A Model of Enterprise Architecture Evolution

Ayed Said Alwadain

Thesis submitted to the Information Systems School, Faculty of Science and Engineering in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

at

Queensland University of Technology

Supervised by:

Prof. Michael Rosemann

Dr. Erwin Fielt

Dr. Axel Korthaus

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made.

QUT Verified Signature

Ayed Alwadain
April 2014

I dedicate this thesis to

my parents, my wife, my kids, my sisters and brothers

For their endless love, support and encouragement!

You give me hope, perspective, direction and light!

Acknowledgement

This PhD journey would not have been achievable without the support of many wonderful people, to only some of whom it is possible to give particular mention here. Thus, it is with great pleasure that I take this opportunity to sincerely thank those many people involved in helping me during this journey.

First, I express my deepest gratitude to my principal supervisor, Professor Michael Rosemann. Thank you for believing in me, and for your tremendous support, encouragement, patience, and understanding. Throughout this journey, you have been a great source of inspiration and strength to me.

I also thank my associate supervisors, Dr. Erwin Fielt and Dr. Axel Korthaus. Your helpful advice and support has been invaluable both academically and personally, for which I am extremely grateful.

I further thank the individual participants and organisations who generously took part in the research activities, and for providing the rich and invaluable insight necessary to fulfil the research objectives.

It is with immense gratitude that I acknowledge the financial support of the King Saud University, and the academic and technical support of the Queensland University of Technology.

I am indebted to many academic staff at QUT's Information Systems School. Special thanks to Professor Jan Recker, Professor Glenn Stewart, Associate Professor Alexander Dreiling, and Dr. Wasana Bandara for their constructive feedback and insightful comments at various phases of this research. Further, I wish to thank my final seminar panellists, Professor Alistair Barros (QUT) and Associate Professor Liisa von Hellens (Griffith University).

Appreciation and very special thanks also go out to the administration staff of the School of Information Systems and the QUT Science and Engineering faculty for their ongoing support, encouragement, and friendship.

I am grateful to my many colleagues who supported me. Stephan Clemens, Thomas Kohlborn, Norizan Safrudin, Eike Bernhard, Rafaele Conforti, and

Andrea Malsbender have all extended their support in a very special way. They provided helpful suggestions at various points in my research, and I learned a lot from our interactions.

Above all, I am most grateful to my parents, Said and Bakheeta, and my wife, Nora, for their love, constant inspiration, support, and encouragement. Their generous sacrifices, ongoing support, loyalty, and patience have enabled this momentous achievement.

Last, but by no means least, I thank my friends in Saudi Arabia, United Arab Emirates, Australia and elsewhere for their support and encouragement.

A very special thank to all of you!

Abstract

Organisations in today's global environment face many challenges such as innovations, new technologies, and new business models. Not surprisingly, in recent years, interest in enterprise architecture (EA) as a means to systematically consolidate and manage organisation artefacts has increased. Indeed, organisations require a well-designed and maintained EA in order to achieve their business goals and maintain a competitive advantage. However, organisations change dynamically, and EA needs to evolve along with them. EA currently faces the challenge of responding to emerging business and IT capabilities such as virtual enterprises, service-orientation, and cloud computing and of embedding them in existing EA frameworks in order to continually evolve EA and truly represent current organisational elements and their relationships. This thesis distinguishes between two levels of EA-related change: changes in architectural descriptions (such as EA meta-model, frameworks, and methods changes), and representational (EA content) changes such as changes in application details. Some studies have addressed the latter. However, little is known about the former, which the thesis calls "EA evolution". The architectural description changes present significant challenges to organisations, particularly with the growth of new business and technology capabilities.

In order to understand EA evolution, this thesis develops a theoretical model that describes the EA evolution process and explains EA evolution outcomes. It focuses on SOA introduction as a major paradigm impacting EA and requiring EA's evolution. This study analyses SOA because (1) EA needs to evolve to address and integrate SOA, (2) SOA's integration into EA has been a challenging topic, and (3) there is a lack of understanding of the relationship between EA and SOA.

