in this research.

The selected theory or approach for the research integration - Enterprise Application Integration (EAI) is discussed. To further understanding the operation of chosen EAI approaches, common platforms for EAI such as ESB, MOM and Integration Broker are also discussed. Finally, a conceptual framework for integration methodology is proposed in order to perform the capability of solving the integration problems.
CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

As refer to Bordens, and Abbott (2008), there are several types of research methodology such as Experimental Research, Observation Research, Survey Research, Specialized Research, Single-Subject Research, and so on. Each of the type represents different way of collecting and analyzing empirical evidence by following its logic. In this research, survey research was chosen as the research methodology due to its suitability to the research objectives. In addition, two modes of survey undertaken during the course of this research were survey questionnaire and interviews.

3.2 Survey Questionnaire

Survey questionnaire is mainly used to collect quantitative information about opinions or factual information based on the research objectives. Survey questionnaire could be adapted by almost any field or discipline, and it is this capacity for wide application and broad coverage which give the survey technique its great usefulness (Campbell, Angus, Katona, & Georgia, 1953).

In this research, survey questionnaire is chosen as the core method for data gathering and analysis. It is an efficient way of collecting information from respondents from different types of company throughout Peninsular Malaysia and Singapore and hence they are relatively easy to administer. In addition, survey questionnaire also provides the most economy way in data collection in term of transportation and the time to collect data could be reduced as well. In general, this method could potentially receive high respond rates from private networks due to its easy of answering, editing and analysis.
3.3 Questionnaire Design

There are several steps to be followed to construct a questionnaire survey. The first step of questionnaire design is to define a clearly objective of the study. According to Bordens and Abbott (2008), having a clearly defined objective will keep the questionnaire to focus on the behavior or attitude chosen for study. It is important to avoid focus too much in a single survey questionnaire that may leads to difficulty in summarizing and analyzing collected data in the future. In this research, a questionnaire form was prepared and designed to conduct the survey. Table 3.1 depicts the title and the main objectives of the survey questionnaire.

Table 3.1: Title and Objectives of the Survey Questionnaire

<table>
<thead>
<tr>
<th>Title:</th>
<th>Integrating ERP, SCM and CRM using Service Oriented Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives:</td>
<td></td>
</tr>
<tr>
<td>1) To collect information regarding typical type of computerized systems used to manage customers, suppliers, and the company business processes.</td>
<td></td>
</tr>
<tr>
<td>2) To get better understand on the current level of system integration in the companies.</td>
<td></td>
</tr>
<tr>
<td>3) To gather the perception of the company about the importance of the system integration.</td>
<td></td>
</tr>
<tr>
<td>4) To collect the Opinions regarding use of Service Oriented Methodology on integrating ERP, SCM and CRM systems.</td>
<td></td>
</tr>
<tr>
<td>5) To gather requirements and information about the feasibility of the promote idea of the research.</td>
<td></td>
</tr>
</tbody>
</table>

The next step of questionnaire design is to select the questionnaire format. Three types of questionnaire items were used in the questionnaire survey, which are open-ended item, partially open-ended item, and restricted item (closed-ended item). Open-ended questions allow respondents to answer the questions based on their own explanation without being restricted by a fixed set of possible responses. Closed-ended questions limit the respondents’ answers to a fixed set of possible responses. In the designed questionnaire, closed-ended questions are include multiple choice questions, yes/no questions, scaled questions (rate from scale 1 to 5 for example), and so on. A good
questionnaire design usually includes partially open-ended questions that used to overcome the limitation of the open-ended and closed-ended questions.

The designed questionnaire form in this survey consists of thirty-five questions prior to achieve the targets of the survey as stated in Table 3.1. The questions were divided into five sections (Section B to F), and distribution of questions are eight, fourteen, six, three, and four questions respectively. Table 3.2 shows the title and the target responses of each of the section A to F of the designed questionnaire.

### Table 3.2: Sub-Title and Objective of Sections in the Questionnaire

<table>
<thead>
<tr>
<th>Section</th>
<th>Sub-Title</th>
<th>Target Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Personal Particulars</td>
<td>To ensure that the responses of survey will be taken from different type of companies.</td>
</tr>
<tr>
<td>B</td>
<td>General Questions</td>
<td>To gather companies’ current infrastructure condition and business processes.</td>
</tr>
<tr>
<td>C</td>
<td>Your Company’s Level of System Integration</td>
<td>To capture most companies’ level of computerize system integration in the area of survey respondents.</td>
</tr>
<tr>
<td>D</td>
<td>Your Company’s Perception of System Integration</td>
<td>To collect information about most companies’ perception to integrate different computerize systems in the future.</td>
</tr>
<tr>
<td>E</td>
<td>Promotion of ERP, SCM and CRM Systems Integration using Service Oriented Methodology</td>
<td>To introduce key research idea, at the same time gather the opinions from enterprises regarding the feasibility of the idea.</td>
</tr>
<tr>
<td>F</td>
<td>Other Opinions</td>
<td>To capture more information from different perspectives related to survey objectives.</td>
</tr>
</tbody>
</table>

Section A of the questionnaire was mainly used to gather respondents’ particulars such as respondents’ job title, company name, location, and so on. From the gathered information on this section, it is easy to ensure that the questionnaire responses were captured from different areas, nature of business, position levels, and companies. Meanwhile, Section B provides more general questions that are intended to be easier to be answered. This is aims to impact the survey participants with favorable first
impression so that the remainder of the questions will not appear to be difficult to answer.

Questions on Section C are focus on the level of system integration among the real business companies while Section D is used to collect the responses on how important system integration is to the responded companies. Section E is the final section that consists of restricted format questions. The key idea of the research is introduced to the respondents before they start to answer the questions. The respondents are expected to provide their opinions on system integration for ERP, CRM and SCM system components.

The final section of questionnaire allows respondents to give their opinions in open-ended format. Respondents have the flexibility to elaborate their answers in this section. Table 3.3 shows distribution of different format questions into Section B to F in the designed questionnaire.

Table 3.3: Distribution of Question Type in the Questionnaire

<table>
<thead>
<tr>
<th>Section</th>
<th>Question Type</th>
<th>Total Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Restricted</td>
<td>Partially Open-Ended</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>Total Questions:</strong></td>
<td></td>
</tr>
</tbody>
</table>

3.4 Data Collection

The Questionnaire was sent to 150 staffs from 50 different companies throughout Peninsular Malaysia and Singapore. The primary business areas of the companies include education, government, health care, finance, retail, service, manufacturing, and non-profits. Hence, those 150 surveyed personnel plays heterogeneity roles such as
executive manager, IS manager, architecture, developer, system maintainer, consultant, business analyst and business manager in their respective company. The target respondents of the questionnaire were set to whose company is implementing or intend to implement more than one enterprise systems.

Due to privacy legislation in respondents’ company, some of the respondents were unable to provide the name and contact or other particulars. For those who had not responded the questionnaire within a fixed period of time, follow up reminder letters were sent to them in shortly.

The questionnaire survey was conducted at the period from mid November 2008 to mid January 2009. A total of 35 questions were asked in the questionnaire, and on average, would take 20 minutes to complete.

3.5 Pilot Test

The last step of questionnaire design involves the process of evaluate the validity of the questionnaire, or piloting the questionnaire. This is the acid test as to whether or not the instrument that has been so painstakingly designed is clear, understandable, and comprehensive with respect to what it purports to measure (Bordens & Abbott, 2008).

During the survey period, ten of the survey participants were invited to join the pilot test against the designed questionnaire. Apart from the prepared questions that related to the survey objectives, pilot test question were intercepted after each of the designed questions respectively. In addition, some of the criteria like Clarify, Loaded, Relevance, Specificity and Unambiguous Answers have been used to evaluate this survey questionnaire, with its explanation are shown in Table 3.4 below.
Table 3.4: Criteria and Description for Pilot Test the Questionnaire

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarify</td>
<td>To determine whether the survey question is easy to read and understand.</td>
</tr>
<tr>
<td>Loaded</td>
<td>To test whether the restricted questions (choose only one answer) have only one obvious answer or the respondent may answer the questions using more than one response choices.</td>
</tr>
<tr>
<td>Relevance</td>
<td>To determine whether the question is meaningful to the participant or related to his work site.</td>
</tr>
<tr>
<td>Specificity</td>
<td>To test whether the designed question is too general in nature causes respondent difficult to answer.</td>
</tr>
<tr>
<td>Unambiguous answer</td>
<td>To ensure the provided answers are unambiguous in term of meaning and the way it’s presented.</td>
</tr>
</tbody>
</table>

Besides using the above mentioned criteria, some other questions were asked to assess the quality of the questionnaire, such as:

- Total time to complete the questionnaire
- Reactions of the survey objectives, instructions, definition of terms, content, and language (clear, understandable, and so on)
- Whether Response scale (1 represents “not important at all” extent to 5 represent “very important”) provide enough choices for responses
- Any offensive or insulting questions that have been rejected by respondents
- Other opinions, comments, issues, suggestions or concerns

Table 3.5 depicts pilot question for an example multiple choice question asks on how integrated would the respondent rate his company’s various enterprise systems in terms of business processes as being right now, with its available answer include “Totally Integrated”, “Very Integrated”, “Somewhat Integrated”, and “No Integration”.

54
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Yes</th>
<th>No</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clarify:</strong> Are this question easy to read and understand?</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td><strong>Loaded:</strong> Are this question has <strong>ONLY</strong> one <strong>OBVIOUS</strong> answer for you?</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td><strong>Relevance:</strong> Are this question relevant to your work site?</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td><strong>Specificity:</strong> Are this question too general in nature? (difficult to answer)</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td><strong>Unambiguous Answers:</strong> Are this question should have answered in more than one way?</td>
<td>☐</td>
<td>☐</td>
<td></td>
</tr>
</tbody>
</table>

### 3.5.1 Pilot Test Result Analysis

Pilot test process was started before the delivery of questionnaire to the companies, or at the early of November 2008. Out of ten representatives for the pilot test, there is one Executive Manager; three of them are IT Architect/Developer, a System Maintainer, an IT Consultant, an Admin Executive, an IT Project Officer, a System Analyst and a Teacher. This indicates that the opinions gathered for piloting the questionnaire are from different kind of professional perspectives.

Discussion was held individually to each of the representatives after collecting and analyzing their feedbacks with pilot test questions. As results of piloting the questionnaire, some of the constructive feedbacks to the questionnaire include the following:

- Questionnaire should have to differentiate which questions do not required respondent’s company to have multiple computerized systems
- Target respondents should be stated
- Grouping similar questions into one
- More space to provide open-ended opinions
All of the designed questions were passed the test of five pilot test criteria (Clarify, Loaded, Relevance, Specificity and Unambiguous Answers) as majority of the respondents were satisfied with the questions. In addition, this result was proved by the most favorable answers selected by the respondents, as shown in Table 3.6. In responding to the participants’ feedback, then the questionnaire was reviewed and redesigned based on the gathered feedback.

Table 3.6: Most Favorable Answers to Five Criteria by Respondents

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Favorable Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarify: Are this question easy to read and understand?</td>
<td>Yes</td>
</tr>
<tr>
<td>Loaded: Are this question has ONLY one OBVIOUS answer for you?</td>
<td>No</td>
</tr>
<tr>
<td>Relevance: Are this question relevant to your work site?</td>
<td>Yes</td>
</tr>
<tr>
<td>Specificity: Are this question too general in nature? (difficult to answer)</td>
<td>No</td>
</tr>
<tr>
<td>Unambiguous Answers: Are this question should have answered in more than one way?</td>
<td>No</td>
</tr>
</tbody>
</table>

3.6 Administration of Questionnaire

After the questionnaire is constructed, the next step is to determine how to administer it. Obviously, the purposes of this process are to get the questionnaire answered and to find out the relevant respondent group amongst the survey population. To achieve these purposes, few methods were used to distribute the questionnaire to the participant volunteers and get the responses from them. The methods include Online Survey and Electronic Mail Surveys.

Online Survey is the key method of the survey questionnaire. This is a method where the questionnaire was created electronically and posted on the Internet through online survey service providers. Two main selected service providers are FreeOnlineSurveys.com and SurveyMonkey.com. Through the given URLs, the
participants were invited to answer the questions by completing the form the posted on the website. The auto generated URL by FreeSurvey.com was:

http://FreeOnlineSurveys.com/rendersurvey.asp?sid=rny1mvq9jgrqvr527074

The significant advantages to using Online Survey include that the data can be collected easily and quickly, hence large numbers of participants are supported by the website.

Another method of administer the questionnaire is through Electronic Mail Survey. This is the more convenience method as compare to Online Survey. This method only involves attach the questionnaire form and mail it to the selected participants. However, the drawback of the method is that many of the email had move into junk mail automatically and this reduce the response rate from the receivers.

Before these written survey methods are conducted, a letter of introduction, definition and explanation of the survey was also attached. Upon responses are received, any incomplete questionnaire forms were sent back to the participants along with the collection comments. Hence, follow up mailings were also sent out to the target respondents with additional words of encouragement to participate.

3.7 The Interview

As preparation to the face to face interview, Phone Survey method was used to conduct the survey. The questions are more easily to ask through verbally than written form. However, some more complex and long questions were not answered truthfully. Besides, it is not easy asking those close-ended questions with list of answer choices because the participants may not remember all the choices through listening.
The key purpose to conduct face to face interview is to gather more open-ended opinions from the participants. However, the questions asked were structured and based on the designed questionnaire forms. All opinions and recommendations of the participants during the interview were recorded word for word. The responses from the closed-ended questions during the interview will be combined with the written survey results for analysis.

3.8 Population and Sampling Techniques

According to Bordens, and Abbott (2008), population of a survey is defined as all individuals making up a definable group of interest in a study. Survey population is also known as the coverage of the survey, which means how the survey information can be obtained. Once population of a survey is defined, a group of respondents consists of smaller number of people are selected from the population.

In order to fulfill the study objectives that mainly focuses on gathering information regarding to the system integration levels among the companies, it is suitable to define the population of this survey as the personnel who are working in companies throughout the survey areas. The main reason to fix the survey area to Peninsular Malaysia and Singapore is because the business environment and the IS demands among companies in these areas are very similar, and hence the geography advantages reduce the administrative cost and time of this survey. In addition, the population of the study was narrow down to focus on the companies that are implementing more than a single computerized systems.
Sample size of the survey is the number of observations or participants that directly responded to a particular survey. In this study, research was defined the number of companies responded to the survey as the sample size. To determine sample size, researcher is required to decide the acceptance margin of sampling error (SE) and magnitude of the differences researcher expected to find (M), as according to Moser and Kalton (1972). The suggested formula is described as:

\[ N' = \frac{M' (1 - M')}{(SE)^2} \]

where \( M' \) is the expected magnitude of differences and SE is the acceptable sampling error. However, there are no specific rules of condition to determine both components (Bordens & Abbott, 2008). For this reason, it is defined that the acceptable SE for this survey compare to real target population is 5 percents (0.05). Hence, most of the companies in the target survey area are using at least one computerized system, the \( M' \) is defined as 90 percents (0.9). According to formula, the required sample size is 36 companies.

Sampling techniques are usually classified into either probability or non-probability qualitative approaches. Jackson (2006) has mentioned that in probability sampling, each of the members of the defined population has the equal possibility to be part of the sample. Meanwhile, non-probability sampling does not involve random selection and every member in the population does not have an equal chance of being selected.
3.9 Justification of Research Sampling Methods

For this survey, sampling techniques used are within the scope of probability and non-probability sampling.

Among the methods of non-probability sampling methods, Judgment Sampling is employed to analyze the perspectives of the survey population. The main reason to use this method is because the population defined for this research (Peninsular Malaysia and Singapore) is cover widely where clustering population may not be as effective as expected. By using judgment sampling, the person most knowledgeable on the topic of the study selects elements of the population that researcher feels are most representative of the population (Anderson, Sweeney, & Williams, 2009).

Respondents who categorized as judgment sampling are who in the best position or most advantageously to provide information in term of the research, according to candidate’s own justification. Particularly, the target respondents by employed this sampling methods are who involved in supply chain activities at their current or previous working experiences. Several criteria for selection respondents of judgment sampling include:

- Experiences in managing, operate, plan, or deliver at least one of ERP, CRM and SCM system
- Basic understanding the concept of service oriented and middleware, hence been applied middleware software to enhance their business process
- Able to evaluate their company level of integration, in term of functionality and performance
- Demand in revolution of current software integration methodology such as MOM, ESB and Integration Broker
For probability sampling survey techniques, Random Sampling is selected as another sampling method to analyze survey result. According to explanation by Babbie (2007) for random sampling, each element has an equal chance of selection independently of any other event in the respondent selection process. However, the main benefit of this random sampling method is that it able to avoid a certain levels of sampling error because every member in the population has a known zero probability of being selected, or some of the elements of population is uncovered. In this research, some of the survey methods are used to facilitate this random sampling method, such as online survey and electronic mail survey as discussed in Section 3.6.

In chapter 4, the results analysis of the survey will be mainly based on these two sampling techniques.

### 3.10 Conclusion

Survey research was used as the research methodology to evaluate the research idea. It is the most suitable way of collecting and analyzing empirical evidence as required by this research. Two modes of the survey research used to collect research data were survey questionnaire and interviews.

Survey questionnaire is a survey method that is easy to administer and efficiently collect information from different companies. However, questionnaire design is a long process and requires careful attention. First step of designing the questionnaire is to define the survey objectives. After the objectives were defined, the format of the questionnaire was then fixed to Open-Ended, Partially Open-Ended and Closed-Ended questions. Therefore, several guidelines were followed to assure the quality of the designed
questionnaire. The survey questionnaires were distributed to the participants throughout the Peninsular Malaysia and Singapore.

Pilot test is an important process of conducting questionnaire survey. Ten volunteers participants were joined the pre-test to improve the quality of the questionnaire and eliminate mistakes. The feedbacks of the pilot test were analyzed and necessary collections had been made to the questionnaire.

The questionnaire survey was conducted through written survey method, such as Online Survey and Electronic Mail Survey. The questionnaire was posted to Internet via survey service provider FreeOnlineSurveys.com. Reminder emails were sent to the participants who did not response the survey invitation for certain periods. Hence, incomplete responses questionnaire were also sent back to the participants for correction.

Besides the written survey methods, phone survey was taken as a preparation of face to face interview to participants. Face to face interviews were conducted to several participants aimed to capture more perceptions to the research ideas. The accumulated results of the survey questionnaire and interviews were administered and will be analyzed and reported in chapter 4 (Data Analysis and Findings).
CHAPTER 4: DATA ANALYSIS AND FINDINGS

4.1 Introduction

As mentioned earlier in chapter 3, several written survey methods like online survey, electronic mail survey, and interview have been conducted to gather result for a designed questionnaire. The results gathered through the survey methods are main focuses of this chapter. Before the discussion on the results of each questionnaire section, population of target respondents and sampling techniques used to collect the survey data are outlined.

As mentioned in the Chapter 3 of this dissertation report, the key objective of the survey questionnaire is getting more details information regarding the current level of system integration among the companies and the perception of respondents on the importance of system integration to their companies. In addition, this survey also seeks for the respondents’ opinions regarding the flexibility in the key research idea, which is integrating ERP, CRM and SCM system components using a serviced oriented architecture.

In order to comply with the confidentiality and anonymity of the respondents’ business, the individual responses, including the any provided contact information and business name are not appeared in this report of survey results analysis.
4.2 Data Analysis for Judgment and Random Sampling

The collected data from the survey is analyzed in the following sub-chapters, and the discussions are made based on each question. Due to judgment and random sampling have two different characteristics and roles in term of system integration, the following analysis will also compare the responses provided by both sample groups.

4.2.1 General Information about the Respondents and Companies

This sub-section focuses on the results collected for Section A and B of the survey questionnaire. The respondents were asked to provide their particulars job title, company name and location so that it could be ensured that the surveyed respondents are match with the defined population. Hence, respondents were asked whether their company is implementing IT infrastructures like ERP, CRM or SCM enterprise systems.

4.2.1.1 Respondents’ Role in the Company

This question covers the respondents’ role in their respective company. The main purpose this question is to ensure that the samples were responded by the relevant respondents. Roles as listed in Table 4.1 indicate the roles that candidate feels are most representative of the survey population. In other words, the responses with “Other” roles are analyzed using random sampling.
Table 4.1: Role of Judgment Respondents in Company

<table>
<thead>
<tr>
<th>Role</th>
<th>Frequency of Judgment Respondents</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Manager (CEO / COO / Director / LOB)</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>IT / IS Manager</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>IT Architect / Developer</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>System Maintainer</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>IT Consultant</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Business Manager</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Business Analyst</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>22</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

According to the results, the respondents of this survey play different roles from each other. Comments from them indicate can be seems as heterogeneity and applicable to the defined survey population.

The judgment respondents are as expected consists of eight types of roles as defined in the survey. As compare the judgment respondents to the random respondents from the perspective of their role played, 94 percents of the random respondents of respondents had other roles than the defined eight categories. However, these respondents had admitted that they had certain levels of the experience on processing or managing the supply chain activities in their respective company.

4.2.1.2 Size and Business Area of Respondents’ Company

In order to have better understanding of the respondents’ company size and business area, each respondent estimate the number of employees in their companies and provide their primary business area. The question categorize company size into three categories, which are small (less than 50 employees), medium (50 to 500 employees) and large (more than 500 employees), according to candidate’s own justification.
There are not much different between the percentage of judgment respondents and random respondents on their company size. Hence, the similar results gathered for the company primary business area, where distribution of number companies on each business area were not much different between judgment and random respondents.

4.2.1.3 ERP, SCM and CRM Related Systems in the Company

Respondents were asked whether their companies were using any accounting based computerized system that mainly use for identifying and planning resources within the organization. Obviously, the question was used to justify whether most of the respondents’ company had implemented ERP related systems.
More than 75 percents of the respondents for both judgment and random sampling have implementing ERP systems. This indicates that ERP is one of the most popular enterprise systems used in the market. As according to Sheikh (2003), the ERP has been long time became one of the most important applications since 1990’s.

Meanwhile, respondents were asked whether their company purchase products or services from suppliers via Internet or any other computerize systems. This is one of the common functions provided by a SCM system. 25 percents of random respondents and 20 percents of judgment respondents did not implement SCM systems. This result indicated that the judgment respondents were more actively involved in computerized supply chain activities.

Besides that, respondents were also asked whether their company had prepared any information system for their customers to get to know or purchase company products or services. This question were used to gather percentage of companies were using CRM related systems or any web pages to communicate with their customers. Result was shown when the majorities (81 percents) of random were using CRM systems. CRM related systems used by the respondents included:

- Organization official website
- Microsoft Dynamic Axapta
- Customized Customer Relationship Systems

### 4.2.1.4 ERP, CRM and SCM Integration in the Companies

Integration between ERP and SCM were weak based on the responses from judgment and random respondents. Most of them (65 percents of judgment respondents and 75 percents of random respondents) were process the quotation via email and telephone.
Equal percentage of responses (25 percents) from both sampling did the processes via computerized systems.

However, there were not much different between the results of judgment and random respondents on the issue of integration between ERP and CRM systems. Most of the companies did not implement integration of these systems.

On SCM and CRM integration issue, almost half of judgment respondents disagreed with the collaboration, but 94 percents of random respondents were agreed with direct collaboration between customer and supplier through the systems. The reason might because the random respondents were too optimistic on the integration or some of them did not have experiences on supply chain activities. In addition to judgment respondents, they might have experienced with some problems to enable collaboration of customers and suppliers.

4.2.2 Level of System Integration in the Companies

This sub-chapter reporting the results captured from Section C of the survey questionnaire. The respondents were asked to provide information regarding the level of the computerized system integration for the survey respondents. Hence, respondents were required to self-assess the level of integration of information systems throughout their company and their partners.

4.2.2.1 Current Systems Implemented in the Companies

Several popular enterprise systems were listed in the question for respondents to select the implemented systems in their companies. Table 4.2 depicts the distribution of the companies’ implemented systems.
Table 4.2: Implemented Systems in the Companies

<table>
<thead>
<tr>
<th>Enterprise Systems</th>
<th>Judgment Responses</th>
<th>%</th>
<th>Random Responses</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Relationship Management</td>
<td>6</td>
<td>27</td>
<td>8</td>
<td>44</td>
</tr>
<tr>
<td>Enterprise Asset Management</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>Enterprise Resource Planning</td>
<td>14</td>
<td>64</td>
<td>14</td>
<td>78</td>
</tr>
<tr>
<td>Financial Management System</td>
<td>4</td>
<td>18</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>Human Capital Management</td>
<td>4</td>
<td>18</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>Performance Management</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>22</td>
</tr>
<tr>
<td>Product Lifecycle Management</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Supply Chain Management</td>
<td>6</td>
<td>27</td>
<td>4</td>
<td>22</td>
</tr>
</tbody>
</table>

Most of both judgment and random respondents selected CRM, ERP, SCM, and HCM as the systems used by the companies. These systems have proved to be the most popular systems in the markets nowadays. According to the collected statistic above, it can be concluded that ERP is the most popular systems among the companies, where 64 percents from judgment responses and 78 percents from random responses were chosen ERP. Other unlisted implemented systems included SAP and Field Service Systems.

As referring to the survey results, most of the companies had implemented the systems since 10 years ago. The results indicate the companies were continuing to implement new systems for the current five years periods. In addition, it can be concluded that the companies will continue to develop new systems, or integrate the existing systems in the coming years. From the other perspectives, the proposed methodology in this research will be beneficial to most of the companies.