In order to theorise about EA evolution, this thesis recognises both EA evolution's complexity and temporality, and the need for an analytical lens for comprehending its evolution. The study examines EA evolution using Archer's (1995) morphogenetic theory to understand the evolution process through SOA introduction. This thesis employs the morphogenetic theory because this theory considers an explicit temporal dimension to study change, which is fitting for investigating EA evolution (the need for a longitudinal investigation). Using Archer's (1995) theory, this thesis views EA evolution as an interaction between existing structural settings (existing EA) and the action of introducing new business or IT capability into an organisation (here SOA), which results in EA evolution outcomes (here SOA's integration into EA outcomes).

The morphogenetic theory is built on critical realism (CR) philosophical foundations, and this thesis thus adopts the same philosophical foundation and an iterative five-stage critical realist methodological framework to guide its overall conduct. A qualitative approach was undertaken to collect and analyse the empirical data for this thesis (namely, explorative interviews followed by multiple case studies) because of the explorative nature of this thesis and the complexity of investigating EA evolution (an open system issue).

This thesis makes several theoretical contributions. It develops the first theoretical model that describes EA evolution and explains EA evolution outcomes. It identifies and classifies nine generative mechanisms (factors) that influence EA evolution outcomes. It further identifies and classifies EA evolution outcomes into five levels.

The implications that derive from this thesis are important for both theory and practice. At a theoretical level, the developed theoretical model extends the body of knowledge on EA evolution and opens new avenues for research. It extensively describes EA evolution and explains its evolution outcomes. By doing so, it builds a foundation to further examine EA evolution beyond SOA due to the emergence of, for example, cloud computing or enterprise mobility. At a practical level, the thesis delivers a model that can be used as a guidance tool by professionals to manage EA and continually evolve it in response to emerging business and IT capabilities.

Key Words

Enterprise Architecture, EA, Enterprise Architecture Evolution, Service-Oriented Architecture, SOA, Critical Realism, Archer's Morphogenetic Theory

Publications From This Research

While pursuing the research described in this thesis until the beginning of 2014, a total of six refereed scholarly articles related to this research have been published in conference proceedings and a journal:

- Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013). A Comparative Analysis of the Integration of SOA Elements in Widely-Used Enterprise Architecture Frameworks. *International Journal of Intelligent Information Technologies (IJIIT), 9*(2), 54-70.

- Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013). *A Critical Realist Perspective of Enterprise Architecture Evolution: Preliminary Findings*. Paper presented at the The 24nd Australasian Conference on Information Systems (ACIS 2013). Melbourne, Australia

- Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013). *Service-Oriented Architecture Integration Within Enterprise Architecture: A-Priori Model*. Paper presented at the The 24nd Australasian Conference on Information Systems (ACIS 2013). Melbourne, Australia.

- Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2011). *Where do we find services in enterprise architectures? A comparative approach.* . Paper presented at the The 22nd Australasian Conference on Information Systems (ACIS 2011). Sydney, Australia.

- Alwadain, A., Rosemann, M., Fielt, E., & Korthaus, A. (2011). *Enterprise Architecture And The Integration Of Service-Oriented Architecture*. Paper presented at the The 15th Pacific Asia Conference on Information systems (PACIS 2011)
- Alwadain, A., Korthaus, A., Fielt, E., & Rosemann, M. (2010). *Integrating SOA into an Enterprise Architecture – A Comparative Analysis of Alternative Approaches*. Paper presented at the The Fifth IFIP International Conference on Research and Practical Issues of Enterprise Information Systems., Natal, Brazil.