Judgment and random respondents were chosen same group of systems – ERP, CRM, SCM and HCM as most systems used in the companies. ERP are most company used at the moment, followed by CRM, SCM and HCM. The result has proves that the selection
of ERP, CRM and SCM as participants of system integration is very useful to the companies at the moment.

4.2.2.2 Users of the Systems

Respondents were asked to describe the categories of users who play as stakeholder of the systems. Both judgment responses (75 percents) and random responses (81 percents) were feel that administrative staffs were among the users who contribute most to the existing systems. Similar results were gathered from CIO or CTO group, who involved less in the systems operation as according to both groups of respondents.

Both results might be affected by the previous result where ERP were most popular systems in the companies for this moment. Administrative staffs usually involve most in the ERP systems to conduct business processes such as issue purchase order, sales order, invoice, delivery order, finance and so on. For the reasons, administrative staffs will be the main target users of the system integration in the research.
4.2.2.3 Systems Developed by the Same Vendor

Respondents were asked whether their company computerized systems was developed under the same software vendor. Respondents were also provided the reasons to their answer.

The responses from judgment respondents are not able to justify whether the companies were preferred on same or different software vendors. However, random respondents give the answer on this issue, where 38 percents of the companies assigned systems development to the same vendors but 62 percents were not. This means that the companies were preferred to assign different vendors to develop the systems, and this might create many issues on integration and information sharing problems.

4.2.2.4 Information Sharing among the Systems

Respondents were asked whether their existing systems were able to share information whenever was needed to support particular business processes.

![Information Sharing among Systems](image)

**Figure 4.4: Information Sharing among Existing Systems**
The responses from random responses were not provided any constructed results on this issue because of the same number of responses answered ‘Yes’ or ‘No’. However, the responses from the judgment respondents have show that the companies systems were able to share information to achieve business purposes. However, it does not mean the systems were able to support any business workflows without having any troubleshoot and maintenance required.

4.2.2.5 Maintaining Systems that are Not Initially Work Together

This question determines whether the companies were maintaining systems that were not initially to be work together. In other words, some companies were tried to make different systems co-operate to achieve certain processes, either by combining their functionalities, or developing brand new platform to link them together.

Both judgment and random respondents indicated the same results, where almost 60 percents of the companies were maintaining systems that were not initially designed and developed to be work together. Due to more companies’ systems were developed by different vendors, it is easy to explain why more company faced the issue on maintaining the systems that were not work together. More importantly, the results have further justified the needs of system integration to the companies.

4.2.2.6 Integrating Different Systems to Fulfill Business Requirements

As refer to previous session, more companies were facing issue on maintaining different systems to be worked together. The next concern is to determine whether companies were facing problems on integrating different systems.
The responses from random responses were not provided any constructed analysis on this issue because of the same number of responses answered ‘Yes’ or ‘No’. However, the responses by the judgment respondents indicated more companies were faced problems when integrated different systems to fulfill business processes. This result was consistent with the issue where more companies were maintaining systems that were not initially worked together. Besides, the result were indirectly affected the result of next question, where respondents were asked about the integration level of the systems.

4.2.2.7 Integration Level of the Systems

This topic was used to determine integration level among the existing systems in the companies. The summary of the analysis are shown in the Figure 4.5 below:

![Integration Level of Systems](image)

**Figure 4.5: Integration Level among Existing Systems**

Both groups of respondents provided similar results where most of them felt their systems were somewhat integrated only, and fewer were rated their systems totally integrated. Not like “Very Integrated”, “Totally Integrated” means the systems implemented are represents the same concept (conceptual equivalence) regardless the technologies applied by the systems. Most of the respondents felt that their company systems were somewhat integrated, or totally no integration. It might because more
companies were maintaining systems that were initially developed separately, or they might face problems when operate processes involved more than a single system.

4.2.2.8 Middleware Suites Implemented in the Companies

Respondents were asked whether their company were used any middleware suites or programs to integrate different systems. For those who had used the programs were asked to state the name or vendor of the programs, and their satisfaction regarding the effectiveness of the programs.

Both groups of respondents indicated more responses were not implemented middleware suites to integrate systems. This group of responses might choose to develop their own platform or software systems to make the systems as a whole, rather than purchase any middleware suites to achieve same objective.

Judgment and random respondents were preferred to use Microsoft programs to integrate separate systems. This might because Microsoft is a well-known software vendor and the user friendly interfaces of the software packages are able to reduce the development period. Sun Microsystems and SAP were also popular among the middleware suites.

As refer to same answer rate by both judgment and random responses (75 percents satisfied and 25 percents dissatisfied with the middleware), it can be concluded that most of them had not much argument with the beneficial of middleware programs. However, a quarter of the companies still could not achieve business purposes via middleware suites available in the market. This could be one of the reasons for the necessity to the proposed research integration methodology.
4.2.2.9 Electronic Data Interchange (EDI) Implemented in the Companies

Usage of Electronic Data Interchange (EDI) in Companies is analyzed in both judgment and random respondents. EDI refers to technology to transmit data between companies electronically, by using agreed standardized business terms. Due to EDI term is more technically and might be unfamiliar to certain respondents, an answer of “Don’t Know” was provided in the question. The summary of the analysis are shown below:

As compare to the both groups of respondents, more random respondents were implemented Electronic Data Interchange (EDI) technology rather than judgment respondents. This was indicates that EDI was used by different kind of companies in order to share data with their business partners. However, 31 respondents or 86 percents of the responses indicate that they did not have the knowledge or yet to implement EDI technology. This means EDI is more likely to have big potential to be used in any integration platform, including the research systems.

Figure 4.6: Companies Implemented EDI
4.2.3 Perceptions of Respondents on System Integration

The respondents’ perceptions regarding the system integration issues to their company will be analyzed in the next sub-chapters. Understanding the perceptions of the survey respondents means to know better the acceptance level of companies throughout Peninsular Malaysia and Singapore regarding the system integration issues. The objective of this section of survey is to collect information about most companies’ perception to integrate different computerize systems in the coming periods of time.

4.2.3.1 Importance of System Integration

The importance of system integration benefit to the companies is analyzed in both judgment and random respondents. Of 40 respondents, 75 percents of them were think that combine different systems functionalities as a whole was “Essential” or “Very Important” to support company business processes. Therefore, most of the judgment responses show that the integration was an essential but most of the random respondents only felt it was very important to the processes.

To better understanding of respondents’ opinion regarding the importance of system integration, respondents were asked to rate the importance (from “Not important at all” to “Very important”) of the integration by its benefits. Seven benefits of system integration for respondents to rate were:

- Improve the information exchange between customers, suppliers and your company
- Increase the competitive advantage of the company
- Increase the capability of supply chain collaboration
- Centralize business policies and rules among each of the existing systems
- Increase customer satisfaction
- Improve communication with suppliers
- Reduce cost of maintaining and enhancing independent systems

The summary of the findings were shown in Figure 4.7 as below:

![Importance of Integration by Benefits](image)

**Figure 4.7: Importance of System Integration by benefits**

Each of seven benefits of integration as stated in the survey question were received more than 60 percents of respondents who rated the benefits above average. Regardless of respondents group, the respondents were felt that system integration is important to improve information exchange, increase competitive advantage, improve collaboration between partners, centralize business policies, increase customer satisfaction, improve communication with suppliers, and reduce cost of systems maintenance.
4.2.3.2 Current Levels of System Integration

Respondents were asked about their company position in term of system integration at the moment. Table 4.3 depicts the collected statistic for this issue.

Table 4.3: Current Level of System Integration among the Companies

<table>
<thead>
<tr>
<th>Integration Level of the Company</th>
<th>Judgment Respondents (%)</th>
<th>Random Respondents (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has no intention to integrate systems</td>
<td>7 (32)</td>
<td>4 (23)</td>
</tr>
<tr>
<td>Has plans to integrate systems</td>
<td>2 (9)</td>
<td>5 (27)</td>
</tr>
<tr>
<td>Implementing system integration</td>
<td>10 (45)</td>
<td>7 (39)</td>
</tr>
<tr>
<td>Already has integrated system</td>
<td>3 (14)</td>
<td>2 (11)</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>22 (100)</strong></td>
<td><strong>18 (100)</strong></td>
</tr>
</tbody>
</table>

According to statistic, most of the companies were implementing system integration to achieve business purposes. Of 40 responses, only five responded companies already had an integrated system that able to fulfill business needs at the moment.

Result from both judgment and random respondents depicts not many of companies throughout the survey population were already had an integrated system. This has allows new idea of system integration to be implemented into the company network. In other words, the proposed system integration system would have great potential to be accepted by the companies according to the result.

4.2.3.3 Preferred Methods of System Integration

Respondents were asked whether they preferred purchase middleware programs, or develop a brand new system to integrate different information systems. Table 4.4 describes the outcome of the survey.
Table 4.4: Preferred Methods of System Integration

<table>
<thead>
<tr>
<th>Method of System Integration</th>
<th>Judgment Respondents</th>
<th>(%)</th>
<th>Random Respondents</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchase or develop middleware suites or programs</td>
<td>12</td>
<td>55</td>
<td>13</td>
<td>72</td>
</tr>
<tr>
<td>Develop brand new systems to integrate existing systems</td>
<td>10</td>
<td>45</td>
<td>5</td>
<td>28</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>22</strong></td>
<td><strong>100</strong></td>
<td><strong>18</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Survey result shows that approximately 55 and 72 percents of judgment and random respondents respectively preferred to develop middleware suites rather than develop a brand new system from scratch in the issue of system integration. This has gives a clearer facts that the proposed integration methodology that involved development of middleware programs would be best suites to current market demands.

4.2.3.4 Barriers to System Integration Implementation

Respondents were asked to describe what they considered to be the limitation or barriers that influence the implementation of system integration in their company. A range of responses were given and a number of limitations were often selected by single respondent. These are listed below (with the number of responses specifying that particular barrier or limitation in brackets):

- Lack of integration expert (23)
- Complexity of legacy systems (19)
- Users from one department refuse to share their data with other department (16)
- High integration cost and time consuming (16)
- Existing systems are constantly changing (13)
- Problem on mapping the data between different systems (13)
- Conflict requirements of different departments (12)
- No consensuses on the final outcome of interface design of the integration (10)
Besides the limitations above, some other miscellaneous reasons why system integration was yet to be implemented in the companies included:

- Lower Return on Investment (ROI) from system integration
- Size of existing systems or the company was too small, not suitable to implement system integration
- Most of the integration package required customization and modification of existing systems
- Some integration software were not compatible with common business supportive software like Microsoft Office
- Data migration was difficult

4.2.3.5 Potential Systems to Participate in the Integration

Several popular enterprise systems were listed in the question for respondents to choose their expected systems that capable to increase collaboration between customer, supplier and their company.

CRM, ERP and SCM are three most potential systems to increase company competitive advantage as according to the survey result. From the collected statistic, it can be concluded that CRM is the most popular systems required by the companies, where 80 percents from judgment responses and 81 percents from random responses were selected CRM.
4.2.4 Promotion of System Integration using SOA

The main idea of the research system was introduced to the survey respondents through this section of survey. The respondents were asked to give their opinions regarding the feasibility of the proposed research system. Hence, the aim of this survey section is to gather requirements for design and development of the research system.

4.2.4.1 Participants of the System Integration

Respondents were asked whether they agreed that ERP, CRM and SCM were most suitable group of enterprise systems to be integrated functionally to support business processes. Almost all of the respondents were agreed with the statement while only one respondent of judgment and another one of random respondents were disagreed with it. The disagreed judgment response was suggested that the participants of integration consist of Field Service Management system instead of CRM together with ERP and SCM systems. In other views, the disagreed random respondents were felt that other systems like Human Resource Management, Product Quality Management, and Plant Management should have the equal chances to be participated in the research integration system.

4.2.4.2 Service Oriented Methodology in the Integration

Two important components of proposed service oriented methodology - Conceptual Model (CM) and Adaptive Service Connector (ASC) were introduced to the respondents before answering questions. The respondents were asked whether they agreed that the methodology have potential to integrate existing systems to be conceptually equivalent. Similarly to 4.2.4.1, majority of respondents were agreed that the methodology will be a good idea to integrate disparate systems.
Of 40 respondents, only 2 of judgment responses and one of random respondents were disagreed with the idea of service oriented methodology. One of the responses stated that the methodology were technically difficult to be implemented because different system vendors do have different maintenance policies, and normally existing database cannot be modified by different vendors. Another disagreed respondent felt that the methodology will required more manual work to map the systems. Hence, one more opinion stated that the partial automated business process were difficult to be fully automated through the methodology.

4.2.4.3 Functionalities of the Integration

In order to gathered opinions regarding to the functionalities to be included into the proposed integration methodology, respondents were asked to describe their expected functionalities for the proposed methodology. A range of responses were given and a number of functionalities were often selected by single respondent. These are listed below (with the number of respondents specifying that particular functionalities in brackets):

- Mapping data from different systems that have the same or similar meaning (27)
- Create, modify or delete functionalities which are not available for existing systems (28)
- Security feature to authorize specific users to access particular functionalities (26)
- Accumulate specific data from different systems and generate one single report listing (23)
- Customize the report listing columns based on different requirements (22)
- Standardized business terms or naming understandable by participated ERP, CRM and SCM subsystems (22)
Guideline to integrate functionalities from different systems (14)

Some other recommendations by respondents were included:

- Centralize functionalities, for reuse and easy to access
- Secure file transfer using Encryption, PKI, SSH, and SSL standards
- No hardcode of conversion rules. All conversion or mapping should be maintained in database
- Log files for file transmission monitoring and auditing purposes

4.2.5 Other Opinions

The last section of the survey questionnaire allowed respondents to provide opinions based on several particular topics in open-ended format. The objective of this section of questionnaire is to gather more information regarding system integration from different perspectives. The analysis of gathered data is focuses on two topics, which are benefits of system integration and weaknesses of existing systems in the companies. The questions in this section were not compulsory to be answered by the respondents.

4.2.5.1 Beneficial of System Integration

For those company was implementing or implemented system integration, the respondents were asked whether the system integration beneficial to their company. Hundred percents of random responses were agreed that system integration was beneficial to their company, while only 2 judgment responses had the opposite opinions. It can be concluded that the majority of responses were felt system integration is a must for companies with multiple systems. Some of the benefits stated by the respondents were:
- Every system getting up-to-dated information
- Better operation and management control
- Encourage management to make accurate decision and planning
- Provide centralized information and library sharing
- Increase customer satisfaction
- Reduce paper works, simplify day to day activities
- Save time and cost to maintain systems separately
- Help to create truly global partner community, increase company revenue
- Reduce the waste of different resources
- Streamlined business processes and increase productivity and efficiency
- Alternative way to upgrade or customize legacy systems because normally they are not as user friendly
- Prevent duplicate works and incompatible data

Besides that, one of the respondents felt that system integration would not beneficial to the companies because they had not yet implemented it, while another one felt no needed for them to implement system integration.

### 4.2.5.2 Limitations and Weaknesses of Current Systems

As results shown the needs of system integration and a lot of its advantages, it is meaningful to discuss the limitations and weaknesses of existing systems at the moment, which caused most of the companies promoted system integration. In short, the limitations and weaknesses of existing systems were listed as:

- Slow processing complicated processes especially involve wide area data transfer
- Compatible issues between existing systems
- No existed intact integrated system
- Too complicated to operate
- No centralized operation, lack integration and no policy defined
- Discrepancy of same information from different resources
- Too many systems involved sometimes to support single business process
- Legacy systems like mainframe are hard to maintain, and usually difficult to integrated with other systems

Hence, any proposed integration system has the ability to overcome these limitations of company systems.

4.3 Open-Ended Interview Results Analysis

As mentioned in chapter 3, the interview session of the survey was through face to face interview and phone survey. The interview was normally divided into two sessions, where at first the respondents were asked with closed-ended questions, and then followed by open-ended questions. However, this session will only focus on the discussion of open-ended responses because the results of closed-ended interview responses have been combined into the discussion of previous sub-chapters which categorized as judgment responses. The findings from interviews from the respondents are summarized as below:

- In view of the barriers to integration success in companies, each of the companies in the study commented on the critical shortage to integration expert as a major barriers to integration success. It is being addressed that the cost to hire professional or outsourcing system integration is very high and time consuming. In view of technical issues, data being extracted out from a system
Sometimes are incomplete or incompatible with other technologies. Further modification is needed before it is ready to be used by other systems.

- Majority of the interviewees was agreed the limitations and weaknesses of their current computerized systems caused a lot of problems to the daily business processes, and they believed system integration would be able to overcome the problems. One interviewee described the major problem was because the lack of customization options of existing systems and it was not updated for current uses. Another interviewee has said they had to repeat the same works via different systems due to poor integration level of existing systems.

- Most of the interviewees were expressed the beneficial of system integration due to the limitations and weaknesses of existing systems encourage implementation of system integration. One of the interviewees was expressed that his company was implemented system integration and it helped the company to reduce conflict among different departments information. The implementation was succeeded to reduce time and cost of process operations, and output data of integrated systems were more accurate.

- Besides that, another interviewee indicated that the outcome of system integration was helping their company to better understanding customer needs and improved communication with customers. For instance, customers are able to check the service and product delivery status by their own through the integrated of ERP and CRM systems without waiting response from internal staffs. Generally, all of the interviewees were believed that system integration will improve the integration between the company with their customers and suppliers, whilst increase their competitive advantages and then increase the company revenue.
4.4 Results and Objectives of the Survey

To assess the successful degree of the conducted survey, the objectives of the survey are reviewed to compare with the collected results. As mentioned in chapter 1, the main purpose of the survey is to gather information regarding current level of system integration and the role of the middleware software played within organizations located in Peninsular Malaysia and Singapore. Candidate also received stimulate responses regarding to other research objectives such as getting known the perception of system integration, assessment on service oriented technology for enterprise systems integration, and feedback on the research ideas. The key objectives were achieved through collected answers from questions in Section C (Current level of system integration), D (respondents’ perception of system integration), E (introductory of research idea) and F (Open ended opinions regarding research topics) of the questionnaire. The following describes survey results along with objectives by questionnaire section:

- Section A of the survey was aimed to ensure each of the survey respondents had different unique criteria, in terms of their business area, company types, position of respondents in company and so on. According to collected answers, most of the respondents played different roles in their respective company if compare to others. Hence, all of 40 respondents were come from more than eight different types of business areas, and there was almost equal percentage for small, medium and large size of the responded companies.

- As expected for Section B of the survey, the collected results indicated the basic computerized infrastructures and business processes in the companies. According to the results of combination of random and judgment responses, more than 80 percents of the companies implemented ERP, approximately 75 percents implemented SCM and more than 60 percents implemented CRM.
Hence, most of the surveyed companies business processes needed supports from an integrated ERP, SCM and ERP system. This is because the business processes involved integration between two or three of the systems.

- Questions in Section C of the survey were used to capture level of computerize system integration among the surveyed companies at the moment. Survey results were shown ERP, SCM and ERP as most favored enterprise systems and administrative staffs were most active user group to the systems. Besides, more companies indicated their systems had certain level of capability to share data, but most of the companies were still maintaining the systems that were not initially designed to be worked together. Due to most of the companies faced problems while integrating the systems, they felt that their systems were somewhat integrated or totally no integration. Fewer companies used middleware and EDI software might be the reason caused the lower level of system integration.

- In Section D, respondents were given their opinions regarding the importance of system integration, and their company intention to system integration. According to their opinions, system integration plays important roles in improving communication between business partners and information exchange, increasing customer satisfaction and competitive advantages, centralizing business processes and reducing cost of maintenance. Due to so many advantages of system integration, most of the companies have plans to implement system integration by developing or purchase middleware suites. The middleware suite is expected to be used to integrate some popular systems chosen by the respondents like ERP, SCM, CRM and HCM.

- As expected for Section E of the survey, respondents provided their opinions on system integration for ERP, CRM and SCM system components by using
proposed service oriented methodology. According to the results, ERP, SCM and CRM were confirmed to be three most suitable participants of the proposed system integration method. Hence, most of the companies also agreed the proposed service oriented architecture that consists of ASC and CM component will be a good idea to integrate different applications as a whole to automate business processes.

- The final part of the survey questionnaire, Section F, was used to collect other perspectives regarding to the survey topics. The captured information was related to the problems while integrating systems, limitations of existing systems, and the beneficial of system integration.

### 4.5 Chapter Summary

Population of the survey was defined as personnel who are working in companies throughout Peninsular Malaysia and Singapore, and the companies must have implemented more than a single computerized system. Two sampling techniques were used to collect the survey data, judgment and random sampling. Judgment sampling is type of non-probability survey sampling technique which target on the representatives who involved in supply chain activities at their current or previous working environment. Random sampling is common type of probability survey sampling technique. In this survey, random sampling was conducted via an online survey.

The analysis of the survey was based on the responses by the judgment and random respondents. The reason judgment sampling is employed is due to wide coverage of the research population. In order to avoid certain levels of sampling error such as uncovered population elements through judgment sampling, random sampling is applied to facilitate the survey.
According to 22 judgment respondents and 18 random respondents, the level of system integration among the companies was still quite low even though most of them were agreed with the importance of system integration. In order to determine the acceptance of system integration among the companies, the survey was focuses on the perceptions of respondents on system integration. Not surprisingly, most of the companies had plans or is implementing system integration by purchase or develop middleware programs. However, the limitations like lack of integration experts, complexity of legacy systems and high integration cost and time consuming were among the barriers to integration success.

The majority of both judgment and random respondents of this survey (over 85 percents) agreed that ERP, CRM and SCM are most suitable enterprise systems to be participated in the proposed research integration system. Hence, the respondents also agreed two components of the proposed service oriented methodology (ASC and CM) will be a good idea to effectively increase the integration level of ERP, CRM and SCM systems.
CHAPTER 5: SYSTEM DEVELOPMENT

5.1 Introduction

In this chapter, the concepts of the proposed system integration methodology - Serviced Oriented Conceptual Adaptive Integration Methodology (SCAIM) is developed. A subsystem of each of ERP, CRM and SCM system are selected respectively to be participated in the system integration development. Two important components of SCAIM - Conceptual Model (CM) and Adaptive Service Connector (ASC) are constructed based on the service-oriented architecture.

The development of the proposed integration system methodology is based on the traditional information system development life cycle, where several common phases like requirement analysis, system design, system implementation and testing are followed.

5.2 System Requirement Analysis

System requirement analysis is a problem-solving technique that documents the detail of system service, features and constrains. The preliminary objective of the requirement phase is to decompose the proposed integration system into components parts in order to determine the capability of the components and how well it beneficial to the business needs. Based on the discussions in previous chapters, requirement phase lays out the important concepts to capture the functional and non-functional requirements in order to drive the architecture design decision of the proposed methodology.
5.2.1 Functional Requirements

Functional requirements defined as description of the needs and desires that must be included in the proposed integration system to satisfy business needs of an extended enterprise. The main goal of functional requirements is to capture the functions, features or services that are required to perform by the system. Five main modules of SCAIM integration methodology are included:

1. ERP Module

As mentioned in previous chapters, ERP system is selected to be one of three participants of the proposed integration methodology. In view of this, ERP module mainly focuses on the development of services shared by ERP system and the concepts it represented. Sub-system of ERP selected for the research prototyping is Sales and Distribution System (SDS). Through the integrated system users are able to access two main services of SDS, which are Outstanding PO List and Customer Delivery Status Listing.

2. CRM Module

Another participant of the proposed integration methodology is CRM system. CRM module is mainly focuses on the development of services shared by CRM system and the concepts it represented. As mentioned in chapter two, the concept of CRM system could be very wide and it consists of a lot of sub-systems or modules. In this case, sub-system of CRM selected for the research prototyping is Customer Services and Support (CSS). Two services of CSS to be developed for sharing are Product Satisfaction Level which is rated by customers; and Customer Pre-Order Listing.
3. SCM Module

SCM system plays similar roles with ERP and CRM system in the proposed integration methodology. SCM module is focuses on the development of services to be shared to other two participated modules. Due to widely concepts of SCM system, only one subsystem of SCM is selected to be participated, which is E-Procurement System (EPS). Hence, two services to be developed for sharing are Quotation from Suppliers and New Products Information.

4. Conceptual Model Module

This is one of two core module of SCAIM integration methodology that provides a framework to support service sharing among ERP, CRM and SCM systems. System participant such as SDS of ERP system, CSS of CRM system, and EPS of SCM system is registered and stored into a database of this module, which called service dictionary. The database also used to store all the global concepts and services offered by each of ERP, CRM and SCM system. Only services and that maintained in this module are able to be requested by other integration participant.

5. Adaptive Service Connector Module

Another core module of SCAIM integration methodology, Adaptive Service Connector (ASC) module provides facilities for integration participants to request and share services within the integration system network. Each integration participant should be refers to single ASC respectively, which is installed within the respective system network. In order to identify a requested from and to a particular participated system, ASC should be able to maintain the local services and concepts that provided by the system. Hence, the main function of ASC module is conceptual mapping, which responsible to mapping the local concepts of the respective system to the global
concepts that understandable by the conceptual model. This function is important whenever a service of the system is requested by or share to other system.

5.2.1.1 Use Case Diagrams

Use case diagram is usually used to modeling functionalities of a system, in terms of the business or IS events, how the system responds and who organize the events. In this research, Unified Modeling Language (UML) Use Cases are chosen to model the integration services or functionalities that enable the interactions between ERP, CRM and SCM. Figure 5.1 depicts all the use cases that capture the events of the integration system.

![Figure 5.1: SCAIM Integration System Use Case Diagram](image)

Figure 5.1: SCAIM Integration System Use Case Diagram
There are a total of 14 use cases and 4 actors of the proposed integration system. Four involved actors are the company staff, customer, supplier, and integration maintainer. The main user, the integration maintainer is responsible to control the conceptual model and adaptive service connector modules, which involve events like Maintain Systems List, Maintain Global Services and Concepts, and Maintain Local Services and Concepts. Maintain Global Services, Maintain Local Services, and Maintain Local Concepts are included use case of another three respective use cases, which are Maintain Service Addresses, Map Services and Map Concepts.

In order to depicts the interaction between three integration participants, user of each ERP, CRM and SCM system (company staff, customer, and supplier) are call the two services that provided by another two systems respectively. The services include Check Outstanding PO and View Delivery/Service Status provided by ERP, List Product Satisfaction and Check Customer Pre-Order provided by CRM, and, Request for Quotation and View New Product Information provided by SCM system. The detail elements of each use case are explained in Appendix B.

5.2.2 Non-Functional Requirements

Non-functional requirements, as the name suggest, are those requirements that are not directly concerned with the specific functions delivered by the system. It may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces. The non-functional requirements of SCAIM system are:

- **Security**

  The security feature enables the system to meet the security requirements for storage, communication and displaying of information. Services shared by
particular systems are confidential to different group of stakeholders and need to be protected against accessed by unauthorized personnel. Besides, the research system must authorize specific users to access particular functionalities. It may achieved by implementing encryption method like PKI, SSH, or SSL over the intranet and internet. The Security component of Conceptual Model plays a major role in ensuring the security of the system.

- **Quality of Service**
  The methodology system should achieve certain level of quality for both service provider and consumer. The major issue concern for quality of service includes availability, accessibility, integrity, and reliability of shared services.

- **Usability**
  Appropriate graphical user interface and adequate documentation will enable usability of a system. The proposed system should provides usability by designing user-friendly interface, and easy to use and understand user manual for users.

- **Performance**
  Response time is still a great challenge for distributed environment as transmits of data over Internet or Intranet. This is still slow is compare to Local Area Network, client-server or standalone machine processing. The response time to retrieve the data and information should be in a reasonable internal time. This means that all shared services should be available to users at any point in time.

- **Maintainability and Expandability**
  The system should be designed to be easy to maintain, locate and fix a technical error. Besides that, architecture components, algorithm, data structure and procedures design should be able to be extended and modify with ease. This is important because any future enhancement and expansion can be done easily.
5.3 System Design

In system design phase of system development life cycle, the information gathered during the earlier stage is used to accomplish the logical design of the research system. In this session, system design of Service-Oriented Conceptual Adaptive Integration Methodology (SCAIM) research system is represented using Unified Modeling Diagram (UML) such as Sequence Diagram, Class Diagram and Entity Relationship Diagram. In short, system design of SCAIM includes architecture design, functionality design, component design, database design and interface design.

5.3.1 SCAIM Architecture Design

System architecture design describes in detail about the components structure of SCAIM integration system, and the way each of the components interact between each other. Architecture design also includes process of documenting the system architecture facilities based on the requirement analysis and information from the early stages.

The first step of architecture design is to identify the general modules of the SCAIM integration system. The core modules that built up the integration system are Conceptual Model Module, Adaptive Service Connector Module, ERP, CRM and SCM Module. Figure 5.2 depicts the architecture design of whole SCAIM integration system. Following sub sections describes the functionality of each module and how the high level design facilitates communication between each module.
5.3.1.1 Conceptual Model Module

Conceptual Model (CM) is one of the most important modules to operate SCAIM integration system. All conceptual attributes and respective shared service available are maintained through this module. Of whole architecture of SCAIM, only a single CM is to be implemented for the operation of whole integration system.
As depicted in Figure 5.2, sub-components of CM are Service Management, Configuration Framework, Service Broker (includes Addressing, Messaging, Routing, Invocation, Security and Standard), and Service Dictionary and UDDI. The functionality of each of the sub-components is described in table below.

**Table 5.1: Sub-Components of Conceptual Model**

<table>
<thead>
<tr>
<th>Sub-Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Management</td>
<td>Responsible to monitor and maintain shared services available to be invoked by participated systems. The component also used to maintain global concepts respective to services.</td>
</tr>
<tr>
<td>Configuration Framework</td>
<td>Web services, WSDL addresses, policy and schemas are maintained and managed in this component.</td>
</tr>
<tr>
<td>Addressing</td>
<td>Responsible for manage addressing information of source and destination, message and URI location of web services. It is also used to identify and secure web service endpoints by defining XML Namespace elements.</td>
</tr>
<tr>
<td>Messaging</td>
<td>Responsible for transmission and processing of XML messages. The functions of message transmission include message logging, queuing, delivery acknowledgement, message prioritization, message timeout, and message enhancement.</td>
</tr>
<tr>
<td>Routing</td>
<td>Mainly used to process incoming sender SOAP messages by redirect the message to alternative servers that are not accessible by the sender.</td>
</tr>
<tr>
<td>Invocation</td>
<td>Support synchronous and asynchronous communication operation types between service provider and consumer.</td>
</tr>
<tr>
<td>Security</td>
<td>Responsible for security of whole CM and ASC. It supports functions like message-level security, service authorization and authentication, transport-level security, and encryption. Window based authentication and form based authentication are supported by this component.</td>
</tr>
<tr>
<td>Standard</td>
<td>The component defines the standards of EDI to be supported by the integration model. The defined standards affect the output format of supply chain file like outstanding PO, goods received note, and quotation.</td>
</tr>
<tr>
<td>Service Dictionary</td>
<td>Responsible to store information regarding systems, global services and concepts. It is served as registry to publish the available services to the Internet for public use.</td>
</tr>
</tbody>
</table>

**5.3.1.2 Adaptive Service Connector Module**

In this research, three ASC are required for the integration system, as one for facilitate ERP, one for CRM and the other for SCM system. Each of the ASC maps the global concepts from CM with local concepts to complete the integration processes. Four main sub-components of the ASC are Local Service Management, Conceptual Mapping,
Transformation, and Output EDI/CSV files. The description of each component is shown in Table 5.2.

### Table 5.2: Sub-Component of Adaptive Service Connector

<table>
<thead>
<tr>
<th>Sub-Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Service Management</td>
<td>This component is mainly used to maintain the local service and concepts provided by its respective participated system. The component also used to invoke particular service of its belonging local system whenever it is requested by other system, and then converted the result back to the CM. Besides, it manages process of invoking external services provided by other system.</td>
</tr>
<tr>
<td>Conceptual Mapping</td>
<td>Responsible to map local services to global services stored in CM, including mapping the related concepts.</td>
</tr>
<tr>
<td>Transformation</td>
<td>Responsible to convert the result data received from CM to EDI or CSV standard business file format.</td>
</tr>
<tr>
<td>EDI/CSV output</td>
<td>As final output file as requested and stored in file server before being downloaded by the service requester.</td>
</tr>
</tbody>
</table>

### 5.3.1.3 ERP, CRM and SCM System Module

ERP, CRM and SCM modules represent the participants of the SCAIM integration system architecture. The modules are designed to be 3-tier architecture, where most of the enterprise systems in the worldwide are preferable to be implemented on this kind of architecture. Three tiers are generally means to Application Server, Database, and the Application Client. In SCAIM architecture, Application Server can be described by explanation of four sub-components, which are Service Interface, User Interface, Façade Layer, and Business Component. Table 5.3 explains the functionality of each component.
Table 5.3: Sub-Component of System Module

<table>
<thead>
<tr>
<th>Sub-Component</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Interface</td>
<td>Ensure interoperability among applications by defining WSDL contract between service provider and consumer. It also using code behind to verify and relay incoming XML messages to ensure interoperability among systems.</td>
</tr>
<tr>
<td>User Interface</td>
<td>Provide interface to allow application user to invoke external services that are not belonging to the particular system.</td>
</tr>
<tr>
<td>Business Component</td>
<td>Codes related to system business logic are placed in business component instead of service interface.</td>
</tr>
<tr>
<td>Data Access Component</td>
<td>Responsible to interact with the service provider system backend data store.</td>
</tr>
</tbody>
</table>

SCAIM integration system is designed based on SOA where the systems are communicated through services. The term service can be defined as endpoint of a connection between service provider and consumer. In SCAIM architecture, services are originally based on the existing functionalities of ERP, CRM or SCM system, or it can be re-amended purposely to be invoked by other system. Each of the functionality is a set of actions that applied the business logic in business component, which stated in the application server.

5.3.1.4 ERP, CRM and SCM Integration through the Architecture

In order to better understanding the background philosophy of the proposed architecture, an example of scenario regarding interoperability problems among ERP, CRM and SCM is explained.

For an example of a grocery and general merchandising retail chain that has implemented ERP system in their company for several years. The company is focusing in consumer goods like electronics, computers, foods, health, home, car, and other groceries. Hence, the company also provides services on their electronic products and manufacture computer accessories.
Although ERP system is effective for planning the resources within the company, but it does not have any method to assess customer satisfaction. To enhance the function, internal staffs require assistance from the CRM system that provides service called \textit{Product Satisfaction Level} to collect customers’ feedback regarding to particular products and services, and other remarks. The price of the products as provided by ERP sometimes is always out-to-dated, and the internal record of prices does not following the changes of market price for products. In this case, a service called \textit{Product Quotation} offered by the SCM system can be shared to internal staff to check the latest prices of products and services.

Typical CRM system provides less information regarding the actual delivery or service status. To overcome this issue, the ERP system that consists of the information about the progress status like Work in Progress (WIP), Received from Production, On Shipping, Reach at Warehouse, which is able to provide the customer about the information. The service shared by ERP to CRM system is called \textit{Delivery/Service Status}. Besides, in order to better understand the actual needs of customer, existing products or services may not fulfill their requirements. The customer may request to get more information regarding new products or existing product that does not selling in the company. In this case, \textit{New Product Info} service that offered by SCM e-tendering function that allows users to check the products information supplied by particular supplier which is not covered in CRM system.
Although SCM system provides functionalities for design, planning, and controlling the operations of supply chain, but the suppliers are not able to get the up-to-dated information regarding the outstanding PO of their servicing companies. This service is usually provided by a typical ERP system where users are able to refer a Goods Received Notes (GRN) details regarding outstanding PO and product received quantities by their servicing companies. A service called Outstanding PO is available to be shared by the ERP system. Thus, consider another drawback of SCM system where most of the suppliers do not have idea to foreseen their customer order for the next months through SCM system. In this case, CRM provides a service named Customer Pre-Order to calculate the customer pre-order for next months.

5.3.2 SCAIM Functionality Design

After having overview of core modules of SCAIM architecture, functionality of each module is discussed. As mentioned in Section 5.2.1.1, total of 14 functions provided by five main modules in previous section. The activities flows of several functions are described by using UML Sequence. Each sequence diagram is corresponding to the use case explained in Section 5.2.1.1.

Integration Maintainer can list, add, edit, and delete integration participated systems from the service dictionary. The related sequence diagram is shown in Figure 5.3. Besides, Integration Maintainer maintains the global concepts in the service dictionary through Main Page, Concept List Page, Add/Edit Concept Page interfaces. Integration Maintainer maintains global services through CM module as shown in Figure 5.4 sequence diagram.
Figure 5.3: Maintain Systems List Sequence Diagram
Figure 5.4: Maintain Global Services Sequence Diagram
After particular global service is maintained, its respective service address is can be added or modified. The interfaces for input are Main Page, Service List Page, and Add/Edit Service Address Page. The records will be updated to Table “GlobalService”.

The first function provided by ASC Module is to maintain local concepts list, which provided by a particular participated system. Interfaces involved are Main Page, Concept List Page, and Add/Edit Concept Page. The sequence of activities is depicted in Figure 5.5. Besides, maintain local services are another function provided by ASC module. Integration Maintainer list, add, edit, or delete existed services provided by the particular system through Main Page, Service List Page, and Add/Edit Service Page interfaces.

After local and global concepts are added, Integration Maintainer maps the global concepts to its equivalent local concepts. The processes involve entry through interface like Main Page, and Concepts Mapping List Page, and finally the Table “ConceptMap” is updated. The sequence of the processes is shown in Figure 5.6.

The last function involves Integration Maintainer is mapping the services. The function is started only after local services are well maintained. The mapping processes are done through Main Page, and Services Mapping List Page interfaces. Table “ServiceMap” is updated at the end of the processes.
Figure 5.5: Maintain Local Concepts Sequence Diagram
Figure 5.6: Map Concepts Sequence Diagram
Figure 5.7: Check Outstanding PO Sequence Diagram
Check Outstanding PO is one of the functions to facilitate interoperability between SCM and ERP systems, where supplier invokes service of ERP through SCM system. Supplier checking outstanding PO or GRN details of products through interfaces like Main Page, External Service Page, and Outstanding PO Page of SCM system. Table “GRN” of ERP database and tables of Service Dictionary are selected and modified during the processes of the function. Sequence activities of the function are depicted in Figure 5.7.

List Product Satisfaction is another function to facilitate interoperability between ERP and CRM systems, where company staff invokes service of CRM through ERP system. Staff collects information regarding customer satisfaction level against products or services through interfaces like Main Page, External Service Page, and Product Satisfaction Page of ERP system. Table “ProductSatisfaction” of CRM database and tables of Service Dictionary are selected and modified during the processes of the function, as shown in Figure 5.8.

Request for Quotation is another function to facilitate interoperability between ERP and SCM systems, where company staff invokes service of SCM through ERP system. Staff requests for prices, information and quotation for a particular products from suppliers through interfaces like Main Page, External Service Page, and Product Quotation Page of ERP system. Table “Quotation” of SCM database and tables of Service Dictionary are selected and modified during the processes of the function, as shown in Figure 5.9.
Figure 5.8: List Product Satisfaction Sequence Diagram
5.3.3 Integration Success through Conceptual and Service Mapping

It's has been proved that three participants of SCAIM integration system - ERP, CRM and SCM systems have fulfilled the integration prerequisites, which have many similarities regardless from representation or operation perspectives. Due to the similarities, two drivers to the integration success - conceptual equivalence (for semantic interoperability) and execution knowledge (for functional interoperability)
could be achieved through any architecture that fulfills the prerequisites. The simple diagram demonstrates the integration prerequisites are shown in Figure 2.2.

In this section, Conceptual and Service Mapping are discussed on how the integration prerequisites could be achieved through SCAIM architecture. Generally, Conceptual Mapping is a method to ensure the exchanged information between the service requestor and provider has the same meaning. It is done by using the ASC module of SCAIM to map the local concepts to the respective global concepts within the integration domain. Conceptual Mapping has the ability to improve semantic interoperability among the participating systems.

Every participating system is ready to share the internal service to other participating systems. However, it is impossible for service requestor to direct access other system services due to privacy and confidential issues. Because of this, Service Mapping is used to map the private local services of provider to the public global services that store in the CM module. Service Mapping is targeting on improve service interoperability among systems being integrated. To better understanding on how the integration works through Conceptual and Service Mapping, a scenario of service sharing involves ERP (service provider) and SCM (service consumer) is discussed.

The E-Procurement module (EPS) of SCM system do not provides any functionality for supplier to check information related to outstanding purchase orders and good received notes. Upon a trading partner company of the suppliers is decided which supplier to be selected for products and purchase order is issued, the next business process for the supplier is to check the delivery status. The SCM user (the supplier) requests for the Outstanding PO service through CM module to search for the global services. The CM
module determines Outstanding PO service is provided by ERP system that requires Supplier Identifier and Vendor PO Number as input global concepts. The ASC module for SCM is then determines whether it has the equivalence local concept with Supplier Identifier or Vendor PO Number global input concept required by the service. This process is done through the Conceptual Mapping.

Once the ASC module for SCM detected the requestor support input global concept, it tells CM module to precede the service request. At this moment, Sales and Distribution module of ERP receives request from CM module and ASC for ERP uses the Conceptual Mapping to map the global input concept of CM to local input concept of ERP system. In this example, attributes like Supplier Identifier are mapped to the SupplierGlobalLocation (SCM) and SupplierGLN (ERP) local concepts. Figure 5.10 depicts the Conceptual Mapping for shared service of Outstanding PO.

**Figure 5.10: Conceptual Mapping for Outstanding PO Global Service**

After the Conceptual Mapping for SCM is complete, CM module then uses Service Mapping to map the requested global service (Outstanding PO) with the ERP local service. To consume the local service, web service is used to communicate with the ERP business component static methods, and then process the data through data access layer before returning result to the CM module. By using .NET reflection technology, SCAIM instructs and binds the web service dynamically to execute methods based on
configured service address in the service dictionary. Proxy class for web service is created and modified automatically at runtime and no maintenance is required for the proxy.

Before the result is return to CM module, ASC for ERP uses Conceptual Mapping to define columns or concepts of the dataset to be returned. Based on the mapped output global and local concepts in CM module, required data from ERP system is returned. In addition, the result dataset format is based on the EDI format as stated in the service broker of SCAIM. The final CSV or EDI file is downloadable through SCM web server to the system client side.

5.3.4 SCAIM Component Design

The details architecture design of the Service-Oriented Conceptual Adaptive Integration Methodology is described in previous section. In more simple form, the SCAIM system design can be depicted in multiple tiers of architecture components. The n-tiers architecture of SCAIM is consist of Conceptual Model Tier (Web Service), Adaptive Service Connector Tier (Service Mapping and Binding), Service Dictionary Tier, Application Server Tier (Act as Service Provider and Consumer, location of web services), Business Component Tier, Application Database Server Tier, Data Access Tier (To access Service Dictionary and Application Database), and Application Client Tier (User Interface).

Based on the basic component design of n-tier architecture, SCAIM development is technically divided into ten individual projects, each project will be implemented at one of the tiers of SCAIM system. Ten projects as mentioned to construct SCAIM integration system are shown in Table 5.4.
### Table 5.4: Projects to Simulate SCAIM Integration System

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) CM Web Application</td>
<td>Web application to be implemented at Conceptual Model to maintain systems list, global services, service address, and global concepts.</td>
</tr>
<tr>
<td>2) ASC Web Application</td>
<td>Web Application to be implemented at Adaptive Service Connector to maintain local concepts, local services, to map concepts and services.</td>
</tr>
<tr>
<td>3) Service Dictionary</td>
<td>Database that consists of global services and concepts, local services and concepts, services and concepts mapping information, and participated systems list.</td>
</tr>
<tr>
<td>4) System Application</td>
<td>Web Application allows users to access available external services, and download output files after service is invoked.</td>
</tr>
<tr>
<td>5) Application Database Server</td>
<td>Database of ERP, CRM or SCM system.</td>
</tr>
<tr>
<td>6) Conceptual Mapping and Service Binding</td>
<td>Library to mapping and auto-binding specific web services based on CM and ASC setting.</td>
</tr>
<tr>
<td>7) Systems Web Services</td>
<td>Web Services located at ERP, CRM and SCM application server separately.</td>
</tr>
<tr>
<td>8) Business Component</td>
<td>Business Component library located at application servers to access business logic of respective system.</td>
</tr>
<tr>
<td>9) Data Access</td>
<td>Library to access Service Dictionary and to access Application Database.</td>
</tr>
<tr>
<td>10) EDI/CSV Output</td>
<td>Final result of service request and response, either in comma separated value (CSV) of EDIFACT standard industry format.</td>
</tr>
</tbody>
</table>

CM of SCAIM consists of four basic classes - **CmSystem**, **CmGlobalConcept**, **CmGlobalService**, and **CmServiceAddress** as shown in Figure 5.11. The classes are respectively represents the functionalities of CM component, as described in previous section of SCAIM functionality design.
Two basic classes of ASC component of SCAIM, **AscLocalConcept** and **AscLocalService** play important roles to fulfill the requirements of the functionalities provided by the component. There is no linking between two classes in the package, as shown in Figure 5.12.

Figure 5.13 depicts class package of Business Component of SCAIM integration system. The package consists of six classes in general, which are **BcERPOutstandingPO**, **BcERPDeliveryStatus**, **BcCRMPProductSatisfaction**, **BcCRMCustomerPreOrder**, **BcSCMProductQuotation**, and **BcSCMNewProductInfo**. Each of the class is represents the conceptual business layer (middle layer) that manages
a single business abstraction. For example, **BcERPOutstandingPO** is a business component to manage a requested service to retrieve outstanding purchase order which is provided by ERP system.

![Class Package for Business Component](image)

**Figure 5.13: Class Package for Business Component**

Classes like **MapConcepts**, **MapServices**, and **ServiceBinding** are used to represent the modular structure of Conceptual Mapping and Service Binding project. **MapConcepts** and **MapServices** represent the functionalities of the component, while **ServiceBinding** class is used to bind the configured services to the actual web services located in the participated systems. The classes are depicted in a package as shown in Figure 5.14.
5.3.5 Database Design

Database design involves the activity on modeling the structure of database that will store and maintain data records. It will also include the transformation of user’s processing requirements and unordered information into proper functional requirements. The SCAIM integration system is based on a relational data model to keep and manipulate data.

5.3.5.1 Database Structure

As explained in component design, SCAIM has two important database tiers to store temporary and permanent information required for operation of the integration system. In this section, Entity Relationship (ER) Model is used to describe the type of information that is to be stored in Service Dictionary. It helps the integration system developers to interpret different view of information from different perspectives especially during the implementation of the database.
In order to prove that the proposed SCAIM integration system supports different platform of participated systems, a heterogeneous tools of Database Management System (DBMS) have been used to simulate System Databases Tier. For ERP, Microsoft Office Access 2007 is selected as the DBMS, while Microsoft SQL Server 2005 for CRM and MySQL for SCM System. Hence, Microsoft SQL Server 2005 is selected as the DBMS for Service Dictionary Tier.

Figure 5.15 depicts the Service Dictionary database design by using ER diagram. The database consists of nine tables – User, System, GlobalConcept, GlobalService, ServiceConcept, LocalConcept, LocalService, ConceptsMap, and ServicesMap Table.
5.3.5.2 Data Dictionary

Data dictionary is defined as a centralized repository of information which consists of the structure of all data types. Table description for SCAIM Service Dictionary, ERP, CRM, and SCM database can be found in Appendix C.

5.3.6 SCAIM Interface Design

User interface design is a very important part of SCAIM integration system design as this is further verified when system designers and user share the same opinion. In general, the interface design is to ensure the user and system interaction as friendly and efficient as possible. Besides, there are several more specific objectives of interface design to be achieved:

- **Effectiveness** means how well the interface is able to interact with the user. This is measured by how well users can accept and adapt to the interface to perform tasks
- **Productivity** is attained by designing the user interface based on ergonomically proven methods
- **Simplicity** means how simple and common the tasks can be carried out through system interface based on user’s own language, and effective shortcut is provided for experience user
- **Tolerance** means how well the interface design allow user to rollback and commit any actions in order to prevent errors or mistakes.

5.3.6.1 Input Design

The quality of system input determines the quality of system output. This concept is known as “Garbage in, garbage out (GIGO)”. Therefore, a well designed input is needed to ensure the quality and accuracy of data.
Forms are the best way to capture data. To make the form more attractive, graphical user interface can be used. Forms have to be very user friendly so that the data is easier to capture and is accurate. The correct use of form controlling ensures the success of the capturing process. The form controlling includes text box, radio button, check box, list box and drop-down list and so on. Figure 5.16 shows one of the examples input design of SCAIM integration system.

![Sample Input Screen - Register New Local Service of ERP](image)

**Figure 5.16: Sample Input Screen - Register New Local Service of ERP**

5.3.6.2 Output Design

The design of output serves the purpose of providing the information that the user needs, based on the information criteria selected by them. The identified output will be related to the service invoked or requested by the user. These outputs will be presented in the form of EDI, as shown in Figure 5.17.
5.4 System Implementation

According to Avison, D. (2006), technical specification is converted into executable software component, programs or information system during the system development phase. The technical specification and algorithm that defined in component design (Section 5.3.4) are the key input to the implementation phase of SCAIM integration.

The deliverable scope and final goal of system implementation is to realize the technical aspect of each of the 10 projects which represents each tier of the SCAIM components, as defined in Table 5.4.