Table of Contents

ACKNOWLEDGEMENT .... **III**

ABSTRACT .... **V**

PUBLICATIONS FROM THIS RESEARCH .... **VII**

TABLE OF CONTENTS .... **VIII**

LIST OF FIGURES .... **XVII**

LIST OF TABLES .... **XX**

CHAPTER 1: INTRODUCTION **1**

1.1 Introduction 1

1.2 Background 2

1.3 Motivation and Research Questions 5

1.4 Theoretical Foundation 8

1.5 Philosophical Foundations 11

1.6 Research Methodology 12

1.7 Contributions 13

1.8 Thesis Structure 14

1.9 Chapter Summary 14

CHAPTER 2: LITERATURE REVIEW **17**

2.1 Introduction 17

2.2 Enterprise Architecture 17

2.2.1 EA Definition 18

2.2.2 Enterprise Architecture's Development 19

2.2.3 EA Layers 19

2.2.4 EA Objectives 21

2.2.5 IEEE Standard 1471-2000 22

2.2.6 Enterprise Architecture Frameworks 23

2.2.7 Enterprise Architecture Maturity 24

2.3 Enterprise Architecture Evolution 31

2.4 Service-oriented Architecture....36
2.4.1 SOA Reference Architecture....39
2.4.2 Services....41
2.4.3 View of SOA....43
2.4.4 SOA Perceived Benefits....46
2.4.5 SOA Scope....50
2.5 SOA's Integration into the Zachman Framework....53
2.5.1 Type One: SOA in A New Column....55
2.5.2 Type Two: SOA in Nine Squares....58
2.5.3 Type Three: SOA in the Network Column....60
2.5.4 Comparison....61
2.6 SOA's Integration into Other EA Frameworks....63
2.6.1 ArchiMate....65
2.6.2 The Open Group Architecture Framework (TOGAF)....67
2.6.3 The Federal Enterprise Architecture Framework (FEAF)....68
2.6.4 The Department of Defence Architecture Framework (DoDAF)....69
2.6.5 The Ministry of Defence Architecture Framework (MODAF)....70
2.6.6 Comparison....70
2.7 SOA's Integration into EA: Other Examples....74
2.7.1 Business Architecture....75
2.7.2 Information Systems Architecture....79
2.7.3 Technology Architecture....81
2.8 Chapter Summary....82
CHAPTER 3: RESEARCH METHODOLOGY....83
3.1 Introduction....83
3.2 Philosophical Foundations....83
3.3 Critical Realism....85
3.3.1 Overview....85
3.3.2 Iterative Five-Stage Critical Realist Framework....88
3.4 Research Methods....90

3.4.1 Explorative interviews ... 92
3.4.2 Case Study Research Method ... 93
3.5 Research Plan ... 94
3.5.1 Context and Literature Review Phase ... 96
3.5.2 The Model Development Phase ... 97
3.5.3 Explorative Interviews Phase ... 97
3.5.4 Interview Data Analysis ... 98
3.5.5 Multiple Case Study Phase ... 100
3.6 Research Quality ... 109
3.6.1 Reliability ... 109
3.6.2 Validity ... 111
3.7 Chapter Summary ... 113
CHAPTER 4: RESEARCH MODEL ... 115
4.1 Introduction ... 115
4.2 Archer's Morphogenetic Theory ... 115
4.2.1 Structural Conditioning ... 118
4.2.2 Social Interaction ... 119
4.2.3 Structural Elaboration ... 119
4.2.4 Generative Mechanisms ... 120
4.2.5 Justification for Archer's Morphogenetic Theory ... 123
4.3 Building the A-Priori Model ... 126
4.3.1 Architectural Conditioning ... 127
4.3.2 Architectural Interaction ... 130
4.3.3 Architectural Elaboration ... 134
4.4 Chapter Summary ... 137
CHAPTER 5: INTERVIEW FINDINGS ... 139
5.1 Introduction ... 139
5.2 Interview Participants ... 139
5.3 Interview Analysis Process ... 143
5.3.1 Getting Familiar with Data ... 144