5.4.1 Conceptual Model Implementation

Basically, Conceptual Model can be implemented by developing a web application that provides four core functionalities; maintain system list, global services, service address and global concepts. In other words, the application represents CM components such as Service Management, Configuration Framework, and Security as discussed in Section 5.3.1.1. Figure 5.18 illustrates web forms and classes developed by using VB.NET in Visual Studio solution explorer view.
Seven web forms were developed to perform the interfaces for integration maintainer to perform four core functions. The web forms to execute the functionalities are:

- **wfSystemList.aspx** and **wfSystem.aspx** implement maintain system list function
- **wfGlobalServiceList.aspx** and **wfGlobalService.aspx** implement maintain global service function
- **wfGlobalConceptList.aspx** and **wfGlobalConcept.aspx** for maintain global concept function
- **wfGlobalServiceAddress.aspx** for service address maintenance function

**Figure 5.18: CM Web Application in view of VS Solution Explorer**

Besides the functionalities, implementation of CM web application also involves the security issues. As discussed in the architecture design, the CM component of Security supports both windows based and form based authentication. The coding shown in table 5.5 illustrates form based authentication method.
Table 5.5: Codes for CM Form Based Authentication

```vbnet
Public Class clsAccessControl
    Public Function AuthenticateUser(ByVal username As String, ByVal password As String) As Boolean
        Dim exist As Boolean = False
        Dim existmsg As String = "Failure"
        Dim pwd As String = ""
        Try
            exist = VerifyPasswordByUsername(username, pwd)
            Catch e As Exception
                End Try
            If exist Then
                If pwd <> password Then
                    exist = False
                    Throw New Exception("Fail to authenticate user. Invalid password.")
                End If
            End If
            existmsg = "Success"
        End If
        Else
            Return True
        End Function 'AuthenticateUser

    Private Function VerifyPasswordByUsername(ByVal username As String, ByRef password As String) As Boolean
        Dim sql As String
        Dim objDB As New clsDatabase
        Try
            sql = "SELECT Password FROM [User] WHERE Username=": SQLStr(username)"
            With objDB.OpenDataReader(sql)
                If Not .Read Then
                    password = ""
                    Return False
                Else
                    password = .Item("Password").ToString()
                    Return True
                End If
            End With
        Catch e As Exception
            Throw e
        Finally
            objDB.CloseDataReader()
            objDB = Nothing
        End Try
    End Function
End Class
```
Generally, the `clsAccessControl.cls` public class is used to authenticate entered user name and password by system user to the valid users stored in the database. The class is passed by when entered information is matched with the database and user is able to access the application.

In order to implement windows based authentication to access CM application, the first step is to configure IIS server. The steps to follows as:

- In the IIS right panel, right click the CM application’s virtual directory and select `Properties`.
- Select `Directory Security` tab. Click on `Edit` button under `Anonymous access and authentication control`.
- Select `Integrated Windows authentication` and uncheck all other checkboxes.

After IIS is configured, the next step is to set `web.config` file of the application to Windows mode, which include codes in Table 5.6. After the basic configuration, only authenticated users from specific domain are able to access the application.

<table>
<thead>
<tr>
<th>Table 5.6: Setting <code>web.config</code> File to Support Windows Authentication Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;system.web&gt;</code>&lt;br&gt;<code>&lt;authentication mode=&quot;Windows&quot;/&gt;</code>&lt;br&gt;<code>&lt;authorization&gt;</code>&lt;br&gt;<code>&lt;deny users=&quot;?&quot;/&gt;</code>&lt;br&gt;<code>&lt;/authorization&gt;</code>&lt;br&gt;<code>&lt;/system.web&gt;</code></td>
</tr>
</tbody>
</table>

Complete interfaces for CM and description of each process flow can be found from the `Appendix D` of this report while source codes are available from `Appendix E`. 
5.4.2 Adaptive Service Connector Implementation

Adaptive Service Connector can be implemented by developing a web application that support four basic functionalities - maintain local concepts, local services, map concepts and services. Hence, the implementation fulfills the requirements of ASC components like Local Service Management, and Conceptual Mapping as discussed in Section 5.3.1.2. Figure 5.19 depicts web forms and classes developed by using VB.NET in view of Visual Studio solution explorer.

![Figure 5.19: ASC Web Application in view of VS Solution Explorer](image)

Six web forms were developed to perform the interfaces for integration maintainer to perform four functions of ASC application. The web forms to execute the functionalities are:

- `wfLocalConceptList.aspx` and `wfLocalConcept.aspx` implement maintain local concept function
- `wfLocalServiceList.aspx` and `wfLocalService.aspx` implement maintain local service function
- `wfMapConcepts.aspx` for map concepts function
- `wfMapServices.aspx` for mapping services function

Besides the forms, another web form called `wfInvokeService.aspx` is acts as intermediate connector that allows system application to invoke external service provided by other system. The details process description will be discussed in Section 5.4.10. As similar to CM web application, ASC uses the same method of form based and windows based authentication for the security. Only authenticated users are able to access the system.

### 5.4.3 Service Dictionary Implementation

The implementation of service is purposely used to realize the technical specification of the storage of global services and concepts, local service and concepts, service and concepts mapping information and integration participated systems list. Attributes, tables, and their relationship are created in MS SQL Server Management Studio. Figure 5.20 illustrate few simple steps to create tables, properties and security in SQL Server.

Based on ER diagram and table description as designed in Figure 5.15 and Appendix C, the relationships of the service dictionary tables can be implemented by creating a Database Diagram in SQL Server Management Studio, as depicted in Figure 5.21. The schema of the tables is flexible to be modified at anytime through the IDE.
Figure 5.20: Create Tables and Attributes through SQL Server

Figure 5.21: Service Dictionary ER Diagram in SQL Server Database Diagram
5.4.4 ERP, CRM and SCM Systems Implementation

This section focuses on the implementation of service provider and consumer. To simulate the implementation, one of the global services called Outstanding PO is chosen. The service provider is sales and distribution of ERP system, while service consumer is e-procurement of SCM system. The process explanation has been discussed in Section 5.3.1.4.

![Figure 5.22: Interface of SCM to Invoke a Service of ERP](image)

As shown in Figure 5.22, SCM system is incorporated with screen allows SCM users to invoke any service available in SCAIM integration domain. In the real industry practices, the interface can be installed in the existing system server in condition that the server is able to access to the ASC web application for communication purposes. The communication between ASC and SCM system is hardly rely on the web configuration setting as shown in Table 5.7.
Table 5.7: SCM System Configuration Setting

```xml
<appSettings>
<!-- Setting to Connect SCAIM -->
<add key="ConnectionString_SD" value="server=JACKY-PC\SQL2K5;database=SCAIM_ServiceDictionary;uid=APPLUSER;pwd=APPLUSER"/>
<add key="SCM_SystemCode" value="SCM_EPS"/>
<add key="ASC_IISURL" value="http://SCM_Server/SCAIM_ASC"/>
</appSettings>
```

The statement of “ConnectionString_SD” is used for existing system to get connected to the settings stored in the service dictionary of CM domain. As service consumer, SCM is hardly rely on this connection to retrieve service and concepts settings before a service request is send to the service provider.

In order to get ready for service sharing, service provider (ERP system) is performed with several settings, as shown in Table 5.8. Firstly, the statement of “ConnectionString” ensures the system is connected to its own database. In this example, ERP is using MS Access Database named SCAIM_ERP.accdb. Meanwhile, the statement of “ConnectionString_SD” is used to retrieve concepts and services settings. It enables ERP to retrieve fields from its own database based on the settings and return the result to the service requester (SCM system).

Table 5.8: ERP System Configuration Setting

```xml
<appSettings>
<!-- Existing System Setting -->
<add key="ConnectionString" value="Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\Dissertation WXGA 6181\Development\Databases\SCAIM_ERP.accdb;Persist Security Info=False;"/>
<!-- Setting to Connect SCAIM -->
<add key="ConnectionString_SD" value="server=JACKY-PC\SQL2K5;database=SCAIM_ServiceDictionary;uid=APPLUSER;pwd=APPLUSER"/>
<add key="ERP_SystemCode" value="ERP_SDS"/>
</appSettings>
```
5.4.5 Systems Database Implementation

The implementation of ERP, CRM and SCM database are based on the data dictionary as defined in Appendix C. As defined in Database Design (Section 5.3.5), ERP database represented by using MS Office Access, while SQL Server and MySQL represents the databases of CRM and SCM respectively. The purpose of the implementation is to validate the capability of SCAIM in supporting the heterogeneity data storage platform. Previous section has discussed about the implementation of ERP, CRM and SCM systems and how its work with their respective database. For ERP, a MS Access database named \textit{SCAIM\_ERP.accdb} is created to simulate its objects, attributes, and the relationships. Figure 5.23 shows the environment of the database being developed.

![Figure 5.23: ERP Database in MS Access Environment](image)

As for CRM, a SQL Server database named \textit{SCAIM\_CRM} is created to simulate detail view of CRM data storage. Figure 5.24 shows the database developed by using SQL Server Management Studio.
Besides, SCM MySQL database is developed by using one of the third party freeware IDE called Toad provided by Quest Software.
5.4.6 Conceptual Mapping and Service Binding Implementation

The implementation is simply to realize the theory and scenario of service sharing as discussed in Section 5.3.3. In this case, the SCM user (the supplier) is intend to invoke a service called *Outstanding PO*, which is an existing functionality of ERP systems. As prerequisites to invoke the function, the SCAIM integration maintainer have to complete the setting prepared by CM and ASC web application as discussed in Section 5.4.1 and 5.4.2. The complete user guidance on the prerequisites like maintain systems list, global service and concepts, service address, local services and concepts, mapping services and concepts can be refer to Appendix D.

After the prerequisites configuration is done, interface as discussed in Section 5.4.4 is available for SCM user to invoke any external global services. As refer to Figure 5.22, the required input concepts and output fields can be viewed at a glance based on the configuration. When the supplier invokes a particular global service, *Outstanding PO* for instance, the page will be redirected to another interface of ASC web application as shown in Figure 5.26 below.

![Figure 5.26: Interface of ASC for Service Requesting Based on Configuration](image-url)
As shown in the figure, there are two row of input parameters are auto generated by this interface, which is based on the prerequisite configuration. In other words, the code behind the screen is object-oriented and there are no hard-codes needed for the construction of this module. Table 5.9 shows how a core private function called \texttt{pnlInputParameter\_InitializeControl} to achieve the purpose.

Table 5.9: Source Codes to Implement Object Oriented Input Parameter

```vbnet
Private Sub pnlInputParameter\_InitializeControl()
    Dim objDB As New clsDatabase, strSQL As String, intRowCount As Integer = 1
    Try
        strSQL = "SELECT SystemCode+LocalConcept AS ReferenceName FROM ServiceConcept"
        strSQL &= "WHERE SystemCode+LocalService = '" & Session.Item("InvokeService\_SystemCode\_LocalService") & "' "
        strSQL &= "AND InputOutput = 'Input'"
        strSQL &= "ORDER BY ColumnSequence"
        With objDB.OpenDataReader(strSQL)
            While .Read
                Select Case intRowCount
                    Case 1
                        pnlSearchType1\_1.Visible = True
                    Case 2
                        pnlSearchType1\_2.Visible = True
                    Case 3
                        pnlSearchType1\_3.Visible = True
                    Case 4
                        pnlSearchType1\_4.Visible = True
                    Case 5
                        pnlSearchType1\_5.Visible = True
                    Case 6
                        pnlSearchType1\_6.Visible = True
                    Case 7
                        pnlSearchType1\_7.Visible = True
                    Case 8
                        pnlSearchType1\_8.Visible = True
                    Case 9
                        pnlSearchType1\_9.Visible = True
                    Case 10
                        pnlSearchType1\_10.Visible = True
                    End Select
                    pnlSearchType1\_RowControlsInitialize(intRowCount, .Item("ReferenceName").ToString())
                    intRowCount += 1
                End While
            End With
        Catch ex As Exception
        Finally
            objDB.CloseDataReader()
            objDB = Nothing
        End Try
    End Sub
```
Based on conceptual mapping between global and local concepts as depicted in Figure 5.10 earlier, the Supplier Identifier input concept in Figure 5.26 is auto represented by a dropdown list control. A list of global location numbers is selected from SCM local concept called SupplierGlobalLocation that mapped to the global concept (Supplier Identifier) earlier. The selected list is then bound to the dropdown list. As a result, more equivalence and accurate conceptual input parameters is chosen by the SCM user before invoking the service, therefore many of the integrity issues can be overcome through the implementation.

For service binding implementation, as mentioned in Section 5.3.3 earlier, .NET reflection technology is applied to enable web service being invoke dynamically based on configured service address in the prerequisite configuration. A class library called clsDynamicWSProxy.cs is created to generate the proxy for the web service to the web service at run time. By using the dynamic web service method, supplier is able to call unlimited different web services from ASC of SCM system.

In order to construct the class library to create dynamic proxy to a web service, five important functions are required, which are:

- Discover service based on WSDL file of the global service
- Build proxy class at run time
- Discover methods of the web service
- Enumerate the parameters
- Invoke web method
Discover Service

The first step to discover a web service through the proxy is to retrieve service address from service dictionary, and then pass it to the class library. As refer to Table 5.10, dynWSProxy represents an instance of the dynamic proxy that consists of parameter strServiceAddress string. Build() method is called to generate the dynamic proxy based on a service interface called IRequestService. Dynamic web service DynamicWebService is discovered through the interface.

Table 5.10: Code Snippets for Calling Dynamic Proxy Class

<table>
<thead>
<tr>
<th>Code Snippets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dim strServiceAddress As String = FKeyField(&quot;GlobalService&quot;, &quot;ServiceAddress&quot;, &quot;GlobalService = &quot; &amp; Session.Item(&quot;InvokeService_GlobalService&quot;) &amp; &quot;)</td>
</tr>
<tr>
<td>Dim dynWSProxy As clsDynamicWSProxy(Of IRequestService) = New clsDynamicWSProxy(Of IRequestService)(strServiceAddress)</td>
</tr>
<tr>
<td>dynWSProxy.UseGenericList = True</td>
</tr>
<tr>
<td>Dim IReqService As IRequestService = dynWSProxy.Build()</td>
</tr>
<tr>
<td>Dim ds As DataSet = IReqService.DynamicWebService(args)</td>
</tr>
</tbody>
</table>

Build the Proxy Class

The purpose of building the proxy class is to create source codes that will be added into web reference of the project. While the proxy class is generating, an instance of ServiceDescriptionImporter class named importer is created. The instance is used to import the basic information from the passing WSDL file into a System.CodeDom.CodeCompileUnit class which consists of a collection of assemblies being referenced by the project. Then, AddServiceDescription method is called along with the ServiceDescription object of the web service being invoked. Meanwhile, SOAP is set as the communication method to the web service by assigning it to the ProtocolName property. Code snippets for these processes are shown in Table 5.11.
Table 5.11: Code Snippets on Building the Proxy Class

```csharp
private DiscoveryProtocol _protocolName = DiscoveryProtocol.Soap;

public clsDynamicWSProxy(Uri wsdlLocation, DiscoveryProtocol protocolName)
{
    _wsdlLocation = wsdlLocation;
    _protocolName = protocolName;
    _serviceContract = typeof(IRequest);
}

public IRequest Build()
{
    ServiceDescriptionImporter importer =
        new ServiceDescriptionImporter();
    importer.ProtocolName = _protocolName.ToString();

    //Get WSDL
    DiscoverService(importer);
}

private void DiscoverService(ServiceDescriptionImporter importer)
{
    using (DiscoveryClientProtocol discoClient =
        new DiscoveryClientProtocol())
    {
        discoClient.DiscoverAny(_wsdlLocation.ToString());
        discoClient.ResolveAll();

        foreach (object value in discoClient.Documents.Values)
        {
            if (value is ServiceDescription)
                importer.AddServiceDescription(
                    (ServiceDescription)value, null, null);

            if (value is XmlSchema)
                importer.Schemas.Add((XmlSchema)value);
        }
    }
}

After the basic information is imported successfully, a method named
GenerateCodeFromNamespace is used to generate code for the proxy class. There are
three parameters are required for this method. First parameter is the namespace to use
for the proxy class named mainNamespace. Second parameter is a StringWriter named
sw to store code before compilation into assemblies, and the last parameter is optional to
indent nested blocks of code, as listed in Table 5.12.
```
Table 5.12: Source Codes onGenerating Codes for the Proxy Class

```csharp
CodeNamespace mainNamespace = new CodeNamespace(ProxyNameSpace);
StringBuilder sourceCode = new StringBuilder();
string ret;
using (StringWriter sw = new StringWriter(sourceCode,
CultureInfo.CurrentCulture))
{
    codeProvider.GenerateCodeFromNamespace(mainNamespace, sw, null);
    ret = sourceCode.ToString();
    sw.Close();
}
```

The last step in building the proxy class is to create a set of compiler parameters, and then invoke the compiler on the codes that was stored in `sw`. To achieve this, an instance of constructor for `CompilerParameters` named `parameters` is created. Assemblies need to be referenced by the compiler are added into the `parameters`. Finally, an object of `CompilersResults` is created to store the results of compiling the code using `CompileAssemblyFromSource` method, as shown in Table 5.13.

Table 5.13: Code Snippets for Compiling Codes Stored in the StringWriters

```csharp
CompilerParameters parameters = new CompilerParameters();
parameters.ReferencedAssemblies.Add("System.dll");
parameters.ReferencedAssemblies.Add("System.Xml.dll");
parameters.ReferencedAssemblies.Add("mscorlib.dll");
parameters.ReferencedAssemblies.Add(Assembly.GetExecutingAssembly().Location);
CompilerResults cr =
csharpCodeProvider.CompileAssemblyFromSource(parameters,
proxyCode);
```

Discover Web Methods

Upon the service is discovered and the proxy class is built successfully, assembly that consists of methods and properties of the web service is ready to use. The next step is to discover the web methods by using the system reflection technology. As refer to Table
5.14, method **GetMethods** is called to retrieve web methods of the web service from the compiled assembly. The results of the **GetMethods** are then stored into a **MethodInfo** object.

**Table 5.14: Source Code for Discovering Web Methods**

```csharp
private void AddImports(CodeNamespace codeNamespace)
{
    MethodInfo[] methods = _serviceContract.GetMethods();
    foreach (MethodInfo method in methods)
    {
        AddImport(codeNamespace, method.ReturnType);
        foreach (ParameterInfo parameter in method.GetParameters())
        {
            AddImport(codeNamespace, parameter.ParameterType);
        }
    }
}
private void AddImport(CodeNamespace codeNamespace, Type type)
{
    if (!type.IsPrimitive)
    {
        codeNamespace.Imports.Add(
            new CodeNamespaceImport(type.Namespace));

        if (type.IsGenericType)
        {
            Type[] genericArgs = type.GetGenericArguments();
            foreach (Type genericArg in genericArgs)
            {
                AddImport(codeNamespace, genericArg);
            }
        }
    }
}
```

**Enumerate the Parameters**

Once the methods are retrieved from the assemblies successfully, method **GetParameters** is called to return **ParameterInfo** objects such as parameters and type of parameters.
Invoke Web Method

As refer to Table 5.15, the last function to construct the class library is to call the web method. To achieve this function, `Activator.CreateInstance` is called to create instance of the type information object for the web service and return back to a service interface named `IReqService` as stated in Table 5.17 earlier.

<table>
<thead>
<tr>
<th>Table 5.15: Source Codes for Invoking Web Method by the Proxy Class</th>
</tr>
</thead>
</table>
| ```csharp
class IRequest {
    public IRequest Build() {
        using (CSharpCodeProvider codeProvider = new CSharpCodeProvider()) {
            assembly = GenerateProxyAssembly(codeProvider,
                GenerateProxySourceCode(importer, codeProvider));
        }
        //Clean up resources
        importer = null;
        return Activator.CreateInstance(assembly.GetTypes()[0]) as IRequest;
    }
}
``` |

From the discussion about service binding, it can be concluded that each global service is required a web service and a service interface in order to construct the dynamic proxy class. The detail explanation regarding web service will be discussed in next section.

5.4.7 System Web Services Implementation

This section discusses on the implementation of web services at the respective participated system domain. Hence, the web service implementation realizes the architecture design of CM’s components like Security, Invocation, Addressing, Messaging, and Routing (Section 5.3.1.1), while Service Interface of the service provider (Section 5.3.1.3) is also implemented.
In order to share a service to the public within the SCAIM integration domain, the service provider is required to implement a web service that belonging to the global service being shared. In .NET terms, service interface usually implemented by using an .asmx Web Service file and the code behind construct its logic. However in this research, service interface for *Outstanding PO* global service is implemented by using two class libraries named *IRequestService.cs* and *wsOutstandingPO.asmx* to fulfill the requirements of the service oriented architecture.

As continue from discussion in previous section, the result of constructed dynamic proxy class was stored into object *IReqService* of *IRequestService.cs*. The object is responsible to ensure the interoperability among the service request and the web method format. Table 5.16 indicates the source codes of the service interface object. For simulation purposes, the output of *DynamicWebService* method is fixed to a data set while input is a list of arguments objects.

**Table 5.16: Source Codes for Service Interface Object**

```csharp
public interface IRequestService
{
    #region Methods
    DataSet DynamicWebService(List<clsArgument> parameter);
    #endregion
}
```

File named *wsOutstandingPO.asmx* is created to implement rest of the components like security, messaging, routing, and invocation. Table 5.17 illustrates code snippets of how the web service of Outstanding PO global service invokes its respective business component. According to the codes, each of the arguments passed through the service interface will be converted into an array list. Then, a public method of business
component named **ChooseMethod** is called and the array list is passed into business component through this method.

**Table 5.17: Source Codes of Web Service for Outstanding PO Global Service**

```csharp
[WebService(Namespace = "http://tempuri.org/")]  
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class wsOutstandingPO : System.Web.Services.WebService
{
    public wsOutstandingPO() { }

    [WebMethod]
    public DataSet DynamicWebService(List<clsArgument> args)
    {
        DataSet dsResult = new DataSet();
        ArrayList arrList = new ArrayList();
        //Call Business Component
        BcERPOutstandingPO GRNDetails = new BcERPOutstandingPO();
        //Convert args to array list
        foreach (clsArgument argIn in args)
            arrList.Add(argIn.Value.ToString);
        dsResult = GRNDetails.ChooseMethod(arrList);
        return dsResult;
    }
}
```

**5.4.7.1 Web Services Enhancement 3.0**

Web Services Specifications (WS-specification) technology is chosen to implement the rest of the CM components like Addressing, Messaging, Routing, and Invocation. According to Ferguson, Storey, Lovering, and Shewchuk (2003), WS-specification is designed based on XML, SOAP, and WSDL extensibility model to provide interoperable a set of standard protocols for security, reliable messaging, and transactions in loosely coupled systems. The features are best suite with SOA based SCAIM architecture where the shared services within the integration domain are become secured, reliable and transacted.
To build WS-specification in SCAIM .NET projects, Web Service Enhancement (WSE) 3.0 software development toolkit is applied. WSE is basically serves as a processing engine for applying WS-specification to SOAP messages, more secure, easily and quickly. The first step to apply WSE in web service projects is to add a reference named `Microsoft.Web.Service3` to every system web services project. Once the reference is added, web configuration class and WSE SOAP extension class are added into web configuration file of web service, as shown in Table 5.18.

**Table 5.18: Registered Configuration and SOAP Extension Classes for WSE**

```
<configSections>
    <section name="microsoft.web.services3"
        Version=3.0.0.0, Culture=neutral,
        PublicKeyToken=31bf3856ad364e35" />
</configSections>

<soapExtensionImporterTypes>
    <add type="Microsoft.Web.Services3.Description.WseExtensionImporter,
        Microsoft.Web.Services3, Version=3.0.0.0,
        Culture=neutral,
        PublicKeyToken=31bf3856ad364e35" />
</soapExtensionImporterTypes>

<soapServerProtocolFactory
    type="Microsoft.Web.Services3.WseProtocolFactory,
    Microsoft.Web.Services3, Version=3.0.0.0,
    Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
```

Then, the dynamic web service proxy class that constructed during the discussion in Section 5.4.6 is registered with Web Services Configuration class (`Microsoft.Web.WebService3.WebServicesClientProtocol`). After the steps are completed, WSE filters are applied to all processing SOAP messages. The filters are dedicated to specific WS-specifications such as WS-Security, WS-Policy, and WS-Addressing.
5.4.7.2 Security Sub-Component

The Security sub-component of CM module is implemented by applying WS-Security of WS specifications. As compared to the traditional security mechanisms in service oriented architecture, WS-Security is more flexible, yet more costly in terms of performance and complexity (Bhargavan, Fournet, & Gordon, 2008). It provides a secure environment to service provider and consumer in SCAIM domain to exchange SOAP messages. In this research, several security standards like SSL, X.509 and Username token will be the focuses of WS-Security Implementation.

An X.509 token is used to digitally sign and encrypt SOAP messages based on public-key cryptography. X.509 test certificates provided by WSE toolkit is used for the prototyping of this research. In a normal service sharing process, an outgoing message is signed digitally by the consumer using the certificate. After that, service provider uses a public key to encrypt the message received from the consumer. The consumer is then uses a private key to decrypt message responded by the provider. To implement X.509 in Windows environment, the public key and private key are imported into certificate stores called Local Computer and Current User, respectively. Figure 5.27 depicts the Current User and Local Computer certificate stores in Microsoft Management Console (MMC).
WSE provides a set of strategies to easily secure web service, called Turnkey Security Assertion (TSA). Among the strategies, *UsernameForCertificationSecurity* of TSA is selected for the implementation of SCAIM security features. Basically, *UsernameForCertificationSecurity* is used to secure most of the common service sharing process at message level by using X.509 certificate. It also uses a username and password to identify service consumer, while the consumer uses a public certificate to encrypt message before SOAP message is sent via HTTP protocol.

WSE security policy is used to implement *UsernameForCertificationSecurity* assertion, and applying digital signatures, and encryption using an XML configuration file. It is a group of policy strategic that allows service consumer and provider to specify any custom policy requirements. The policy file will be separated from the code in order to increase the maintainability of the system. Table 5.19 shows the policy file applied to SCAIM web service named *UserNamePolicy* that generated using WSE Security Settings Wizard.
Meanwhile, a client security policy that complies with the rules defined in the web service policy (UserNamePolicy) is defined for the service consumer (ASC module). Figure 5.28 depicts the summary of security policy defined for ASC module.
5.4.7.3 Invocation Sub-Component

Invocation sub-component of CM is considered as another module related to the web service implementation. Implementation of invocation basically focuses on two types of communication pattern, which are synchronous and asynchronous web service calls. Both synchronous and asynchronous communication pattern have pros and cons when implementing service oriented architecture. The decision to make choice of the patterns is highly depends on the business requirements of the legacy systems.

Discussions in previous section (Web Service Implementation) have already focused on synchronous communication, which means service provider sends back a response message immediately after receive a service request. Due to the reason, this section focuses the implementation of service invoke asynchronously.
According to Harris (2006), asynchronous communication means data is sending without synchronization to an external clock, and it support any number of parallel tasks by using multi-threaded functions. Hence, it does not block any other process when a request is being processed by or sent to the provider.

Implementation of asynchronous processing is more complicated than traditional synchronous send and request communications. For simulation of SCAIM, the implementation involves creating polling mechanism and call back methods to the service consumer (ASC of SCM) as implemented during Section 5.4.2 earlier. Table 5.20 depicts on how **DynamicWebService** method in Table 5.17 will be invoked asynchronously.

<table>
<thead>
<tr>
<th>Table 5.20: Code Snippets for Asynchronous Calling Dynamic Proxy Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protected Sub btnInvoke_Click(ByVal sender As Object, ByVal e As System.EventArgs)</strong> Handles btnInvoke.Click</td>
</tr>
<tr>
<td>Dim IReqService As IRequestService = dynWSProxy.Build()</td>
</tr>
<tr>
<td>Dim AsynResult As IAsyncResult</td>
</tr>
<tr>
<td>Dim AsynCallback As New AsyncCallback(AddressOf fcCallBack)</td>
</tr>
<tr>
<td>AsynResult = dynWSProxy.BeginDynamicWebService(args, AsynCallback, Nothing)</td>
</tr>
<tr>
<td>End Sub</td>
</tr>
<tr>
<td><strong>Private Sub fcCallBack(ByVal AsynResult As System.IAsyncResult)</strong></td>
</tr>
<tr>
<td>Dim ds As DataSet = dynWSProxy.EndDynamicWebService(AsynResult)</td>
</tr>
<tr>
<td>End Sub</td>
</tr>
</tbody>
</table>

In order to starts asynchronous programming, an object type of **IAsyncResult** is created to poll for the completion of the method call, and then stores the final result. In this case, the result of the method call is a dataset listing object. Besides, two core asynchronous methods include a Begin (**BeginDynamicWebService**) and an End (**EndDynamicWebService**) method. The Begin method is responsible to kick off the asynchronous process while End method gather the result and return back to the service consumer. Next, **AsyncCallback** delegate object is created to serve as the pointer to a
callback function named \textit{fcCallback}. The callback function is used to determine the status of the asynchronous method call and to retrieve the result from the method call, as similar to previous synchronous pattern. The dynamic proxy class listed in Table 5.21 gives better illustration on how asynchronous web service is being invoked.

\begin{table}[h]
\centering
\caption{Partial Codes for Asynchronous Service Proxy Class}
\begin{verbatim}
public partial class wsOutstandingPO :
    SCAIM_ASC_DynamicWSLibrary.IRequestService {
     
    public System.IAsyncResult BeginDynamicWebService(System.Collections.Generic.List<clsArgument> > args, System AsyncCallback callback, object asyncState) {
        return this.BeginInvoke("DynamicWebService", new object[]
            { args}, callback, asyncState);
    }

    public System.Data.DataSet EndDynamicWebService(System.IAsyncResult asyncResult) {
        object[] results = this.EndInvoke(asyncResult);
        return ((System.Data.DataSet)(results[0]));
    }
}
\end{verbatim}
\end{table}

5.4.7.4 Addressing Sub-Component

The Addressing sub-component of CM module is implemented through WS-Addressing of WS-specifications technology. It enables the SOAP messages to store and manage own addressing information of source, destination, and URI location of a requested service that across multiple endpoints. Hence, it is impossible to construct asynchronous web services because messages only able to travel through traditional HTTP connection without applying WS-Addressing.
To achieve the functionalities, two supported constructs of WS-Addressing - message information headers (MIH) and endpoint references (EPR) are applied to WSDL of the service. MIH standardize information pertaining to message processing that related to replies, action, faults and the relationship to the prior messages (Fox, Pallickara, & Parastatidis, 2004). It works in a similar to email message addresses, which define full set of addressing information. Table 5.22 shows the SOAP message of Outstanding PO service incorporated with MIH.

**Table 5.22: Service Message with Message Information header**

```xml
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
    <soap12:Header>
        <wsa:MessageID>uuid:d728se11-1231-4d4d-1947-7382d192ele3</wsa:MessageID>
        <wsa:To>http://ERP_Server/wsOutstandingPO</wsa:To>
        <wsa:Action>wsOutstandingPO#DynamicWebService</wsa:Action>
    </soap12:Header>
    <soap12:Body>
        <DynamicWebService xmlns="http://tempuri.org/">
            <args>
                <clsArgument>
                    <Value>string</Value>
                </clsArgument>
                <clsArgument>
                    <Value>string</Value>
                </clsArgument>
                <clsArgument>
                    <Value>string</Value>
                </clsArgument>
            </args>
        </DynamicWebService>
    </soap12:Body>
</soap12:Envelope>
```

Meanwhile, EPR refer to standardization of reference format for a web service and its instance. This has includes the supported operations at a given port and how SOAP message is appointed to the port. EPR for the service is shown in Table 5.23. The EPR consists of references information for message destination such as **AccountID** and **PortType** that are separated in difference XML tags. This is used to represent SOAP binding information to the specific endpoint of the message destination.
In SCAIM architecture, Addressing sub-component provides specification to implement messaging sub-component. The context of the addressing will be implemented through the next sub-component - Messaging.

### 5.4.7.5 Messaging Sub-Component

The component is used to transport the SOAP message during service requests and responses. In many cases, SOAP messaging would be based on traditional request/response semantics or they could be based on asynchronous one-way messaging (Fox, Pallickara, & Parastatidis, 2004). Unlike HTTP, TCP protocol supports asynchronous messaging to reduce scalability problems of sender for waiting responses of sender.

This section describes on the implementation of TCP transport protocol into SCAIM integration system for transporting and processing SOAP messages across different application boundaries. Table 5.24 and 5.25 indicates the code snippets of SOAP sender and receiver through TCP protocol.
The implementation is using WSE dedicated classes such as “SoapReceiver” and “SoapSender” to construct the service consumer and provider via TCP protocol. By using the SCAIM Addressing sub-component, the SoapSender class sends out a request message to a specific pre-defined endpoint (URI) of the service provider.
Table 5.25: The TCP Service Provider Code Snippets

```vbnet
Public Class ServiceRequestReceiver
    Inherits SoapReceiver
    Protected Overrides Sub Receive(ByVal message As SoapEnvelope)
        Try
                Dim request As RequestService = DirectCast(message.GetBodyObject(GetType(RequestService)), RequestService)
                Dim response As New SoapEnvelope()
                Dim responseService As New ResponseService

                Dim strSystemCode_LocalService As String = request.args(0).Value.ToString()
                Dim strServiceAddress As String = FKeyField("GlobalService GS LEFT JOIN ServicesMap SM ON SM.GlobalService = GS.GlobalService AND SM.SystemCode = GS.Provider LEFT JOIN LocalService LS ON LS.LocalService = SM.LocalService AND LS.SystemCode = SM.SystemCode", "ServiceAddress", LS.SystemCode + LS.LocalService = "" & strSystemCode_LocalService & "' ")
                Dim dynWSProxy As clsDynamicWSProxy(Of IRequestService) = New clsDynamicWSProxy(Of IRequestService)(strServiceAddress)
                dynWSProxy.UseGenericList = True
                Dim IReqService As IRequestService = dynWSProxy.Build()

                responseService.ds = IReqService.DynamicWebService(request.args)
                response.SetBodyObject(responseService)
                Dim toUri As Uri = message.Context.Addressing.ReplyTo
                Dim fromUri As Uri = message.Context.Addressing.[To]

                response.Context.Addressing.From = New From(fromUri)
                Dim soapSender As New SoapSender(toUri)

                soapSender.Send(response)
            End If
            Catch ex As Exception
                End Try
            End Catch
        End Sub
    End Class
End Class
```
When the request message is received by the service endpoint, the first step for the `SoapReceiver` class is to de-serialize the soap message body by using "getBodyObject" class. Secondly, the receiver processes the serialized message body and stored the body into correct data type (`args`). Then, the values of `args` are used as parameters to invoke web service of SCAIM shared service, in this case, the `DynamicWebService` method. Upon receiving of the result from the web service (a dataset), the `SoapReceiver` class sends the response message back to the service requester.

### 5.4.7.6 Routing Sub-Component

In normal cases, a SOAP message that sent out by service requester or provider follows their message header setting. No intermediaries are allowed to be visited by the message during the routes to the destination (Yu, Liu, Bouguettaya, & Medjahed, 2008). However, the Routing sub-component enables flexible redirecting of the message to alternative destination, which is not accessible by the message sender.

In SCAIM architecture, the Routing component is focuses on the Load Balancing design model, where a message is routed from a service endpoint to destination endpoints that located in multiple clustered service providers. The load balancing routing scales the performance of the service provider server and it maintains the high availability of the services. The implementation code is demonstrated by a configuration file named referral cache, which is shown in Table 5.26.
The referral cache is used to specify where the service request should be routed to. In the example when SCM request for an ERP’s service (Outstanding PO List), the service requester can invoke the service via the message router instead of request the Outstanding PO web service directly as normal service invoke. There are two elements to be set in the xml of referral cache - `<r:for>` and `<r:via>`. The element `<r:for>` specify the incoming URL where element `<r:via>` specify the actual address of the service.

5.4.8 Business Components Implementation

As mentioned in 5.4.1.3, all business logics of each particular participants of SCAIM are placed in their business component respectively instead of service interface. For example, a participated ERP system has implemented a business component named “SCAIM_ERP_BC” in the same application server with ERP System. The structure of the solution is shown in Figure 5.29.
Whenever a service of ERP in “SCAIM_ERP_WS” web service project is being invoked by SCM service requester, the web service will call a method of the business component named “ChooseMethod” as discussed early in Table 5.24. Based on the arguments passed in by the requester, a specific function “BcGRNDetailsBySupplierGLN” is called. This function is responsible to return a dataset value that is retrieved from “GetResultDataSet” method of Data Access component to the service interface. Code snippets of “ChooseMethod” and “BcGRNDetailsBySupplierGLN” method are listed in Table 5.27.
Table 5.27: Code Snippets of ERP Business Component

```csharp
public DataSet ChooseMethod(ArrayList arrList)
{
    switch(arrList[1].ToString())
    {
    case "GetOutstandingPOBySupplierGLN":
        return BcGRNDetailsBySupplierGLN(arrList);

    case "GetOutstandingPOByPONo":
        return BcGRNDetailsByPONo(arrList);

    default:
        return null;
    }
}

public DataSet BcGRNDetailsBySupplierGLN(ArrayList arrList)
{
    DataSet dsResult = new DataSet();
    DataTable dtTableField = new DataTable();
    string strFieldsSQL = "";
    string strTableSQL = "";
    try
    {
        dtTableField = new DataTable();
        if ((dtTableField != null))
        {
            if (dtTableField.Rows.Count >= 1)
            {
                for (int i = 0; i < dtTableField.Rows.Count; i++)
                {
                    strFieldsSQL += dtTableField.Rows[i]["LocalFieldSQL"].ToString() + ", ";
                }
            }
        }
        if (strFieldsSQL.Trim() != "")
        strTableSQL = GetTableSQL(strFieldsSQL, arrList[0].ToString(), arrList);

        //Call Data Access Component
        dsResult = GRNDetails RESULTbageSet(strTableSQL);

        return dsResult;
    }
    catch
    { throw;  }
}
```
5.4.9 Data Access Implementation

As refer to Section 5.3.1.3, Data Access (DA) component is responsible to implement the command from business component to access the backend data store of the service provider system. To CRUD (Create, Read, Update and Delete) the data of, an OLEDB class library name “clsOledbDatabase” has been developed to streamline the database operations such as database connection creation and closing, data records, execute database command query, invoke stored procedures, views, functions, open a data reader, data view, dataset and so on. Table 5.28 indicates on how the DA component uses one of the methods of class library named “OpenDataSet” to retrieve result of service into a dataset.

Table 5.28: Data Access Component to Access ERP Data

```csharp
public class DaERPGRNDetails: DaBaseClass
{
    public DataSet GetResultDataSet(string strSQL)
    {
        clsOledbDatabase objOledbDB = new clsOledbDatabase(ConfigurationSettings.AppSettings["ConnectionString"]);
        DataSet ds = new DataSet();
        try
        {
            objOledbDB.OpenDataSet(ref ds, strSQL);
            return ds;
        }
        catch
        {
            throw;
        }
        return null;
    }
}
```

5.4.10 Integration Output Implementation

EDI and CSV are only two output format for final output of the integration between integration participants. The output file is stored in the file server before being downloaded by the service requester. Thus, the Transformation component of ASC responsible to convert the dataset received from CM into predefined output format. The
predefined format of a shared service is done through a UI of CM (Maintain Global Service) before the service is available to be shared to other participants. Figure 5.30 depicts the screen for maintain a service’s output format.

![Figure 5.30: UI of CM to Maintain Service Integration Output](image)

For demonstration purposes, two classes have been created to support the converting process of the result dataset to CSV and EDIFACT format, named `CSV.cs` and `EDIFACT_RECADV.cs`. Table 5.29 shows the partial source codes for CSV converting method.
First part of the method converts each column header of the result data into the first line of the CSV. Second part indicates the way of converting the data line by line into CSV cells.

UN/EDIFACT has been chosen as the system output EDI format. For demonstration purposes, Receiving Advice Message (RECADV) message type of UN/EDIFACT has been chosen for the shared service example (Outstanding PO) which is so far discussed. The full description of REVADV can be found at the Message Type Directory EDMD (2009). Hence, the complete code snippets of `EDIFACT_RECADV.cs` class are stated in Appendix E. Finally, the output EDI based on the shared service is depicted in Figure 5.31.
5.5 System Testing

System testing is aimed to ensure the entire program or information system, of which the modified program was a part, still works (Whitten, Bentley, & Dittman, 2002). It is used to evaluating design and attributes of the system compliance with its required results. It is more difficult challenges for the testing of service orientation system, especially when it comes to testing the interaction between heterogeneous, loosely coupled and independent developed services (Bartolini, Bertolino, Elbaum, & Marchetti, 2009). Generally, system testing is falls within the scope of unit testing, module testing, integration testing and system testing. In this research, black box and white box testing are emphasized during the unit testing stage. Hence, non-functional requirements of SCAIM as discussed earlier in Section 5.2.2 are also tested with its required results.
5.5.1. Unit Testing

Unit testing is the first testing procedure. Unit testing is done after the completion of each function according to functional requirements. The unit testing process is similar as to find faults in components. There were several steps being carried out for this research:

- Examine all the program codes by reading through it. Consecutively, try to spot algorithm, data and syntax faults.
- Compare the code with the specifications and with design to make sure all relevant cases have been considered.
- Develop black-box oriented test cases to show the input is properly converted to the desired output.
- Boundary conditions are tested to make sure the functions run at boundaries established for limiting or restricting processing.
- Test all errors handling paths.
- Examining the program logics based on white-box method

Unit testing follow exactly these steps, and one specific unit is examined with them one at a time.

5.5.1.1 Program Coding Correction

For the purposes of unit testing, a program is correct if it implements the function and data properly as indicated in the design, and if it’s interface properly with other components. One way to investigate program correctness is to view the code as a statement of logical flow.
The SCAIM integration system maintainer uses this technique for the unit testing of adding new global concepts, which is one of the core functionalities required for the configuration before a service is available for sharing. Based on the discussion in Section 5.3.2 (Functional Design), a flow diagram was drawn to demonstrate the logic of various conditions after a global concept is added, edited or deleted. For instance, if a global concept is added successfully, the page is redirected to the global concept listing page and the new concept is shown. Yet, an error message is prompted if there are errors occurred during the concept maintenance.

5.5.1.2 Debugging Strategy

In the progress of developing the DEQPMA system, the following debugging strategies are applied:

- **Using the Debugger Provided by .NET framework**

  The .NET framework includes a visual debugger that enables us to step line-by-line through the statements of an executing ASP.NET page. The debugger was used to create breakpoints and watches, to view the values of variables at any point during page execution.

- **Using Query Analyzer Provider by Database IDE**

  When the database transaction error occurred, the error message will be displayed and most of the time, it is caused by incorrect syntax in the SQL statement. Thus, SQL statement being used to execute the transaction will be tested by using Query Analyzer. Query Analyzer provides more information about the error message. Moreover, Query Analyzer is used to correct the SQL statement when wrong information is being retrieved although the SQL statements being used are correct.
- **Algorithm Review**

  A database error or logic error could be occurred even though the program is running well, due to the information is not consistent as intended. Reviewing algorithm and computations for their correctness and efficiency is needed for this purpose. The efficiency of the program is different when different algorithms are used.

5.5.1.3 Black Box Testing

In SCAIM integration system, no limitation for the number of global services can be added into CM module for sharing purposes. Each of the global service is compliance with a respective web service, and the service provide syntactical interface for general black-box tests. As according to Whalen, Rajan, Heimdahl, and Miller (2006), black-box testing is a software testing technique that do not care about the internal structure of implementation while derive test case. This means it only focuses on valid and invalid inputs being selected and the produced outputs as compare to the expected test results.

![Figure 5.32: Input for Black-Box Testing of a Shared Service](image)

Figure 5.32 depicts the test input for a supplier to invoke a service called “Outstanding PO”. Supplier’s identifier and vendor’s PO number are two input parameters are required as according to the configuration in CM module. The output result of the
service is in EDIFACT format, a well known business. The expected correct result (list of PO Details) is shown in Figure 5.33. In other words, the correct result indicated the test is successfully and vice-versa.

![Figure 5.33: Correct Output of Black-Box Testing](image)

5.5.1.4 White Box Testing

Another method used for SCAIM unit testing is white-box testing. Recently, the testing method has become popular for SOA applications testing (Bartolini et al., 2009; Bertolino, 2009; Tillmann & Halleux, 2008). The main concern of it is to ensure that the logical part of the source codes to construct a single program is workable and produce expected correct results. To use white-box testing, the knowledge of execute functionalities to be tested is required. The implementation of white-box testing against SCAIM unit testing is illustrated in Table 5.30. This table represents partial .NET code snippets to invoke an external service as shown in the interface of Figure 5.32. For the instance, two input parameters are required - Supplier Identifier and Vendor’s PO (from and to value respectively). By setting any possible input values (decimal points, characters, negative value etc.), the expected outputs are analyzed.
### Table 5.30 (a): White-Box Testing on Button Click Events

```vbnet
Protected Sub btnInvoke_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnInvoke.Click
    Try
        Dim ds As DataSet = NormalInvoke(args)
        ExportToFile(ds)
    Catch ex As Exception
        Response.Write("Invalid service invoked, Error:" & ex.Message)
    End Try
End Sub
```

### Table 5.30 (b): White-Box Testing on Invoking Dynamic Web Service

```vbnet
Private Function NormalInvoke(ByVal args As Object) As DataSet
    Try
        Dim IReqService As IRequestService = dynWSProxy.Build()
        Return IReqService.DynamicWebService(args)
    Catch ex As Exception
        Return Nothing
    End Try
End Function
```

### Table 5.30 (c): White-Box Testing on Calling the Business Component

```csharp
[WebMethod]
public DataSet DynamicWebService(List<clsArgument> args) //1
{
    try
    {
        DataSet dsResult = new DataSet();
        BcERPOutstandingPO GRNDetails = new BcERPOutstandingPO();

        //Call Business Component
        dsResult = GRNDetails.BcGRNDetailsBySupplierGLN(arrList);
        return dsResult;
    }
    catch
    {
        throw;
    }
}
```
Table 5.30 (d): White-Box Testing on Data Access Component

```csharp
public DataSet BcGRNDetailsBySupplierGLN(ArrayList arrList)
{
    try
    {
        //Call Data Access Component
        dsResult = GRNDetails.GetResultDataSet(strTableSQL);
        return dsResult;
    } catch { throw; }
}
```

Table 5.30 (e): White-Box Testing on Auto-Generated SQL Strings

```sql
SELECT  GRN.GRNNo, GRN.ETADate, GRN.PONO, GRN.PODate, GRN.TotalAmount,
        GRN.SupplierGLN, GRN.SupplierName, GRN.ShipToGLN,
        GRNDetail.LineNo, GRNDetail.GTIN, GRNDetail.ItemNo,
        GRNDetail.OrderQty, GRNDetail.ReceiveQty, GRNDetail.UOM,
        GRNDetail.ActualDeliveredDate
FROM GRNDetail INNER JOIN GRN ON GRNDetail.GRNNo = GRN.GRNNo
WHERE  GRN.SupplierGLN >= '9557363915003'
AND GRN.SupplierGLN <= '9557363915003'
AND GRN.PONO >= '123456'
AND GRN.PONO <= '999999'
ORDER BY  GRN.GRNNo, GRN.ETADate, GRN.PONO, GRN.PODate,
        GRN.TotalAmount, GRN.SupplierGLN, GRN.SupplierName,
        GRN.ShipToGLN, GRNDetail.LineNo, GRNDetail.GTIN,
        GRNDetail.ItemNo, GRNDetail.OrderQty, GRNDetail.ReceiveQty,
        GRNDetail.UOM, GRNDetail.ActualDeliveredDate
```

Table 5.30 (f): White-Box Testing on Retrieving Result Data Set

```csharp
public DataSet GetResultDataSet(string strSQL)
{
    DataSet ds = new DataSet();
    try
    {
        objOledbDB.OpenDataSet(ref ds, strSQL);
        return ds;
    } catch { throw; return null; }
}
5.5.2 Module Testing

Module testing is a necessary testing phase to ensure that all tested units could be work together as a single system module. Also known as component testing, it concerns with the testing of all related components for which a separate specification exists. Figure 5.34 depicts SCAIM participated system Module, which consists of four main units – User Interface, Service Interface, Business Component, and Data Access Component, as discussed earlier in Section 5.3.1.3.

Figure 5.34 (a): Units in SCAIM Participated System Module

Figure 5.34 (b): Module Testing on User Interface
//Dynamic Generated Service Interface
Dim IReqService As IRequestService = dynWSProxy.Build()

//Business Component
dsResult = GRNDetails.BcGRNDetailsBySupplierGLN(arrList);

//Data Access Component
dsResult = GRNDetails.GetResultDataSet(strTableSQL);

Figure 5.34 (c): Module Testing on Service Interface, Business and Data Component

5.5.3 Integration Testing

After satisfied that individual components are working correctly. These components will be combined into a working system. This will be the beginning of integration testing. The integration process is planned and coordinated properly with the intention that the faults that occur during this stage do not rely within the unit of the system. This will be easier for the task to detect the cause of the faults. The emphasis of integration testing is on testing database between modules. Hence, the top down integration approach has been adopted for this research. The top level-controlling component is tested by itself.

Table 5.31: Integration Testing and Evaluation

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Tested</th>
<th>Evaluated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of navigation</td>
<td>√</td>
<td>Good</td>
</tr>
<tr>
<td>Linkage</td>
<td>√</td>
<td>Good</td>
</tr>
<tr>
<td>Information shared</td>
<td>√</td>
<td>Good</td>
</tr>
</tbody>
</table>

5.5.4 System Testing

System testing is very different from unit testing and integration testing. In unit testing, developer has complete control over the testing process, including prepared test data, and test cases. However, in system testing, the customer also test the system, making sure that it meets their understanding of the requirement, which may be different from the developers. Different from other testing objective, the purpose of system testing is to
ensure that the system does what the stakeholders (company staffs, customer, supplier, and integration maintainer) want it to do.

System testing begins with functional testing which is based on the functional requirements. A function test checks that the integrated system performs its function as specified in the system analysis. In SCAIM, functional testing will be carried out on five main sub-systems, which are CM, ASC, ERP, CRM, and SCM as discussed in 5.2.1 (Functional Requirements).

5.5.5 Non-Functional Requirement Testing

Security Testing

The SCAIM integration system is tested against unauthorized access to storage, communication and information during the activities of service sharing. The testing is done by implementing encryption method such as SSL over the intranet and internet. As mentioned in Section 5.4.1 (CM Implementation), security features like Windows Authentication Mode is tested and able to restrict only users within a network domain able to invoke shared services. Thus, the Security component of CM is also tested to ensure the protection of whole SCAIM system. It includes the features like system login, SQL injection protection (security vulnerability against the database level security), and password encryption using MD5 algorithm (to protect password cracking). Finally, WS-Security of WS-Specifications that using SSL, X.509 and Username token standard is tested to ensure the security of SOAP messages while transferring between service provider and consumer.
Performance Testing

Performance testing compares the integrated components with the non-functional system requirements. It determines how efficiency for a system to response under a particular tasks. For SCAIM, the performance testing would be focus on the system attributes such as system processor usage (%), RAM usage (%), and SQL Duration (milliseconds). During the testing, a SCM user is invokes a shared service named “Outstanding PO” registered in CM, which is a function located in ERP system. Thus, the testing begins with 10 PO details records, following by 20, 50, 100, 200, 500, 1000, 2000, 5000 records according to the UI selection criteria. The result of the performance test is shown in Table 5.32, Figure 5.35, and Figure 5.36. All alternative system programs are terminated to increase the accuracy of the testing.