5.3.2 Coding 144

5.3.3 Identifying Themes 153

5.3.4 Reporting the findings 156

5.4 Architectural Conditioning 156

5.4.1 EA Framework 156

5.4.2 EA Objectives 161

5.4.3 EA Maturity 163

5.5 Architectural Interaction: SOA's Introduction 166

5.5.1 View of SOA 167

5.5.2 SOA Perceived Benefits 171

5.5.3 SOA Scope 172

5.5.4 SOA Governance 174

5.5.5 SOA Design 177

5.5.6 Business and IT Collaboration 181

5.5.7 Summary 184

5.6 Architectural Elaboration 184

5.6.1 Business Architecture 186

5.6.2 Information Systems Architecture 189

5.6.3 Technology Architecture 190

5.6.4 Enterprise Architecture Governance 191

5.6.5 Enterprise Architecture Methods and Tools 193

5.7 Chapter Summary 195

CHAPTER 6: DUBAI CUSTOMS CASE STUDY FINDINGS 197

6.1 Introduction 197

6.2 Data Collection and Analysis 198

6.3 Case Background 203

6.4 Organisation Structure 205

6.4.1 HR, Finance, and Administration 205

6.4.2 Customer Management 205

6.4.3 Policy and Legislation 206

6.4.4 Customs Cargo Operations 206

6.4.5 Customs Development 207

6.5 Architectural Conditioning 208

6.5.1 EA Framework 210

6.5.2 EA Objectives 211

6.5.3 EA Maturity 212

6.6 Architectural Interaction: SOA's Introduction 218

6.6.1 Overview of SOA's Introduction 219

6.6.2 View of SOA 221

6.6.3 SOA Perceived Benefits 223

6.6.4 SOA Scope 225

6.6.5 SOA Governance 225

6.6.6 SOA Design 226

6.6.7 Business and IT Collaboration 227

6.6.8 Summary 228

6.7 Architectural Elaboration: Reproduction or Transformation 230

6.7.1 Business Architecture (Transformed) 232

6.7.2 Information Systems Architecture (Transformed) 237

6.7.3 Technology Architecture (Transformed) 238

6.7.4 EA Governance (Transformed) 239

6.7.5 EA Methods and Tools (Transformed) 241

6.8 Summary 242

CHAPTER 7: BUSINESSLINK CASE STUDY FINDINGS 245

7.1 Introduction 245

7.2 Data Collection and Analysis 246

7.3 Case Background 252

7.4 Organisational Structure 254

7.5 Architectural Conditioning 256

7.5.1 EA Framework 256

7.5.2 EA Objectives 258

7.5.3 EA Maturity 259

7.5.4 Summary.. 261

7.6 Architectural Interaction: SOA's Introduction 262

7.6.1 Overview of SOA's Introduction 263

7.6.2 View of SOA 265

7.6.3 SOA Perceived Benefits...266

7.6.4 SOA Scope 267

7.6.5 SOA Governance 268

7.6.6 SOA Design...269

7.6.7 Business and IT Collaboration 270

7.6.8 Summary 271

7.7 Architectural Elaboration: Reproduction or Transformation 271

7.7.1 Business Architecture (Transformed) 274

7.7.2 Information Systems Architecture (Reproduced) 277

7.7.3 Technology Architecture (Reproduced)...279

7.7.4 EA Governance (Reproduced)...279

7.7.5 EA Methods and Tools (Reproduced) 280

7.7.6 Summary... 281

7.8 Chapter Summary 281

CHAPTER 8: CROSS-CASE ANALYSIS 285

8.1 Introduction 285

8.2 Contextual Description 285

8.3 Architectural Conditioning 287

8.3.1 EA Framework...287

8.3.2 EA Objectives 289

8.3.3 EA Maturity...290

8.3.4 Summary of the Architectural Conditioning Phase 294

8.4 Architectural Interaction (SOA's Introduction) 295

8.4.1 View of SOA 296

8.4.2 SOA Perceived Benefits 296

8.4.3 SOA Scope 296
8.4.4 SOA Governance 297
8.4.5 SOA Design........ 297
8.4.6 Business and IT Collaboration 298
8.4.7 Summary 302
8.5 Architectural Elaboration 303
8.5.1 Business Architecture 303
8.5.2 Information Systems Architecture........ 303