Table 5.32: Performance Testing Results

<table>
<thead>
<tr>
<th>Number of PO Details Record in EDI output</th>
<th>Memory Usage (%)</th>
<th>CPU Usage (%)</th>
<th>SQL Duration (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.0</td>
<td>3.0</td>
<td>151</td>
</tr>
<tr>
<td>20</td>
<td>5.0</td>
<td>5.0</td>
<td>169</td>
</tr>
<tr>
<td>50</td>
<td>7.0</td>
<td>6.0</td>
<td>173</td>
</tr>
<tr>
<td>100</td>
<td>8.0</td>
<td>8.0</td>
<td>277</td>
</tr>
<tr>
<td>200</td>
<td>14.0</td>
<td>10.0</td>
<td>385</td>
</tr>
<tr>
<td>500</td>
<td>23.0</td>
<td>16.0</td>
<td>401</td>
</tr>
<tr>
<td>1000</td>
<td>30.0</td>
<td>29.0</td>
<td>440</td>
</tr>
<tr>
<td>2000</td>
<td>40.0</td>
<td>38.0</td>
<td>604</td>
</tr>
<tr>
<td>5000</td>
<td>59.0</td>
<td>57.0</td>
<td>884</td>
</tr>
</tbody>
</table>

Figure 5.35: System Performance on Bar Chart View
SQL Profiler is used to accumulate the database duration as gathered in the above chart. The interface of SQL Profiler is configurable to customize the performance parameters, as shown in Figure 5.37. To analyze the system processor and memory usage, Window Server Performance Utility Tool is used as depicted in Figure 5.38.

Figure 5.36: Database Performance on Bar Chart View

Figure 5.37: SQL Response Time Testing Using SQL Profiler
The testing results showing that system is perform in optimal level regardless the increasing of the total output records.

5.6 System User Evaluation

In order to gather the opinions and assessment from users regarding to the research system, a system evaluation form is designed for the purposes (refer Appendix F). The designed evaluation form consists of 5 Yes/No questions and a scaled question (rate from 1 to 5). During two weeks evaluation period in June 2010, 10 working personnel from different companies were participated in the evaluation process. On top of this, the respondents played different job specification in their respective company. Three respondents work as IT developers, two as system maintainers, two as business analysts, an IS manager, a CIO, and an accountant.

Before seeking the responses, the system is configured and hosted in an intranet allowing the respondents to access at any time during the evaluation period. User guidance manual is distributed to the respondents by email. During the evaluation
process, phone interview was conducted to gather the responses due to time constraints. All the assessments from respondents during the interview were recorded word for word.

Responses to each particular question are analyzed. For the first question, 80 percents of the respondents agreed SCAIM have the capability to interpret data from one to another system. The motivation behind the selection is because the system able to reorganize the data in EDI output, which represent common meaning understandable by the source system user. Besides, 7 out of 10 respondents agreed the system is able to share out any of the system functionality to other system users. Particularly, 4 of them mentioned “Outstanding PO” provided by ERP as their desired shared function to the supplier.

70 percents of responses agreed that the system is capable of encouraging and improving the collaboration between the supplier and customer. 80 percents of them agreed SCAIM supports open standard of information exchange, which is readable by all system users. Hence, most of them provide the good assessment on the interface of the system. The reason is because it provide scalable interface tool for conceptual mapping and it is configurable to handle any system changes.

In conjunction with the system design, the last question focuses on non-functional requirements of SCAIM as discussed earlier. In short, most respondents rate over average of satisfaction on its security, quality of service, usability, performance, and expandability.
5.7 Chapter Summary

The functional and non-functional requirements of SCAIM development are identified. Thus, the architecture, functionality, component, database, and interface design of SCAIM are explained in details. Besides that, the chapter is also discussed on how integration success could be achieved through conceptual and service mapping by SCAIM methodology. After the development requirements of SCAIM integration system are defined, system implementation is conducted through the development of ten sub-projects which represents each tier of SCAIM components. The methodology is tested through unit, module, integration and system testing. Thus, system user has evaluated the capability of SCAIM in terms of data or service interpretation between participated systems, collaboration between business partners, system ease of use, and non-functionalities.
CHAPTER 6: CONCLUSION AND IMPLICATION

6.1 Introduction

As outlined in Chapter 1, the main objective of this research was to design and simulate service based methodology that enables interoperability among three conceptually equivalence enterprise system – ERP, CRM and SCM. This dissertation has fulfilled this objective by outlining the design, development and simulation of the SCAIM integration system with two core components – CM and ASC.

In this chapter, the summary of the dissertation consists of the problems encountered during the research stages, the contribution of the research system to the social, limitation of SCAIM, and its future enhancement.

6.2 Problems Encountered

This section describes several difficulties faced during the whole development life cycle of SCAIM integration system. Hence, solution is proposed for each of the problem encountered.

- **Difficulty in Determining the Scope of the System**

  To build a full-scale complete integration system within the time given frame, it is impossible. The scope of ERP, CRM and SCM system integration using SCAIM could be very wide. The main reason is because there are too many modules under the definition of the systems, and defining which modules have similar application domain is difficult. Inexperience with the current integration model available was another hindrance to implement true integration system methodology. Thus, the inability to distinguish which of the integration system features are essential, either desired or optional, has made defining the project scope a complicated task.
To outline the scope of the system planning, the research system is focuses on integration of a single sub-system of each participated system, as discussed in Chapter 1. Besides, some popular integration methodologies of enterprise systems such as MOM, ETL, and ESB has chosen as references in designing the main scope of the research system - CM and ASC modules.

- **Lack of Experience in Domain Knowledge**

Lacking of knowledge in the business processes that involving the use of ERP, SCM and CRM system, has caused the design and development process beyond estimated research schedule. In order to cater the issue, some resources like newsletters, forums, and books are consulted. Thus, the conducted survey had given opinions from respondents who are direct or indirect involved in the process of the research domain.

- **Difficulty in Generating Dynamic Proxy of Shared Service**

It is easy to find a solution to implement a shared service by using static web services technology, but there are less resource regarding the dynamic generating a proxy binding to a service whenever a service is requested. Microsoft .NET framework 2.0 (the development technology) do not include library of auto-generating WSDL of a web service dynamically. However, the solution of this issue is to discuss with some programmers and post question to online forums. Fortunately, the responses are adequate for the candidate to develop that combine several .NET libraries to include the required features into SCAIM system.
6.3 System Strength and Contribution

The main objective of this research is to create a functional service-oriented methodology for ERP, CRM and SCM to share its own services (functionalities) to each other within an application domain. The design and implementation of the research integration system, SCAIM, provides the following contributions:

- **Enable Semantic Interoperability Among Participants**
  The SCAIM design and implementation enable the conceptual level service sharing for independent enterprise applications, especially for ERP, CRM and SCM systems (which had been proven similar from representation and operational perspective as discussed in chapter 2). The capability of the methodology to translate the local concepts (attributes) to global concepts accurately has overcome the problem of misinterpretation of semantic meaning among the systems.

- **Improve Functionality Interoperability within an Application Domain**
  According to user feedback, any functionality of any system within a defined domain is able to be shared out to other another through the integration system. The implementation of service mapping in chapter 5 has proven the capability of service being invoked dynamically based on user’s configuration. By using the dynamic web service method, POS of ERP system is able to retrieve customer transactions through the service shared out by SCM, even though they are not designed to work together initially.

- **Improve Integration of ERP with Customer and Supplier**
  Functionality design of SCAIM discussed in chapter 5 clarify that the methodology enables supplier to review their list of unsettled purchase orders through a service called “OutstandingPO” shared by ERP. Thus, customer can view their transaction status through another service provided by ERP in CRM
system. These scenarios have proven the capability of SCAIM in integrating ERP with SCM and CRM system.

- **Open Standard Integration Methodology**
  SCAIM integration system is implemented by using some independent technologies and approach such as Web Services, EDIFACT business data format, and EAI laws and principles. Web services are web technology that mainly used for realizing service oriented architecture of SCAIM. The sub-components of CM like Routing, Messaging, Addressing, Invocation, and Security are constructed using Web Services technology. Since Web Services are based on open standard like XML, SOAP, WSDL, and HTTP, the components compatible with any operating system, programming language and hardware. Hence, Web Service and SCAIM encourage more secure, reliable and transacted integration domain.

- **Integrated Tool for System Change**
  The activities of service sharing are in a manner that no modification is required to the CM and ASC modules. It is because the assemblies of a service are generated during the runtime when a service is being invoked. With no modification is needed for SCAIM itself, the methodology create a flexible integration environment that increase the scalability and maintainability of the system.

- **Conceptual Model Implementation**
  The research has outlined the construction of CM which located at the central of the architecture of SCAIM. CM facilitates the service-sharing process among the participants by providing Service Mapping features as discussed in Chapter 5. The features allows the service requester to invoke the resources (mainly the functions) provided by any other participated systems. Thus, the implementation
of CM provides a demonstration on how private local services of provider are mapped to public global services within an integration domain for service-sharing purposes. As discussed and demonstrated in chapter 5, any component of participants (ERP, CRM and SCM) such as SDS of ERP, CSS of CRM, and EPS of SCM are registered into storage of CM and provide capability to share out their services to each others. Due to the prototype which involved SDS, CSS and EPS are core system components of participants systems, the concept of CM also applicable to variety of industries such as Manufacturing, Government, Education, Banking, Health Care, Retails, and so on.

- **Adaptive Service Connector Demonstration**

ASC is another important contribution to the enterprise systems integration efforts. The core component of SCAIM enables Conceptual (or attribute) Mapping features. The local private attributes (concepts) are mapped to global public concepts in order to ensure any exchanged information between systems has the same meaning. The construction of ASC provides valuable insight into the effort for semantic interoperability among the participated systems within an application domain. Meanwhile, the prototype of ASC depicts several common business processes such as checking outstanding PO, request for quotation, and customer pre-orders are able to be invoked by any other participated system in the integration domain. Through the conceptual mapping capability of ASC, these common business processes or service sharing could be widely applicable to different kind of industries and market.

The architecture of SCAIM is fully support for open business standard output data format such as UN/EDIFACT. The output of SCAIM integration system is configurable through CM and ASC module to accept any message type of the EDI. Thus, the open
standard data format is acceptable by Worldwide and it can fulfill the business needs of participated system like ERP, CRM and SCM.

The strengths indicate SCAIM is a reliable, scalable, secured, better performance, user friendly and robustness integration methodology. It provides guidelines to software maintainers, managers, system administrators, and developers (who are forced to work independently with each system) to construct an integrated and conceptual equivalence applications environment.

6.4 Limitations and Weaknesses

Due to time constraints and complexity of service sharing concept for isolated systems, SCAIM does not reflect all sub-systems of ERP, CRM and SCM. Most of the time this research is focuses on the issues of interoperability among the SDS of ERP, CSS of CRM, and EPS of SCM. In other words, the current version of SCAIM is support for 3 participants. In other words, there is no practically implementation testing on how complexity the SCAIM concepts and services mapping could be handled, even though in theoretically it is possible to unlimited participants.

The simulation of SCAIM during the development stage of this research is implemented within an intranet. All services and concepts are configured and stored in data store within CM module. There are no public UDDI is registered to the public network in order to share the services worldwide. Hence, the integration system does not provide facilities for real World to lookup for the shared services due to technology constraints of the research environment.
It would be bit complex for the configuration of UN/EDIFACT output data format. The integration maintainer or system end user is required to have the knowledge of EDIFACT message type being applied to a particular shared service. Hence, the user interface of ASC is limited to configure up to second level of the EDIFACT data format (header and trailer). Due to research time constraints, no manual is provided for the basic knowledge of EDIFACT business documents, as it could be found from online resources.

6.5 Future Enhancement

The success implementation and simulation of research system has proves the conceptual level integration between the integration participants using the service oriented methodology. However, there are much efforts required to improve and further enhance SCAIM integration system.

Firstly, it is important for the integration methodology to support conceptual and service mapping between unlimited modules of ERP, CRM and SCM system. The similarities among each of the modules of the enterprise systems are focused, especially from representation and operational perspectives. Hence, other enterprise applications such as Financial Management, Human Capital, Product Lifecycle, Field Service Management, and Enterprise Asset Management system are potentially could be included in the next version of SCAIM. The main objectives to include other applications into integration is to create consolidate all business resources within a global competitive environment, or usually called Extended Enterprise concept.
Secondly, the performance of the integration system has to be improved. The current SCAIM is designed to share out a service by using single thread. It would be interesting to investigate the possibility of serving the incoming service requests in a pool of multi-threading. It could increase the performance of the integration system to handle a huge number of requests at the same time. Besides, the enhancement should include features to allow service consumer to access offline data locally which saved in the previous service consume. It might be using technology such as OLAP to achieve the purposes. Thus, the integration system also could be included with direct application-to-application route between regular service consumer and provider particularly. In this manner, it is suggested to include customize configuration facilities into CM module in order to fulfill the requirements.

Another area to be addressed is the security of service sharing activities. The research system had applied WS-Security technology for service-level security and user level authorization for ASC and CM access. However, it is necessary to have more security measures to prevent unauthorized invoked of the services. The suggesting security enhancement includes applying policy-based SOAP message security, private and public keys encryption for each request, service client certificate authentication, role-based authorization and so on.

6.6 Conclusion

SCAIM is a methodology or simulation system for facilitating service-sharing among ERP, CRM and SCM systems. These systems are proved to be conceptual equivalence, regardless from the perspective of representative and operational. Throughout the research survey, design and implementation, it is believed that the original goals as outlined in chapter 1 are achieved. By sharing the functionalities among systems using
open standard and .NET technologies, goals like Information Integration, Process Integration and Vendor Independent are achieved.

Earlier systems are lack of integration among each other, which caused business processes to be done manually. With the advent of SCAIM, most of the interoperability problems at conceptual level could be overcome. For example, the research system uses conceptual and service mapping to realize the semantic and functional interoperability among systems, which are pre-requisites to integration success. Conceptual mapping is a method to ensure exchange information between two participated systems has the same meaning. Service mapping realize the service interoperability among system being integrated.

Systems maintaining and enhancement are continuous and important jobs in software development life cycle. The successful simulation of integration among ERP, CRM and SCM using the service-oriented methodology can be served as guidelines for system stakeholders to obtain integrated and accuracy output information. Hence, the methodology implementation details provided in this report could be referred to as an advance research for service-oriented integration technology.
REFERENCES


Testing, analysis, and verification of web services and applications, Seattle, Washington.


APPENDIX

Appendix A: Survey Questionnaire

Title: Integrating ERP, SCM and CRM using Service Oriented Methodology

Introduction
The purpose of this survey is to gather information about current level of system integration and the role of the middleware software play within organizations throughout Peninsular Malaysia and Singapore. This survey was made up of 35 questions of varying formats, with intention of gathering data on the following:

- Typical type of computerized systems used to manage customers, suppliers, and the company business processes
- Current level of system integration in the companies
- Perception of the company about the importance of the system integration
- Opinions regarding use of Service Oriented Methodology on integrating ERP, SCM and CRM systems
- Requirements and information about the feasibility of the promote idea of the research

Definition

- **System Integration** provides the means of linking together components of different computer systems functionally to achieve certain business processes
- **Service Oriented Methodology** is defined as IT infrastructure which has the ability to make different applications functionality available to all authorized users
- **Enterprise Resource Planning** (ERP), **Supply Chain Management** (SCM) and **Customer Relationship Management** (CRM) are among the enterprise applications that have similar application domain and have the most significant market demands on their integration for collaboration between company with its customers and suppliers

Instruction
Please answer Section A to F in this survey.
Target Respondents
This survey is mainly targeting on respondents who work in the company that are implementing more than one computerized systems. If your organization does not implemented any computerized systems, please answer the questions based on your personal experiences.

Notes
The survey will take less than 20 minutes to complete. This survey is particularly for the purposes of my master study research at University of Malaya and your responses are confidential. Your individual responses, including any contact information you provide, are confidential and will not be shared with any other companies, departments, colleges, or universities. No information from this survey will be sold and your particulars will not appear in any publication of results.

Section A: Personal Particulars
Please ☐ your answer where applicable. (Click on the check box to select)

Name: ________________________________________________________________
Gender:  □  Male       □   Female
Job Title / Role:  □   Executive Manager (CEO / COO / Director / LOB)
                  □   CIO / CTO
                  □   IT / IS Manager
                  □   IT Architect /Developer
                  □   System Maintainer
                  □   IT Consultant
                  □   Business Manager
                  □   Business Analyst
                  □   Other (Please specify) _________________________________

Company Name: _______________________________________________________
Company URL: _________________________________________________________
Contact No.: (Office) _____________________ (Mobile) _______________________
Email Address: _________________________________________________________
Section B: General Questions

Please ☑ your answer where applicable.

1. How many employees in your organization?
   ☐ Small - Less than 50 employees
   ☐ Medium - 50 to 500 employees
   ☐ Large - More than 500 employees

2. What is your company’s primary business area?
   ☐ Education
   ☐ Services
   ☐ Government
   ☐ Wholesale
   ☐ Health Care
   ☐ Manufacturing
   ☐ Finance
   ☐ Non-Profit
   ☐ Retail
   ☐ Construction
   ☐ Other (Please specify) ______________________________________________

3. Does your company have an accounting based computerized system that mainly use for identifying and planning resources within the organization?
   ☐ Yes  ☐ No

4. Does your company purchase product or services from suppliers via Internet or any computerized system?
   ☐ Yes  ☐ No

5. Is there any existing information system for your customers get to know or purchase your company’s products or services?
   ☐ Yes  ☐ No
   If Yes, please list the information systems used:
   (a) ________________________________
   (b) ________________________________
   (c) ________________________________
6. How does your company staffs request for quotation (RFQ) or product information from the suppliers?

☐ Manual (Paperwork)
☐ Via Email / Telephone
☐ Computerized System
☐ Other (Please specify) ______________________________________________

7. How does your company get to know the up-to-dated products or services delivery status from your customers?

☐ Manual (Paperwork)
☐ Via Email / Telephone
☐ Computerized System
☐ Other (Please specify) ______________________________________________

8. Would you prefer to have a computerized system that enables direct collaboration between the customers and the suppliers? (For instance to let your company’s suppliers to receive customer forecast for regular products or services directly from customers)

☐ Yes      ☐ No

If No, why? __________________________________________________________
If Yes, why? __________________________________________________________

Section C: Your Company’s Level of System Integration

Please ☐ your answer where applicable.

1. What is the current enterprise systems used at your company? (Check all that apply)

☐ Customer Relationship Management (CRM)
☐ Enterprise Asset Management (EAM)
☐ Enterprise Resource Planning (ERP)
☐ Financial Management System (FMS)
☐ Human Capital Management (HCM)
☐ Performance Management (PM)
☐ Product Lifecycle Management (PLM)
☐ Supply Chain Management (SCM)
☐ Other (Please specify) ______________________________________________
2. Who uses the systems? (Check all that apply)

☐ Executive Manager  ☐ CIO / CTO
☐ IT / IS Manager  ☐ Business Manager
☐ Business Analyst  ☐ Administrative Staff
☐ Engineer  ☐ Normal Staff
☐ Other (Please specify) ________________________________

3. What do they use the system for?
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________

4. When was the first enterprise system implemented in your company?
Year: _______

5. When was the most current system implemented in your company?
Year: _______

6. Were the systems developed by the same vendor?
☐ Yes  ☐ No
If No, why? ________________________________
If Yes, why? ________________________________

7. Is the existing enterprise systems share information among each of the system for business purposes?
☐ Yes  ☐ No

8. Does your company currently maintain different separate systems that are not initially designed to be work together?
☐ Yes  ☐ No

9. Does your company facing any problem to integrate different systems component in order to fulfill specific business processes?
☐ Yes  ☐ No
10. How integrated would you rate your company’s various enterprise systems in terms of business processes as being right now?

- [ ] Totally Integrated
- [ ] Very Integrated
- [ ] Somewhat Integrated
- [ ] No Integration

11. Does your company currently utilize any middleware suites or programs to integrate current enterprise applications to achieve certain business purposes?

- [ ] Yes (continue question 12)
- [ ] No (go to question 14)

12. Which vendor of the middleware suites or programs?

- [ ] SAP
- [ ] Sun Microsystems
- [ ] Microsoft
- [ ] IBM
- [ ] Oracle
- [ ] Sage Group
- [ ] Other (Please specify) ______________________________________________

13. Are you satisfied with the current level of integration achieved by applying the middleware suites or programs?

- [ ] Yes
- [ ] No

If No, why? ___________________________________________________________
_____________________________________________________________________
_____________________________________________________________________

14. Does your company use Electronic Data Interchange (EDI) to sharing data with other company?

- [ ] Yes
- [ ] No
- [ ] Don’t know

**Section D: Your Company’s Perception of System Integration**

Please ☐ your answer where applicable.

1. In your opinion, how important is to combine different systems functionality as a whole to your company business processes?

- [ ] Not important
- [ ] Moderately
- [ ] Very Important
- [ ] Essential
2. Which statements below best describe your company’s current position in terms of system integration?

☐ Your company has no intention to combine different systems functionality
☐ Your company has plans to integrate different systems functionality
☐ Your company is implementing system integration to achieve business purposes
☐ Your company already has an integrated system that able to fulfill current business needs

3. Which of the following method do you think is more cost effective and less time consuming to combine different systems functionality?

☐ Purchase or develop middleware suites or programs
☐ Develop a brand new system to integrate different systems information

4. Based on your current and anticipated business requirements, how important are the following benefits of integrated systems to your company? Please rate using scales 1 to 5, where:- (Please circle the number)

1 represents “not important at all”
2 represents “not important”
3 represents “average”
4 represents “important”
5 represents “very important”

a. Improve the information exchange between customers, suppliers and your company
   1 2 3 4 5
b. Increase the competitive advantage of the company
   1 2 3 4 5
c. Increase the capability of supply chain collaboration
   1 2 3 4 5
d. Centralize business policies and rules among each of the existing systems
   1 2 3 4 5
e. Increase customer satisfaction
   1 2 3 4 5
f. Improve communication with suppliers
   1 2 3 4 5
5. Are the following limitations or barriers influence the implementation of system integration in your company? (Check all that apply)

☐ Lack of integration expert
☐ Existing systems are constantly changing
☐ Complexity of legacy systems
☐ Users from one department refuse to share their data with other department
☐ No consensuses on the final outcome of interface design of the integration
☐ Problem on mapping the data between different systems
☐ Conflict requirements of different departments
☐ High integration cost and time consuming
☐ Other, please explain briefly. __________________________________________

6. Which of the following systems below do you think that have the capability to increase the collaboration between customers, suppliers and your company? (Check all that apply)

☐ Customer Relationship Management (CRM)
☐ Enterprise Asset Management (EAM)
☐ Enterprise Resource Planning (ERP)
☐ Financial Management System (FMS)
☐ Human Capital Management (HCM)
☐ Performance Management (PM)
☐ Product Lifecycle Management (PLM)
☐ Supply Chain Management (SCM)
☐ Other (Please specify) __________________________________________

Section E: Promotion of ERP, SCM and CRM Systems

Integration using Service Oriented Methodology

A software platform with integration capability will be developed in particularly to integrate ERP, SCM and CRM subsystems function together as a system. The software consists of two main components, which are called Conceptual Model (CM) and Adaptive Service Connector (ASC). CM provides interface for accessing available functions as a result of integration while different ASCs will be installed into the different participate systems respectively for communication purposes.
1. Do you think ERP, SCM and CRM are most suitable group of enterprise systems to be integrated functionally in order to achieve business purposes?
   □ Yes  □ No
   If No, please suggest other systems. _________________________________________
   _______________________________________________________________________

2. Do you think setting up the service oriented middleware software as mentioned above will be a good idea to integrate different applications as a whole to automate business processes?
   □ Yes  □ No
   If No, why? ____________________________________________________________
   _______________________________________________________________________

3. What are the main features would you like to have in the middleware software as mentioned?
   □ Create, modify or delete functionalities which are not available for existing systems
   □ Security feature to authorize specific users to access particular functionalities
   □ Mapping data from different systems that have the same or similar meaning
   □ Accumulate specific data from different systems and generate one single report listing
   □ Customize the report listing columns based on different requirements
   □ Standardized business terms or naming understandable by participated ERP, CRM and SCM subsystems
   □ Guideline to integrate functionalities from different systems
   □ Other (Please specify) _______________________________________________
   _______________________________________________________________________
Section F: Other Opinions

Please ☑ your answer where applicable.

1. Is the system integration beneficial to your company?
   - Yes  ☐ No  ☑

   If Yes, please state how. _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________

   If No, please state why. _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________

2. What are the limitations and weaknesses of the current computerize systems in your company?

   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________

3. What are the problems that your company is facing while integrating different systems to achieve particular purposes?

   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________

4. Others: -

   _________________________________________________
   _________________________________________________
   _________________________________________________
   _________________________________________________
## Appendix B: SCAIM Use Case Tables

### Table B.1: Maintain System List Use Case

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Maintain System Participant List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Integration Maintainer</td>
</tr>
<tr>
<td>Description</td>
<td>This use case mainly used for list, add, edit and delete system that to be participated in the integration system model.</td>
</tr>
<tr>
<td>Pre Condition</td>
<td>Integration Maintainer has logged on the system and has authorized to access the option.</td>
</tr>
<tr>
<td>Basic Course</td>
<td></td>
</tr>
<tr>
<td>Step 1:</td>
<td>Integration Maintainer selects “Maintain Systems” at side navigator menu of the CM main page.</td>
</tr>
<tr>
<td>Step 2:</td>
<td>CM displays the listing of entered system participants of the integration model.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Integration Maintainer select “New” button at listing page to redirect screen to add new system participant page.</td>
</tr>
<tr>
<td>Step 4:</td>
<td>Integration Maintainer enters new system participant details and submits it by clicking “Save” button.</td>
</tr>
<tr>
<td>Step 5:</td>
<td>CM save the information to Service Dictionary (Database) and redirect back to system listing page.</td>
</tr>
<tr>
<td>Step 6:</td>
<td>Integration Maintainer edits a particular system participant.</td>
</tr>
<tr>
<td>Step 7:</td>
<td>Integration Maintainer modifies the system information and submits updated information by clicking “Save” button.</td>
</tr>
<tr>
<td>Step 8:</td>
<td>CM save the updated information to Service Dictionary and redirect back to system listing screen.</td>
</tr>
<tr>
<td>Step 9:</td>
<td>Integration Maintainer select system participant to be deleted from the integration model.</td>
</tr>
<tr>
<td>Step 10:</td>
<td>Integration Maintainer confirms the deletion.</td>
</tr>
<tr>
<td>Step 11:</td>
<td>CM removes the deleted system records from the database.</td>
</tr>
<tr>
<td>Post Condition (Succeed)</td>
<td>System participant records are added, updated, or deleted successfully based on user selection.</td>
</tr>
<tr>
<td>Post Condition (Failed)</td>
<td>The system participant records are not added, updated, or deleted based on user selection. CM prompting user error messages.</td>
</tr>
</tbody>
</table>
# Table B.2: Maintain Local Concepts Use Case

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Maintain Local Concepts List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Integration Maintainer</td>
</tr>
<tr>
<td>Description</td>
<td>This use case is mainly used for list, add, edit, and delete local concepts provided by a particular participated system.</td>
</tr>
<tr>
<td>Pre Condition</td>
<td>Integration Maintainer has logged on to the Adaptive Service Connector of particular participated system (for example ERP system), and has authorized to access the option.</td>
</tr>
<tr>
<td>Basic Course</td>
<td></td>
</tr>
<tr>
<td>Step 1:</td>
<td>Integration Maintainer selects “Local Concepts” at side navigator menu of the ASC main page.</td>
</tr>
<tr>
<td>Step 2:</td>
<td>ASC displays the listing of existing local concepts.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Integration Maintainer select “New” button at listing page to redirect screen to Add New Local Concept page.</td>
</tr>
<tr>
<td>Step 4:</td>
<td>Integration Maintainer enters new local concept details and submits it by clicking “Save” button.</td>
</tr>
<tr>
<td>Step 5:</td>
<td>ASC save the information to Service Dictionary (Database) and redirect back to local concepts listing page.</td>
</tr>
<tr>
<td>Step 6:</td>
<td>Integration Maintainer edits a particular local concept.</td>
</tr>
<tr>
<td>Step 7:</td>
<td>Integration Maintainer modifies the local concept information and submits updated information by clicking “Save” button.</td>
</tr>
<tr>
<td>Step 8:</td>
<td>ASC saves the updated information to Service Dictionary and redirect back to concepts listing screen.</td>
</tr>
<tr>
<td>Step 9:</td>
<td>Integration Maintainer selects a local concept to be deleted from the integration model.</td>
</tr>
<tr>
<td>Step 10:</td>
<td>ASC displays a dialog window prompting the Integration Maintainer to confirm the deletion.</td>
</tr>
<tr>
<td>Step 11:</td>
<td>Integration Maintainer confirms the deletion.</td>
</tr>
<tr>
<td>Step 12:</td>
<td>ASC removes the deleted concept records from the database.</td>
</tr>
<tr>
<td>Post Condition (Succeed)</td>
<td>The local concept records are added, updated, or deleted successfully based on user selection.</td>
</tr>
<tr>
<td>Post Condition (Failed)</td>
<td>The local concept records are not added, updated, or deleted based on user selection. ASC prompting user error messages.</td>
</tr>
</tbody>
</table>
### Table B.3: Map Concepts Use Case

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Concepts Mapping Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Integration Maintainer</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This use case is mainly used for mapping a local concept with its respective global concept.</td>
</tr>
<tr>
<td><strong>Pre Condition</strong></td>
<td>Particular local and global concepts to be mapped have been entered into ASC and CM respectively.</td>
</tr>
<tr>
<td><strong>Basic Course</strong></td>
<td></td>
</tr>
<tr>
<td>Step 1:</td>
<td>Integration Maintainer selects “Map Concepts” at side navigator menu of the ASC main page.</td>
</tr>
<tr>
<td>Step 2:</td>
<td>ASC displays selection for range of local concepts to be listed.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Integration Maintainer selects from and to local concepts to be listed for mapping, and then click “List” button to query the concepts.</td>
</tr>
<tr>
<td>Step 4:</td>
<td>ASC displays the listing of existing local concepts alongside with its respective mapped global concepts.</td>
</tr>
<tr>
<td>Step 5:</td>
<td>Integration Maintainer selects a particular local service to be mapped or re-mapped.</td>
</tr>
<tr>
<td>Step 6:</td>
<td>ASC stays at the same page, but display the particular concept at editable mode.</td>
</tr>
<tr>
<td>Step 7:</td>
<td>Integration Maintainer select a global concept as refer to the editing local concept, and then click “update” link to map the concepts.</td>
</tr>
<tr>
<td>Step 8:</td>
<td>ASC saves the updated mapping to Service Dictionary and set the editing local concept row to normal mode.</td>
</tr>
<tr>
<td><strong>Post Condition (Succeed)</strong></td>
<td>The selected local concept is mapped to the selected global concept successfully.</td>
</tr>
<tr>
<td><strong>Post Condition (Failed)</strong></td>
<td>The selected local concept is mapped to the selected global concept unsuccessfully. ASC prompting user error messages.</td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Check Outstanding Purchase Order</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Actor</td>
<td>Supplier - SCM System User</td>
</tr>
<tr>
<td>Description</td>
<td>This use case describes the process of a supplier checking outstanding PO or Goods Receive Notes (GRN) details of products. The process indicates how the supplier (User of SCM System) accesses the details through e-Procurement System (EPS) of SCM system that request the service provided by a Sales and Distribution System (SDS) of ERP system.</td>
</tr>
<tr>
<td>Pre Condition</td>
<td>Supplier has logged on to the EPS of SCM system, and has authorized to access the option.</td>
</tr>
</tbody>
</table>

### Basic Course

**Step 1:** Supplier selects “External Services” at side navigator menu of the SCM main page.

**Step 2:** SCM system displays Invoke External Service main page, together with selection of global service to be invoked.

**Step 3:** Supplier selects to invoke Outstanding PO global service.

**Step 4:** SCM systems display details of the selected Outstanding PO global service.

**Step 5:** Supplier click on the “Invoke” button to proceed.

**Step 6:** SCM redirected to Outstanding PO web service page.

**Step 7:** Supplier selects a particular Supplier GLN for output results.

**Step 8:** SCM displays supplier name of the selected Supplier GLN.

**Step 9:** Supplier click on “Invoke” to proceed checking outstanding PO.

**Step 10:** SCM gather service details from Service Dictionary and generate output EDI that consists of outstanding PO records, then prompt supplier to download the EDI file from server.

**Step 11:** Supplier click “Save” button on the prompting window to save the EDI to local computer.

### Post Condition (Succeed)
The Outstanding PO service is invoked and EDI output file download successfully.

### Post Condition (Failed)
The Outstanding PO service is not invoked and EDI output file download unsuccessfully. ASC prompting user error messages.
<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>List Product Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Staff – ERP System User</td>
</tr>
<tr>
<td>Description</td>
<td>This use case describes the process of a company staff collects for customer satisfaction level regarding the product or service provided to them. The process indicates how the company staff (user of ERP System) accesses the details through Sales and Distribution System (SDS) of ERP system that request the service provided by a Customer Service and Support (CSS) of CRM system.</td>
</tr>
<tr>
<td>Pre Condition</td>
<td>Company staff has logged on to the SDS of ERP system, and has authorized to access the option.</td>
</tr>
<tr>
<td>Basic Course</td>
<td></td>
</tr>
<tr>
<td>Step 1</td>
<td>Staff selects “External Services” at side navigator menu of the ERP main page.</td>
</tr>
<tr>
<td>Step 2</td>
<td>ERP system displays Invoke External Service main page, together with selection of global service to be invoked.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Staff selects to invoke Product Satisfaction global service.</td>
</tr>
<tr>
<td>Step 4</td>
<td>ERP systems display details of the selected Product Satisfaction global service.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Staff clicks on the “Invoke” button to proceed.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Staff selects a particular Supplier Code for output results.</td>
</tr>
<tr>
<td>Step 7</td>
<td>ERP displays Supplier Name of the selected Supplier Code.</td>
</tr>
<tr>
<td>Step 8</td>
<td>Staff clicks on “Invoke” to precede viewing product satisfaction level.</td>
</tr>
<tr>
<td>Step 9</td>
<td>ERP gather service details from Service Dictionary and generate output EDI that consists of delivery status records, then prompt staff to download the EDI file from server.</td>
</tr>
<tr>
<td>Step 10</td>
<td>Staff clicks “Save” button on the prompting window to save the EDI to local computer.</td>
</tr>
<tr>
<td>Post Condition (Succeed)</td>
<td>The Product Satisfaction service is invoked and EDI output file download successfully.</td>
</tr>
<tr>
<td>Post Condition (Failed)</td>
<td>The Product Satisfaction service is not invoked and EDI output file download unsuccessfully. ASC prompting user error messages.</td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Request for Quotation</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Actor</td>
<td>Staff – ERP System User</td>
</tr>
<tr>
<td>Description</td>
<td>This use case describes the process of a company staff requests prices, information and quotations for a particular product from suppliers. The process indicates how the company staff (user of ERP System) accesses the details through Sales and Distribution System (SDS) of ERP system that request the service provided by a E-Procurement System (EPS) of SCM system.</td>
</tr>
<tr>
<td>Pre Condition</td>
<td>Company staff has logged on to the SDS of ERP system, and has authorized to access the option.</td>
</tr>
</tbody>
</table>
| Basic Course  | **Step 1:** Staff selects “External Services” at side navigator menu of the ERP main page.  
**Step 2:** ERP system displays Invoke External Service main page, together with selection of global service to be invoked.  
**Step 3:** Staff selects to invoke Product Quotation global service.  
**Step 4:** ERP systems displays details of the selected Product Quotation global service, such as provider, web method name and list of input and output concepts of the service.  
**Step 5:** Staff clicks on the “Invoke” button to proceed.  
**Step 6:** Staff selects a particular Product ID for output results.  
**Step 7:** ERP displays Product Name of the selected Product ID.  
**Step 8:** Staff clicks on “Invoke” to precede viewing product information and prices.  
**Step 9:** ERP gather service details from Service Dictionary and generate output EDI that consists of delivery status records, then prompt staff to download the EDI file from server.  
**Step 10:** Staff clicks “Save” button on the prompting window to save the EDI to local computer. |
| Post Condition (Succeed) | The Product Quotation service is invoked and EDI output file download successfully. |
| Post Condition (Failed) | The Product Quotation service is not invoked and EDI output file download unsuccessfully. ASC prompting user error messages. |
# Appendix C: SCAIM Data Dictionary

## Table C.1: SCAIM Tables and Descriptions

<table>
<thead>
<tr>
<th>Database</th>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service Dictionary</strong></td>
<td>User</td>
<td>SCAIM user login information</td>
</tr>
<tr>
<td></td>
<td>System</td>
<td>Participated systems information</td>
</tr>
<tr>
<td></td>
<td>GlobalConcept</td>
<td>Global concepts available in whole SCAIM domain for semantic mapping purposes</td>
</tr>
<tr>
<td></td>
<td>GlobalService</td>
<td>Global services available in whole SCAIM domain for semantic mapping purposes</td>
</tr>
<tr>
<td></td>
<td>ServiceConcept</td>
<td>Services and concepts</td>
</tr>
<tr>
<td></td>
<td>LocalConcept</td>
<td>Concepts locally to the respective system</td>
</tr>
<tr>
<td></td>
<td>LocalService</td>
<td>Services locally to the respective system</td>
</tr>
<tr>
<td></td>
<td>ConceptsMap</td>
<td>Mapping information for local and global concepts</td>
</tr>
<tr>
<td></td>
<td>ServiceMap</td>
<td>Information related to global and local services mapping</td>
</tr>
<tr>
<td><strong>ERP System Database</strong></td>
<td>GRN</td>
<td>Goods Received Notes header information</td>
</tr>
<tr>
<td></td>
<td>Supplier</td>
<td>Company’s supplier information</td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>Company’s customer information</td>
</tr>
<tr>
<td></td>
<td>SalesOrder</td>
<td>Sales order header information</td>
</tr>
<tr>
<td></td>
<td>SalesOrderDetail</td>
<td>Sales order details information</td>
</tr>
<tr>
<td></td>
<td>Product</td>
<td>Company products and services</td>
</tr>
<tr>
<td><strong>CRM System Database</strong></td>
<td>Customer</td>
<td>Customer information used for CRM system</td>
</tr>
<tr>
<td></td>
<td>Product-Satisfaction</td>
<td>Satisfaction level information provided by customers through CRM system</td>
</tr>
<tr>
<td></td>
<td>CustomerProductSatisfaction</td>
<td>Customer and Product Satisfaction objects</td>
</tr>
<tr>
<td></td>
<td>PreOrder</td>
<td>Related to customer forecast information on products and services</td>
</tr>
<tr>
<td><strong>SCM System Database</strong></td>
<td>Supplier</td>
<td>Supplier information for SCM system uses</td>
</tr>
<tr>
<td></td>
<td>Quotation</td>
<td>Quotation information provided by multiple suppliers</td>
</tr>
<tr>
<td></td>
<td>QuotationProduct</td>
<td>Products or service respective to the quotations</td>
</tr>
<tr>
<td></td>
<td>NewProduct</td>
<td>New product information not yet release to market or not available for customers</td>
</tr>
</tbody>
</table>
Appendix D: SCAIM User Manual

Service-Oriented Conceptual Adaptive Integration Methodology (SCAIM)

User Configuration Manual

SCAIM is a service-oriented integration system to integrate ERP, CRM and SCM at conceptual level.

This manual provides instructions on integration system configuration, implementation, and user guidance. In particularly, the manual covers the guidance of CM and ASC settings and the user guide on invoke shared services.

Contents:

Section 1 – Introduction

Section 2 – System Requirements

Section 3 – CM Module Settings

Section 4 – ASC Module Settings

Section 5 – Invoke Shared Service
SECTION 1 – INTRODUCTION

SCAIM integration system consists of 2 core modules - Conceptual Model (CM), Adaptive Service Connector (ASC). All conceptual attributes and respective shared service available are maintained in CM module. ASC is mainly used to facilitate service sharing activities among participated systems. Of whole architecture of SCAIM, only a single CM is required for the operation of whole integration system, whilst the number of ASC to be implemented is same as the number of system participated, which is three for this demonstration.

SECTION 2 – SYSTEM REQUIREMENTS

This section lists out the necessary software and hardware for client side required to execute the SCAIM system interface.

Minimum hardware requirements for user environment are as follows:

Table D.1: Minimum Hardware Requirement of User Environment

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Minimum Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Pentium 4 and compatible processor or higher, minimum speed of 1.0GHz, 1.8GHz or higher is recommended</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>512 MB or more is recommended</td>
</tr>
<tr>
<td>Hard Disk Space</td>
<td>Minimum of 5GB</td>
</tr>
<tr>
<td>Other</td>
<td>Network Interface Card and other standard computer peripherals</td>
</tr>
</tbody>
</table>

Minimum software requirements for user environment are as follows:

Table D.2: Minimum Software Requirement of User Environment

<table>
<thead>
<tr>
<th>Software</th>
<th>Minimum Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Microsoft Window XP or later</td>
</tr>
<tr>
<td>Browser</td>
<td>Internet Explorer 6.0 SP1 or later, Firefox 3</td>
</tr>
</tbody>
</table>
SECTION 3 – CM MODULE SETTINGS

3.1 Login

Figure D.1: Login Screen of CM Module Interface

Steps:

1. Open an Internet Explorer browser, type in CM module interface URL, for example: http://www.thecompany.com/CM/SCAIM_CM/Login.aspx
2. Key in User’s ID and Password into respective column as shown in Figure D.1. For demon purposes, enter “IntegrationMaintainer001” for User ID and “password” for Password. After that, clicks on Login button to proceed. Upon login successfully, the screen will be redirect to the main page of CM settings – System List, as shown in Figure D.2.

Figure D.2 depicts the main screen after login into CM interface. CM settings are divided into 3 functions as shown in the right navigation menu:

1. System List - Interface for registration of any enterprise system modules to be participated in the system integration.
2. Global Concepts - Function of CM that allows integration maintainer to maintain concepts (attributes) of shared services globally. The concepts are standardized and recognized by all participated systems through this function.
3. Global Services - Interface for the configuration of shared services globally. The information of services registered through this function is recognized by other participated systems.
3.2 System List

Integration maintainer is allowed to add, edit and delete any system module from or into the integration domain of SCAIM.

3.2.1 Register/Add a New System Module

Steps:

1. Starting from screen as shown in Figure D.2, click on the “New System” button to redirect to the detail page of the registration, as depicted in Figure D.3.
2. Choose a “System Type” from the drop down list. It could be only ERP, CRM or SCM to be selected as type of the module.
3. Enter the name of the system module.
4. Enter a unique code for the system that is not existed in current list of the system.
5. Click on “Save” button to proceed.
3.2.2 Modify a Registered System Module

Figure D.4: Edit a Registered Participated System

Steps:
1. Starting from screen as shown in Figure D.2, click on “Edit” link button of a particular row of the system list to edit the registered system module. The screen is redirect to the detail page for editing, as depicted in Figure D.4.
2. Edit the name or select other system type of the module. No editing is allowed to the system code due to the unique criteria represented by the attribute.
3. Click on “Save” button to proceed.

3.2.3 Delete a Registered System Module

Figure D.5: Remove a Registered System Module

Steps:
1. From the system listing as shown in Figure D.2, ticks on check box of a particular row that represents the system module to be deleted.
2. Click on “Delete” button to remove selected system module.
3. A message box is prompted for confirmation of the removing. Click “OK” button to precede the deletion.
3.3 Global Concepts

![Figure D.6: Maintain Global Concepts through CM](image)

Integration maintainer is allowed to add, edit and delete any concepts (attribute of service) from or into the integration domain of SCAIM. Click on the “Global Concepts” from right navigation menu of the screen to view the list of registered concepts.

3.3.1 Register/Add a New Global Concept

![Figure D.7: Register a New Global Concept](image)

**Steps:**

1. Starting from screen as shown in Figure D.6, click on the “New Global Concept” button to redirect screen to the detail page of the registration, as depicted in Figure D.7.
2. Enter the name and description of the global concept.
3. Click on “Save” button to proceed.
3.3.2 Modify a Registered Global Concept

**Figure D.8: Edit a Registered Global Concept**

**Steps:**
1. Starting from screen as shown in Figure D.6, click on “Edit” link button of a particular row of the global concepts list to edit the registered concept. The screen is redirect to the detail page for editing, as depicted in Figure D.8.
2. Edit the description of the concept. No editing is allowed to the indicator of the concept due to the unique criteria represented by the attribute.
3. Click on “Save” button to proceed.

3.3.3 Delete Registered Global Concepts

**Figure D.9: Remove Registered Global Concepts**

**Steps:**
1. From the system listing as shown in Figure D.6, ticks on check boxes of particular rows that represents the global concepts to be deleted.
2. Click on “Delete” button to remove selected concepts.
3. A message box is prompted for confirmation of the removing. Click “OK” button to precede the deletion.
3.4 Global Services

Integration maintainer is allowed to add, edit, edit WSDL and delete any services from or into the integration domain of SCAIM. Click on the “Global Services” from right navigation menu of the screen to view the list of registered services.

3.4.1 Register/Add a New Global Service

Steps:
1. Starting from screen as shown in Figure D.10, click on the “New Global Service” button to redirect screen to the detail page of the registration, as depicted in Figure D.11.
2. Enter the name and description of the global concept.
3. Select Provider (system module) of the service being registered from a drop down list. The drop down list consists of system module as registered in section 3.2 earlier.
4. Choose the output data format as Output Type. The designed service output format could be in EDIFACT or CSV format.
5. If EDIFACT is selected, the maintainer is required to enter the Message Type of UN/EDIFACT.
6. Click on “Save” button to proceed.
3.4.2 Modify a Registered Global Service

Steps:
1. Starting from screen as shown in Figure D.10, click on “Edit” link button of a particular row of the global service list to edit the registered service. The screen is redirect to the detail page for editing, as depicted in Figure D.12.
2. Edit the description of the service. No editing is allowed to the indicator of the service due to the unique criteria represented by the attribute.
3. Modify selection of Provider, Output Type, and Message Type if necessary.
4. Click on “Save” button to proceed.

3.4.3 Modify WSDL of a Registered Global Service

Steps:
1. Starting from screen as shown in Figure D.10, click on “Edit” link button of a particular row of the global service list to edit the registered service. The screen is redirect to the detail page for editing, as depicted in Figure D.12.
2. Edit the description of the service. No editing is allowed to the indicator of the service due to the unique criteria represented by the attribute.
3. Modify selection of Provider, Output Type, and Message Type if necessary.
4. Click on “Save” button to proceed.
Steps:
1. Starting from screen as shown in Figure D.10, click on “Edit WSDL” link button of a particular row of the global service list to edit WSDL of the registered service. The screen is redirect to the detail page for editing, as depicted in Figure D.13.
2. Edit the Service Address of the global service and its web method name.
3. Select the location where WSDL file is located by click on the “Browse…” button and select the location.
4. If necessary, click on “Open & Edit WSDL File” button to open the WSDL file from the location as stated in step 3.
5. Click on “Save” button to proceed.

3.4.4 Delete Registered Global Services

![Figure D.14: Remove Registered Global Services](image)

Steps:
1. From the system listing as shown in Figure D.10, ticks on check boxes of particular rows that represents the global services to be deleted.
2. Click on “Delete” button to remove selected services.
3. A message box is prompted for confirmation of the removing. Click “OK” button to precede the deletion.
SECTION 4 – ASC MODULE SETTINGS

As described earlier, each participating system is required an ASC module within SCAIM architecture.

4.1 Login

![Adaptive Service Connector](image)

Figure D.15: Login Screen of ASC Module Interface

**Steps:**

1. Open an Internet Explorer browser, type in ASC module interface URL, for example: [http://ERP_Server/SCAIM_ASC/Login.aspx](http://ERP_Server/SCAIM_ASC/Login.aspx)
2. Key in User’s ID and Password into respective column as shown in Figure D.15. For demon purposes, enter “IntegrationMaintainer001” for User ID and “password” for Password. After that, clicks on Login button to proceed. Upon login successfully, the screen will be redirect to the main page of ASC settings - Local Concepts List, as shown in Figure D.16.

Figure D.16 depicts the main screen after login into ASC interface. ASC settings are divided into 4 functions as shown in the right navigation menu:

1. **Local Concepts** - Function of ASC that allows integration maintainer to maintain concepts (attributes) of shared services locally. The local concepts registered here are available to be included as output or input attributes of any local service.
2. **Local Services** - Interface for the configuration of services locally. For example, ERP has a function called “GRN Details List”. The function has to be registered into SCAIM through this function before it can be shared out.
3. **Map Concepts** - Function to facilitate the conceptual mapping between local and global concepts. One global concept could be mapped to multiple local concepts.
4. **Map Service** - Interface for local and global service mapping purposes.
4.2 Local Concepts

The maintainer of ERP is allowed to add, edit and delete any local concept from or into the ASC module.

4.2.1 Register/Add a New Local Concept

Figure D.17: Register a New Local System Concept
Steps:
1. Starting from screen as shown in Figure D.16, click on the “New Local Concept” button to redirect to the detail page of the registration, as depicted in Figure D.17.
2. Enter criteria such as Local Concept, Data Type, Length of the field, description and Dropdown SQL Select Statement. The entered SQL statement will constructs the list in the interface of service invoked for this particular concept.
3. Click on “Save” button to proceed.

4.2.2 Modify a Registered Local Concept

Steps:
1. Starting from screen as shown in Figure 16, click on “Edit” link button of a particular row of the concepts list to edit the registered local concept. The screen is redirect to the detail page for editing, as depicted in Figure D.18.
2. Modify criteria such as Data Type, Length of the field, description and Dropdown SQL Select Statement. No editing is allowed to the local concept indicator due to the unique criteria represented by the attribute.
3. Click on “Save” button to proceed.
4.2.3 Delete a Registered Local Concept

![Figure D.19: Remove a Registered Local Concept](image1.png)

**Steps:**

1. From the local concept listing as shown in Figure D.19, ticks on check box of a particular row that represents the local concept to be deleted.
2. Click on “Delete” button to remove selected local concept.
3. A message box is prompted for confirmation of the removing. Click “OK” button to precede the deletion.