8.5.3 Technology Architecture 304
8.5.4 EA Governance........ 304
8.5.5 EA Methods and Tools........ 304
8.6 Overall Comparison Using the Three Phases of the Model 309
8.6.1 High-level Comparison 309
8.6.2 Detailed Comparison 310
8.7 Summary........ 312
CHAPTER 9: DISCUSSION........ 318
9.1 Introduction 318
9.2 EA Evolution........ 318
9.3 Architectural Conditioning (T1) 321
9.3.1 EA Framework........ 325
9.3.2 EA Objectives 326
9.3.3 EA Maturity........ 327
9.3.4 Summary 329
9.4 Architectural Interaction (T2 SOA's Introduction T3) 329
9.4.1 View of SOA 333
9.4.2 SOA Perceived Benefits 334
9.4.3 SOA Scope 334
9.4.4 SOA Governance 335
9.4.5 SOA Design........ 336
9.4.6 Business and IT Collaboration 337

9.4.7 Summary 337
9.5 Architectural Elaboration (T4) 338
9.5.1 Business Architecture 343
9.5.2 Information Systems Architecture 343
9.5.3 Technology Architecture 344
9.5.4 EA Governance 344
9.5.5 EA Methods and Tools 345
9.5.6 Summary 347
9.6 Overall Discussion 347
9.7 Summary 349
CHAPTER 10: CONCLUSION 351
10.1 Introduction 351
10.2 Overview of the Study 351
10.3 Theoretical Contributions 352
10.3.1 Architectural Conditioning (T1) 354
10.3.2 Architectural Interaction (T2-T3) 355
10.3.3 Architectural Elaboration (T4) 356
10.4 Domain Contributions 358
10.5 Limitations 360
10.6 Future Research 361
BIBLIOGRAPHY 364
APPENDIX A: INTERVIEW PROTOCOL 388
Introduction 388
Setting Up 388
Interviews Questions Divided Into Three Parts 388
Retrieve insights about the organisation, the interviewee 388
Retrieve insights about EA prior and during SOA introduction 388
Retrieve insights about SOA practices 389
Retrieve insights about SOA and EA integration 389
APPENDIX B: CASE STUDY PROTOCOL 390

Introduction 390

Candidate Participants 390

Data Collection Procedures 390

Interviews 391

Documents 391

Semi-structured Interviews Questions 391

Retrieve Insights about the Interviewee 391

EA-related Questions Prior to SOA Introduction 391

SOA Related Questions 392

View of SOA 392

SOA scope 392

SOA perceived benefits 392

SOA design 392

SOA governance 392

Business and IT collaboration 392

EA and SOA Integration Aspects 393

Business architecture 393

Information systems architecture 393

Technology architecture 393

EA governance 393

EA methods and tools 394

EA Maturity Survey 395

Field Notes Templates 400

Contact Summary Form 401

Document Summary Form 402

List of Figures

Figure 1.1 EA evolution (thesis's scope) ... 4

Figure 1.2 Mapping between the morphogenetic theory and EA evolution ... 10

Figure 1.3 Employed methodological framework (Danermark, et al., 2002) ... 12

Figure 1.4 This thesis's theoretical model ... 13

Figure 2.1 EA transformation process ... 19

Figure 2.2 Conceptual model for architectural description, IEEE Standard 1471-2000 ... 23

Figure 2.3 EA scope and value (Perko, 2008) ... 25

Figure 2.4 EA maturity model (Ross & Beath, 2006) ... 26

Figure 2.5 EAMMF overview (The US GAO, 2010) ... 27

Figure 2.6 Representational vs. architectural descriptions ((Martin, et al., 2009) ... 34

Figure 2.7 Service-oriented Architecture (Luthria & Rabhi, 2009) ... 37

Figure 2.8 SOA reference architecture (The Open Group, 2009f) ... 39

Figure 2.9 Middleware view of the SOA reference architecture (The Open Group, 2009f) ... 40