4.3 Local Services

![Figure D.20: Maintain Local Service through ASC of ERP System](image2.png)
Integration maintainer is allowed to add, edit and delete any services represented by the particular system that using the ASC module. For example, “GRN Details List” is one of the existing functionalities of ERP system. In order to share out this service, maintenance of the local service is required. Besides, registered local concepts as maintained in previous screens are able to be added as input or output attributes of a particular service through this function.

Click on the “Local Services” from right navigation menu of the screen to view the list of registered local services of ERP system.

### 4.3.1 Register/Add a New Local Service to ASC of ERP System

![Figure D.21: Register a New Local Service to ASC of ERP System](image)

**Steps:**

1. Starting from screen as shown in Figure D.20, click on the “New Local Service” button to redirect screen to the detail page of the registration, as depicted in Figure D.21.
2. Enter the name and description of the local service.
3. If the service data selection involves joining multiple tables from database, enter SQL statement in the column of “Local Table SQL Statement”.
4. Click on “Save” button to proceed.

### 4.3.2 Modify a Registered Local Service

![Figure D.22: Edit a Registered Local Service](image)
Steps:

1. Starting from screen as shown in Figure D.20, click on “Edit” link button of a particular row of the Local Service list to edit the registered service. The screen is redirect to the detail page for editing, as depicted in Figure D.22.
2. Edit description and SQL statement of the concept if necessary. No editing is allowed to the indicator of the service due to the unique criteria represented by the attribute.
3. Click on “Save” button to proceed.

4.3.3 Delete Registered Local Services

![Figure D.23: Remove Registered Local Services](image)

Steps:

1. From the system listing as shown in Figure D.20, ticks on check boxes of particular rows that represents the local services to be deleted.
2. Click on “Delete” button to remove selected service.
3. A message box is prompted for confirmation of the removing. Click “OK” button to precede the deletion.
4.3.4 Maintain Input / Output Concepts of the Local Services

Figure D.24: Maintain Input and Output Concepts of a Particular Local Service

Steps:

1. As refer to Figure D.24, user can add or delete an input or output local concept that belonging to a particular local service.
2. Click on name of local service in the List of Local Service. For example, local service “GRN Detail List” is highlighted upon selected, and the list of input/output concepts that belonging to the service are displayed.
3. To add a new concepts for the service, select Input or Output at “Input/Output”, Local Concept (as registered in previous screens), and click on the “Add Concept” link button to proceed.
4. To delete existing input or output concept, click on “Delete” link button at particular row that represented the concept to be deleted.
4.4 Map Concepts

Integration maintainer is allowed to map a local concept to a global concept which has been registered in the CM module. Click on the “Map Concepts” from right navigation menu of the screen to view a range of mapped or unmapped local concepts.

Steps:
1. As refer to Figure D.25, select a range of local concepts.
2. Click on “Filter” button to filter the list based on selected range, or click on “Show All” button to list down complete list of local concepts as registered for ERP system.
3. Click on “Map” link button of at a particular row of the list. The concept is highlighted and another 2 buttons (“Update” and “Cancel”) appeared.
4. Choose the corresponding global concept to the selected local concept.
5. Click on “Update” link button to precede the concept mapping.

![Figure D.25: Local and Global Concept Mapping](image-url)
4.5 Map Services

As similar to concepts mapping, integration maintainer is also allowed to map a local concept to a global service which has been registered in the CM module. Click on the “Map Services” from right navigation menu of the screen to view a list of mapped or unmapped local services.

Steps:
1. As refer to Figure D.26, list of registered local services in earlier function is shown.
2. Click on “Map” link button of at a particular row of the list. The service is highlighted and another 2 buttons (“Update” and “Cancel”) appeared.
3. Choose the corresponding global service to the selected local service.
4. Click on “Update” link button to precede the service mapping.
SECTION 5 – INVOKE SHARED SERVICE

A demonstration on how a SCM user (Supplier001) invokes a shared service (named “Outstanding PO”) provided by ERP system is described on this section. Any other service sharing among ERP, CRM and SCM systems are follow the same steps of this demonstration.

5.1 Login

Steps:
1. Open an Internet Explorer browser, type in SCM System interface URL, for example:
   http://SCM_Server/SCAIM_SCM/Login.aspx
2. Key in User’s ID and Password into respective column as shown in Figure D.27. For demon purposes, enter “Supplier001” for User ID and “password” for Password. After that, clicks on Login button to proceed. Upon login successfully, the screen will be redirect to the main page of SCM interface – External Services, as shown in Figure D.28.
5.2 External Services

**Steps:**

1. Choose name of global service to be invoked from the drop down list as shown in Figure D.28. The information such as description of service, provider, and web method are displayed based on the selection.
2. Click on “**Show Concepts**” button to display all configured input or output concepts that belong to the selected global service.
3. To invoke the selected service, click on “**Invoke Service**” button to proceed.
4. The page is redirecting to the page of ASC module, as shown in Figure D.29.
5. As shown in the figure, there are 2 input concepts (*Supplier_Identifier* and *Vendor_PO_Number*) as configured in the previous screen.
6. Enter the criteria to the range of the input concepts. Click “Invoke” button to proceed.
7. A message box is prompted for SCM user to save the output EDI file into local directory. Click “**Save**” button and select a local directory to precede the saving, as depicted in Figure D.30.
8. Figure D.31 shows a sample EDI output of “Outstanding PO” service provided by ERP system invoked by SCM user.
Figure D.29: SCM User Invoking an ERP Service - Outstanding PO

Figure D.30: Message Box Prompted for Saving Output EDI

Figure D.31: EDI Output File Provided by ERP, Invoked by SCM User
Appendix E: SCAIM Component Source Codes

1.0 Class of Generating EDI Output

Table E.1: Source Code for EDI Output Generation

```csharp
public class EDIFACT_RECADV
{
    static int intCol_DataType = 1, intCol_EDISegment = 4,
    intCol_EDISegmentPosition = 5,
    intCol_EDIElementPosition = 6, intCol_EDIPrefix =
    7;

    #region Constructor
    public EDIFACT_RECADV()
    {
    }
    #endregion

    public void ConvertDtToREVADV(DataTable dt,
    System.IO.TextWriter httpStream, DataTable dtOriginalField)
    {
        object objBGMSkipRepeat = "START";
        bool blnSkipRepeatHeader = false;
        //Message Header
        httpStream.WriteLine("UNH+RECADV1+REC
        ADV:D:08B:UN:EAN005'");
        //Start Header Segments
        for (int j = 0; j < dt.Rows.Count; j++)
        {
            object objNAD = null, objNAD1 = null,
            objLIN = null, objLIN1 = null, objQTY = null,
            objQTY1 = null;
            for (int i = 0; i < dtOriginalField.Rows.Count; i++)
            {

                //BGM
                if (dtOriginalField.Rows[i][intCol_EDISegment].ToString()
                == "BGM" &&
                dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString()
                == "0020" &&
                dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString()
                == "1004" &&
                dtOriginalField.Rows[i][intCol_EDIPrefix].ToString()
                == "BGM+632+")
                {
                    object obj = dt.Rows[j][i];
                    if (GetWriteableValue(objBGMSkipRepeat)
                    == GetWriteableValue(obj))
                    {
                        blnSkipRepeatHeader = true;
                        continue;
                    }
                    else blnSkipRepeatHeader = false;
                    objBGMSkipRepeat = obj;
                    obj = GetWriteableValue(obj);
                    httpStream.WriteLine("BGM+632+" + obj +
                    "+9'";
                }
```
Table E.1, continued

```
//DTM - 0030 - 2380 - DTM+17:
else if
(dtOriginalField.Rows[i][intCol_EDISegment].ToString()
  == "DTM" &&
  dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString()
  == "0030" &&
  dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString()
  == "2380" &&
  dtOriginalField.Rows[i][intCol_EDIPrefix].ToString()
  == "DTM+17:"
{
    if (blnSkipRepeatHeader) continue;
    object obj = dt.Rows[j][i];
    obj = GetWriteableValue(obj);
    if
(dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper()
  == "DATETIME" ||
  dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper()
  == "SMALLDATETIME")
    {
      obj = modCommon.gfFormatDateyyyyMMddhhmm(obj);
    }
    httpStream.WriteLine("DTM+17:" + obj + ":203");
}
//RFF
else if
(dtOriginalField.Rows[i][intCol_EDISegment].ToString()
  == "RFF" &&
  dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString()
  == "0080" &&
  dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString()
  == "1154" &&
  dtOriginalField.Rows[i][intCol_EDIPrefix].ToString()
  == "RFF+VN:")
{
    if (blnSkipRepeatHeader) continue;
    object obj = dt.Rows[j][i];
    obj = GetWriteableValue(obj);
    httpStream.WriteLine("RFF+VN:" + obj + "'");
}
//DTM - 0090 - 2380 - DTM+4:
else if
(dtOriginalField.Rows[i][intCol_EDISegment].ToString()
  == "DTM" &&
  dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString()
  == "0090" &&
  dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString()
  == "2380" &&
  dtOriginalField.Rows[i][intCol_EDIPrefix].ToString()
  == "DTM+4:")
{
    if (blnSkipRepeatHeader) continue;
    object obj = dt.Rows[j][i];
    obj = GetWriteableValue(obj);
    if
(dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper()
  == "DATETIME" ||
```
Table E.1, continued

    dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper() == "SMALLDATETIME")
    {
        obj = modCommon.gfFormatDateyyyyMMddhhmm(obj);
    }
    httpStream.WriteLine("DTM+4:" + obj + ":203'");
}

else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "MOA" &&
            dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0110" &&
            dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "5004" &&
            dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "MOA+79:")
    {
        if (blnSkipRepeatHeader) continue;
        object obj = dt.Rows[j][i];
        obj = GetWriteableValue(obj);
        httpStream.WriteLine("MOA+79:" + obj + ":203'");
    }

    //MOA - 0110 - 5004 - MOA+79:

    else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "NAD" &&
            dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0190" &&
            dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "3039" &&
            dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "NAD+SU+")
    {
        if (blnSkipRepeatHeader) continue;
        objNAD = dt.Rows[j][i];
        objNAD = GetWriteableValue(objNAD);
    }

    //NAD - 0190 - 3039 - NAD+SU+

    else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "NAD" &&
            dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0190" &&
            dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "3036" &&
            dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "NAD+SU+")
    {
        if (blnSkipRepeatHeader) continue;
        objNAD1 = dt.Rows[j][i];
        objNAD1 = GetWriteableValue(objNAD1);
        httpStream.WriteLine("NAD+SU+" + objNAD + ":9+++" + objNAD1 + ":203'");
    }
//NAD + NAD1 at 1 line
}
//NAD - 0190 - 3036 - NAD+SU+
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "NAD" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0190" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "3039" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "NAD+ST+")
{
    if (blnSkipRepeatHeader) continue;
    object obj = dt.Rows[j][i];
    obj = GetWriteableValue(obj);
    httpStream.WriteLine("NAD+ST+");
}
//LIN - 0780 - 1082 - LIN+
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "LIN" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0780" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "1082" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "LIN+")
{
    objLIN = dt.Rows[j][i];
    objLIN = GetWriteableValue(objLIN);
}
//Start Detail segment
//LIN - 0780 - 7140 - LIN+
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "LIN" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0780" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "7140" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "LIN+")
{
    objLIN1 = dt.Rows[j][i];
    objLIN1 = GetWriteableValue(objLIN1);
    httpStream.WriteLine("LIN++" + objLIN + ":SRV");
} //LIN + LIN1 at 1 line
}
//PIA - 0790 - 7140 - PIA+1+
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "PIA" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0790" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "7140" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "PIA+1+")
{
object obj = dt.Rows[j][i];
obj = GetWriteableValue(obj);
httpStream.WriteLine("PIA+1:" + obj + "::IN::92");
}

//QTY - 0810 - 6060 - QTY+21:
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "QTY" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0810" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "6060" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "QTY+21:")
{
    obj = dt.Rows[j][i];
    obj = GetWriteableValue(obj);
    httpStream.WriteLine("QTY+21:" + obj + ":DZN");
}

//QTY - 0810 - 6060 - QTY+48:
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "QTY" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0810" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "6060" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "QTY+48:")
{
    objQTY = dt.Rows[j][i];
    objQTY = GetWriteableValue(objQTY);
}

//QTY - 0810 - 6411 - QTY+48:
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "QTY" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0810" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "6411" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "QTY+48:")
{
    objQTY1 = dt.Rows[j][i];
    objQTY1 = GetWriteableValue(objQTY1);
    httpStream.WriteLine("QTY+48:" + objQTY + "++" +
                        objQTY1 + ":DZN"); //LIN + LIN1 at 1 line
}

//DTM - 0830 - 2380 - DTM+1::
else if (dtOriginalField.Rows[i][intCol_EDISegment].ToString() == "DTM" &&
dtOriginalField.Rows[i][intCol_EDISegmentPosition].ToString() == "0830" &&
dtOriginalField.Rows[i][intCol_EDIElementPosition].ToString() == "2380" &&
dtOriginalField.Rows[i][intCol_EDIPrefix].ToString() == "DTM+1:")
{
object obj = dt.Rows[j][i];
obj = GetWriteableValue(obj);
if (dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper() == "DATETIME" ||
dtOriginalField.Rows[i][intCol_DataType].ToString().ToUpper() == "SMALLDATETIME")
{
    obj = modCommon.gfFormatDateyyyyMMddhhmm(obj);
}
httpStream.WriteLine("DTM+1:" + obj + ":203");

public static string GetWriteableValue(object o)
{
    if (o == null || o == Convert.DBNull)
        return "";
    else if (o.ToString().IndexOf("," == -1)
        return o.ToString();
    else
        return "\"" + o.ToString() + "\"";
}
}
2.0 Dynamic Proxy Class Generator for Web Service

Table E.2: Class of Web Service Dynamic Proxy

```csharp
using System;
using System.CodeDom;
using System.CodeDom.Compiler;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Reflection;
using System.Text;
using System.Web;
using System.Web.Services.Description;
using System.Xml.Schema;
using Microsoft.CSharp;
using Microsoft.Web.Services3;

namespace SCAIM_ASC_DynamicWSLibrary
{
    public class clsDynamicWSProxy<IRequest> where IRequest : class
    {
        private const string ProxyNameSpace = "clsDynamicWSProxy";
        private const string ListFullClassName = "System.Collections.Generic.List<{0}>";
        private Uri _wsdlLocation;
        private DiscoveryProtocol _protocolName = DiscoveryProtocol.Soap;
        private Type _serviceContract;

        public clsDynamicWSProxy(string wsdlLocation) :
            this(new Uri(wsdlLocation), DiscoveryProtocol.Soap) { }

        public clsDynamicWSProxy(string wsdlLocation,
            DiscoveryProtocol protocolName) :
            this(new Uri(wsdlLocation), protocolName) { }

        public clsDynamicWSProxy(Uri wsdlLocation) :
            this(wsdlLocation, DiscoveryProtocol.Soap) { }

        public clsDynamicWSProxy(Uri wsdlLocation,
            DiscoveryProtocol protocolName)
        {
            _wsdlLocation = wsdlLocation;
            _protocolName = protocolName;
            _serviceContract = typeof(IRequest);
        }
        public bool UseGenericList;
    }
}```
Table E.2, Continued

```
public IRequest Build()
{
    if (_wsdlLocation == null)
        throw new ArgumentNullException("WsdlLocation",
            "The argument 'WsdlLocation' can't be null");
    ServiceDescriptionImporter importer =
        new ServiceDescriptionImporter();
    importer.ProtocolName = _protocolName.ToString();
    //Get WSDL
    DiscoverService(importer);
    //Proxy source code generation
    Assembly assembly;
    using (CSharpCodeProvider codeProvider =
        new CSharpCodeProvider())
    {
        assembly = GenerateProxyAssembly(codeProvider,
            GenerateProxySourceCode(importer, codeProvider));
    }
    //Clean up resources
    importer = null;
    return Activator.CreateInstance(assembly.GetTypes()[0])
        as IRequest;
}

private void DiscoverService(ServiceDescriptionImporter importer)
{
    using (DiscoveryClientProtocol discoClient =
        new DiscoveryClientProtocol())
    {
        discoClient.DiscoverAny(_wsdlLocation.ToString());
        discoClient.ResolveAll();
        foreach (object value in discoClient.Documents.Values)
        {
            if (value is ServiceDescription)
                importer.AddServiceDescription(((ServiceDescription)value,
                    null, null);
            if (value is XmlSchema)
                importer.Schemas.Add(((XmlSchema)value);
        }
    }
    GenerateProxySourceCode(ServiceDescriptionImporter importer,
        CSharpCodeProvider codeProvider)
    {
        CodeNamespace mainNamespace =
            new CodeNamespace(ProxyNameSpace);
        importer.Import(mainNamespace, null);
        if (UseGenericList)
            ChangeArrayToGenericList(mainNamespace);
        AddImports(mainNamespace);
        CodeTypeDeclaration[] temp =
            new CodeTypeDeclaration[mainNamespace.Types.Count];
        mainNamespace.Types.CopyTo(temp, 0);
    }
```
private string
    foreach (CodeTypeDeclaration declarationType in temp)
    {
        if ((declarationType.BaseTypes.Count > 0) &&
            (declarationType.BaseTypes[0].BaseType == typeof(SoapHttpClientProtocol).FullName))
            //WebServiceClientProtocol
            declarationType.BaseTypes.Add(_serviceContract);
        else
            mainNamespace.Types.Remove(declarationType);
    }
    //Clean up resources
    temp = null;
    //Proxy source code generation
    StringBuilder sourceCode = new StringBuilder();
    string ret;
    using (StringWriter sw = new StringWriter(sourceCode, CultureInfo.CurrentCulture))
    {
        codeProvider.GenerateCodeFromNamespace(mainNamespace, sw, null);
        ret = sourceCode.ToString();
        sw.Close();
    }
    //Clean up resources
    mainNamespace = null;
    sourceCode = null;
    return ret;
}
private void ChangeArrayToGenericList(CodeNamespace codeNamespace)
{
    CodeTypeDeclaration[] temp =
        new CodeTypeDeclaration[codeNamespace.Types.Count];
    codeNamespace.Types.CopyTo(temp, 0);
    foreach (CodeTypeDeclaration type in temp)
    {
        foreach (CodeTypeMember member in type.Members)
        {
            CodeMemberMethod methodMember = member
                as CodeMemberMethod;
            if (methodMember != null)
            {
                foreach (CodeParameterDeclarationExpression paremeter in methodMember.Parameters)
                {
                    if (paremeter.Type.ArrayElementType != null)
                    {
                        paremeter.Type = new CodeTypeReference(String.Format(CultureInfo.InvariantCulture, ListFullClassName, paremeter.Type.ArrayElementType.BaseType));
                    }
                }
            }
        }
    }
}
if (methodMember.ReturnType.ArrayElementType != null) {
    string listGenericType = String.Format(CultureInfo.InvariantCulture, _fullClassName, methodMember.ReturnType.ArrayElementType.BaseType);
    methodMember.ReturnType = new CodeTypeReference(listGenericType);
    CodeMethodReturnStatement returnStatement = null;
    foreach (CodeStatement code in methodMember.Statements) {
        returnStatement = code as CodeMethodReturnStatement;
        if (returnStatement != null)
            break;
    }
    if (returnStatement != null)
        methodMember.Statements.Remove(returnStatement);
    methodMember.Statements.Add(new CodeMethodReturnStatement(new CodeArgumentReferenceExpression(String.Format(CultureInfo.InvariantCulture, "{{0}}results[0]", listGenericType))));
}
}

private void AddImports(CodeNamespace codeNamespace) {
    MethodInfo[] methods = _serviceContract.GetMethods();
    foreach (MethodInfo method in methods) {
        AddImport(codeNamespace, method.ReturnType);
        foreach (ParameterInfo parameter in method.GetParameters())
            AddImport(codeNamespace, parameter.ParameterType);
    }
}

private void AddImport(CodeNamespace codeNamespace, Type type) {
    if (!type.IsPrimitive) {
        codeNamespace.Imports.Add(new CodeNamespaceImport(type.Namespace));
        if (type.IsGenericType) {
            Type[] genericArgs = type.GetGenericArguments();
            foreach (Type genericArg in genericArgs)
                AddImport(codeNamespace, genericArg);
        }
    }
}
private Assembly GenerateProxyAssembly(CSharpCodeProvider csharpCodeProvider, string proxyCode)
{
    //Assembly compilation
    string location = String.Empty;
    if (HttpContext.Current != null)
    {
        location += @"\bin";
    }

    CompilerParameters parameters = new CompilerParameters();
    parameters.ReferencedAssemblies.Add("System.dll");
    parameters.ReferencedAssemblies.Add("System.Xml.dll");
    parameters.ReferencedAssemblies.Add("mscorlib.dll");
    parameters.ReferencedAssemblies.Add(Assembly.GetExecutingAssembly().Location);
    GetReferencedAssemblies(_serviceContract.Assembly, parameters);
    parameters.GenerateExecutable = false;
    parameters.GenerateInMemory = false;
    parameters.IncludeDebugInformation = false;
    parameters.TempFiles = new TempFileCollection(Path.GetTempPath());
    CompilerResults cr = csharpCodeProvider.CompileAssemblyFromSource(parameters, proxyCode);
    if (cr.Errors.Count > 0)
    {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < cr.Errors.Count; i++)
        {
            sb.Append(Environment.NewLine);
            sb.Append("'");
            sb.Append(String.Format(CultureInfo.InvariantCulture, "{0} - ", cr.Errors[i].ErrorNumber, cr.Errors[i].ErrorText));
            sb.Append("'");
        }
        throw new ApplicationException(String.Format(CultureInfo.InvariantCulture, "Building web service proxy has failed with {0} errors. Errors:", cr.Errors.Count, sb.ToString()));
    }
    Assembly ret = cr.CompiledAssembly;
    //Clean up resources
    cr = null;
    parameters = null;
    return ret;
}
private static void GetReferencedAssemblies(Assembly assembly, CompilerParameters parameters)
{
    if (!parameters.ReferencedAssemblies.Contains(assembly.Location))
    {
        string location = Path.GetFileName(assembly.Location);
        if (!parameters.ReferencedAssemblies.Contains(location))
        {
            parameters.ReferencedAssemblies.Add(assembly.Location);
            foreach (AssemblyName referencedAssembly in assembly.GetReferencedAssemblies())
            {
                GetReferencedAssemblies(Assembly.Load(referencedAssembly.FullName), parameters);
            }
        }
    }
}

public enum DiscoveryProtocol
{
    Soap,
    Soap12,
    HttpGet,
    HttpPost,
    HttpSoap
}
Appendix F: SCAIM System Evaluation Form

System Evaluation Form

Integrating ERP, SCM and CRM using Service Oriented Methodology

Section A: Personal Particulars

Please ☐ your answer where applicable. (Click on the check box to select)

Name: _________________________________________________________________

Gender:  □ Male  □ Female

Job Title / Role:  □ Executive Manager (CEO / COO / Director / LOB)
    □ CIO / CTO
    □ IT / IS Manager
    □ IT Architect / Developer
    □ System Maintainer
    □ IT Consultant
    □ Business Manager
    □ Business Analyst
    □ Other (Please specify) ________________________________

Company Name: ________________________________________________________

Company URL: _________________________________________________________

Contact No.: (Office) _____________________ (Mobile) _____________________

Email Address: _________________________________________________________
Section B: Evaluation Questions

Please ☐ your answer where applicable.

1. In your opinion, does the system have the capability to interpret information shared by another system accurately as expected?
   ☐ Yes  ☐ No
   If Yes, please describe the function shortly. ___________________________________
   ________________________________________________________________________

2. By configuring the integration system properly, do you think the integration system is able to share existing functionalities of one participated system to another one? (For example, user of SCM system is able to checking outstanding purchase order of their client accurately by invoking the service shared by ERP system)
   ☐ Yes  ☐ No
   If Yes, please clarify the shared functions. ________________________________
   ________________________________________________________________________

3. Scenario as stated in question (2) is an example of service sharing between ERP and SCM system, which involves company staff and supplier. Another example is about customer provides their satisfaction level through CRM system and company staff grab the information through the integration system. In your opinion, is the integration system capable to encourage the collaboration between the ERP system with supplier and customer?
   ☐ Yes  ☐ No

4. In general, do you think the integration system supports open standard of information exchange format that is understandable and readable by most of the users from ERP, CRM and SCM system?
   ☐ Yes  ☐ No

5. Is the user interfaces clear, easy to understand and support for system change with minimum modification?
   ☐ Yes  ☐ No
   If Yes, please explain ONE feature to support your answer. ____________________
   ________________________________________________________________________
6. Based on your personal experience of using the integration system, please indicate your level of agreement of each of the statement below, where: - (Please circle the number)

1 represents “strongly disagree”
2 represents “somewhat disagree”
3 represents “average”
4 represents “generally agree”
5 represents “strongly agree”

a. Protect confidential for shared information during the service (functionality) sharing activities
   1  2  3  4  5

b. Improve integrity, reliability, accessibility, and availability of independent systems
   1  2  3  4  5

c. Provide user-friendly interface, easy to understand user manual, and clear indicate language
   1  2  3  4  5

d. Response time to retrieve information produce of shared service in reasonable time
   1  2  3  4  5

e. Architecture, algorithm, and procedures design are extendable and modify with ease
   1  2  3  4  5