Figure 2.10 Service taxonomy (Jung, 2009) ... 42

Figure 2.11 Value and scope of SOA adoption (Campbell and Mohun, 2007) ... 51

Figure 2.12 Zachman Framework ... 54

Figure 2.13 ArchiMate (Lankhorst, 2005) ... 66

Figure 2.14 TOGAF ADM ... 68

Figure 2.15 Service-oriented enterprise (Hirschheim, et al., 2010) ... 75

Figure 2.16 Service engineering schematic (Chen, 2008) ... 76

Figure 2.17 SOA views of CBDi-SAE (Butler 2007) ... 77

Figure 2.18 Three SOEA layers (Aier & Gleichauf, 2009) ... 78

Figure 2.19 Service layer (Assmann & Engels, 2008) ... 81

Figure 2.20 N.S.W Departments of Lands' SOA ... 81

Figure 3.1 Scientific paradigms and their elements (Healy & Perry, 2000, p. 119) ... 84

Figure 3.2 CR reality domains (Mingers, 2004). 86
Figure 3.3 CR-based methodological framework (Danermark et al., 2002) 88
Figure 3.4 Research plan 95
Figure 3.5 Case design and units of analysis (Yin, 2009, p. 46) 101
Figure 3.6 A morphogenetic cycle boundaries 104
Figure 4.1 Archer's morphogenetic theory (Archer, 1995) 118
Figure 4.2 A typology of social mechanisms (Hedström & Ylikoski, 2010) 122
Figure 4.3 Types of generative mechanisms mapped to Archer's morphogenetic cycle 123
Figure 4.4 Mapping between the morphogenetic cycle and EA evolution 124
Figure 4.5 The a-priori research model 127
Figure 4.6 Value and scope of SOA adoption (Campbell and Mohun, 2007) 133
Figure 5.1 Interview analysis process 143
Figure 5.2 Coding example 144
Figure 5.3 The a-priori research model 156
Figure 5.4 Modified TOGAF [I-2] 158
Figure 5.5 Interpretability framework and EA 159
Figure 5.6 EA objectives 162
Figure 5.7 The extended a-priori model of this thesis 195
Figure 6.1 The morphogenetic cycle of SOA's integration into EA at Dubai Customs 200
Figure 6.2 Employed retrospective analysis 201
Figure 6.3 An extract of the case analysis using Nvivo 202
Figure 6.4 Organisation structure 205
Figure 6.5 This thesis's theoretical model 208
Figure 6.6 EA program implementation phases [D-1] 209
Figure 6.7 The enterprise connected view [D-1] 211
Figure 6.8 Adopted EA methodology 215
Figure 6.9 EA governance [D-1] 216
Figure 6.10 Services classification at Dubai Customs 227

Figure 6.11 Enterprise-connected view changes 234

Figure 6.12 Business services types 235

Figure 6.13 Business services accessible on the Customs website 236

Figure 6.14 The morphogenetic cycle of SOA's integration into Dubai Customs' EA............243

Figure 7.1 The morphogenetic cycle of SOA's integration into EA at Businesslink 249

Figure 7.2 Employed retrospective analysis at Businesslink...250

Figure 7.3 Snapshot of the Businesslink Case Analysis using Nvivo 251

Figure 7.4 Businesslink's operating model...255

Figure 7.5 The thesis's theoretical model 256

Figure 7.6 Architectural domains [D-9] 257

Figure 7.7 Next-generation service model initiative [D-8]...264

Figure 7.8 End-to-end service delivery [D-8] 265

Figure 7.9 Businesslink's new EA framework [D-6] 275

Figure 7.10 The morphogenetic cycle of SOA's integration into EA at Businesslink 281

Figure 8.1 This thesis's' theoretical model 287

Figure 8.2 High-level view of the morphogenetic cycle of SOA's integration into EA at Dubai Customs...309

Figure 8.3 High-level view of the morphogenetic cycle of SOA's integration into EA at Businesslink 310

Figure 8.4 The morphogenetic cycle of SOA's integration into EA at Dubai Customs 312

Figure 8.5 The morphogenetic cycle of SOA's integration into EA at Businesslink 312

Figure 9.1 This thesis's theoretical model 321

Figure 10.1 EA evolution model 353

List of Tables

Table 1.1 Theory classification (Gregor, 2006) 9

Table 1.2 Summary of chapter one........ 15

Table 2.1 E2AMM........ 28

Table 2.2 The DyA maturity matrix 29

Table 2.3 NASCIO maturity stages 29

Table 2.4 Adopted maturity model (based on NASCIO) 31

Table 2.5 Classification of the view of SOA 46

Table 2.6 SOA benefits (Yoon and Carter, 2007)........ 48

Table 2.7 SOA benefits (Mueller, Viering, Ahlemann, & Riempp (2007)........ 49

Table 2.8 SOA benefits (Becker et al., 2009) 49

Table 2.9 Combined and adapted SOA benefits........ 50

Table 2.10 SOA scope........ 53

Table 2.11 Zachman Framework and SOA: first approach 56

Table 2.12 Zachman Framework and SOA: second approach........ 57

Table 2.13 Zachman Framework and SOA: third approach 57

Table 2.14 Zachman Framework and SOA: multiple squares approach........ 59

Table 2.15 Zachman Framework and SOA: existing column approach 60

Table 2.16 Comparison of SOA integration into Zachman Framework........ 62

Table 2.17 SOA elements in EA frameworks 71

Table 3.1 Mapping between the research plan's phases and the adopted CR framework stages 96

Table 3.2 Case selection criteria........ 107

Table 3.3 Employed reliability measures 110

Table 3.4 Employed validity measures 112

Table 5.1. Interview participants' information........ 142

Table 5.2 Using NVivo for the data analysis........ 146

Table 5.3 Interviews analysis codebook 149

Table 5.4 Final themes and their associated codes 154

Table 5.5 Examples of EA frameworks used 157

Table 5.6 EA maturity 164

Table 5.7 Reported views of SOA 167

Table 5.8 Views of SOA and supporting quotes 169

Table 5.9 Reported SOA perceived benefits 171

Table 5.10 Example of service classifications 179

Table 5.11 Service identification methodologies 180

Table 5.12 Levels of SOA's integration into EA (EA elaboration) 185

Table 5.13 Cases reported EA evolution 186

Table 6.1 Participants' information 198

Table 6.2 Extra collected evidence (documents) 199

Table 6.3 EA objectives at Dubai Customs 212

Table 6.4 EA maturity prior to SOA's introduction 212

Table 6.5 Quotes of reported SOA benefits at Dubai Customs 223

Table 6.6 Reported SOA perceived benefits at Dubai Customs 224

Table 6.7 Summary of SOA's introduction at Dubai Customs 229

Table 6.8 The observed architectural evolution 230

Table 6.9 Business architecture 236

Table 6.10 Information systems architecture (transformed) 238

Table 6.11 Technology architecture (transformed) 239

Table 6.12 EA Governance 240

Table 6.13 EA methods and tools 242

Table 6.14 Contextualisation of SOA's integration into EA at Dubai Customs 244

Table 7.1 Businesslink case participants 246

Table 7.2 Businesslink obtained documents 247

Table 7.3 EA at level one of maturity (informal program) 259

Table 7.4 Workstation platform artefacts [D-10] 260
Table 7.5 Architecture forecast template [D-9] 261
Table 7.6 Quotes regarding the next generation service model benefits 266
Table 7.7 SOA perceived benefits 267
Table 7.8 The architectural elaboration 273
Table 7.9 Services groups [D-13] 276
Table 7.10 Elaboration of the business architecture of Businesslink 276
Table 7.11 Service to application mapping [D-18] 278
Table 7.12 Elaboration of the information systems architecture of Businesslink 278
Table 7.13 Contextualisation of SOA's integration into EA at Businesslink 282
Table 8.1 Case study profiles 286
Table 8.2 Case study EA frameworks 288
Table 8.3 Case study objectives 289
Table 8.4 EA maturity comparison 292
Table 8.5 Case study architectural conditioning 295
Table 8.6 SOA's introduction comparison 299
Table 8.7 Summary of SOA introduction in both cases 302
Table 8.8 Architectural elaboration comparison 306
Table 8.9 Contextualisation of the two organisations compared 316
Table 9.1 Triangulation of methodological approaches 320
Table 9.2 Summary of findings related to the architectural conditioning phase 323
Table 9.3 Summary of findings related to the architectural interaction phase 331
Table 9.4 Summary of findings related to the architectural elaboration phase 339
Table 10.1 Architectural conditioning related contributions 354
Table 10.2 Architectural interaction-related contributions 355
Table 10.3 Architectural elaboration related contributions 357

Chapter 1: Introduction

1.1 Introduction

Enterprise architecture (EA) is used to reduce organisational complexity, improve communication, align business and IT, and drive organisational change (Lankhorst, 2005; Schekkerman, 2005). A recent survey by Gartner estimated that EA practitioners strongly influence organisations' IT budgets (Gartner, 2012a). Globally, EA practitioners are either the final decision maker or greatly influence more than $1.1 trillion in enterprise IT spending. Lankhorst (2005, p. 3) defines EA as:

> *a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organisational structure, business processes, information systems, and infrastructure.*

EA needs to evolve in response to organisational business and IT changes. Moreover, it is essential to plan its evolution (MacLennan & Van Belle, 2012; McKendrick, 2010; Shah & Golder, 2011). This thesis views EA evolution as an interaction between existing structural settings (existing EA) and the action of introducing a new IT phenomenon (capability), which results in EA evolution outcomes.

EA evolution presents significant challenges to organisations (Land, Proper, Waage, Cloo, & Steghuis, 2009; Short, 2013) and research on this phenomenon is very limited. For example, there are no empirical studies that describe or explain how EA evolves due to emergent business and IT trends such as the Service-oriented Architecture (SOA) (Knippel & Skytte, 2007; McKendrick, 2010; Postina, Trefke, & Steffens, 2010; Saat, Aier, & Gleichauf, 2009; Shah & Golder, 2011).

Thus, *the aim of this thesis* is to develop a theoretical model that describes the EA evolution process and explains EA evolution outcomes. It specifically focuses on the Service-Oriented Architecture (SOA) introduction (implementation) in organisations as one exemplary trigger to lay the foundation for understanding the concept of EA evolution.

In this chapter, Section 1.2 outlines the background of this thesis. Section 1.3 discusses the motivation for this thesis and its research questions. Section 1.4 discusses this thesis's underlying theoretical foundations. Section 1.5 briefly outlines the study's philosophical foundations and Section 1.6 overviews the selected research methodology. Section 1.7 summarises this thesis's contributions, and Section 1.8 presents the thesis structure. Lastly, Section 1.9 summarises the chapter.

1.2 Background

The complexity of contemporary organisations is regularly increased by disrupting, continuing, and recurrent changes in their business, legal, and technological environment. Taking an architectural approach can help reduce this complexity and enable informed decision making processes (Lagerstorm, 2010; Winter & Fischer, 2007). Such an architectural approach is called an enterprise architecture (EA) (Lankhorst, 2005; Shah & Golder, 2011). EA expresses the different elements, domains, and their relationships in an enterprise. It creates an overview of the structure, strategies, products, business processes, applications, and the technical infrastructure in order to manage the growing complexity and expediting rates of change (Land, et al., 2009; Sowa & Zachman, 1992; Zachman, 1987). An enterprise is described using EA frameworks and architectural descriptions. These EA frameworks provide one or more meta-model(s) of the architectural elements and their relationships, one or more method(s) for EA models' design and evolution, and/or a common vocabulary (Winter & Fischer, 2007). An EA framework is defined as "a conceptual structure used to develop, implement, and sustain an architecture" (The Open Group, 2012b, p. n.a).

Over the last few decades, EA has been considered an important approach to managing and guiding the modelling process of organisational artefacts and their relationships. It is used to describe an organisation's current operational environment ("as-is"), its desired future target state ("to-be"), and the roadmap to transform the "as-is" into the "to-be" state (Buckl, Matthes, Schulz, & Schweda, 2010; Lange, 2012).

A well-designed EA is a requirement in today's global business environment for organisations to achieve their desired business goals and a competitive advantage. It is critical to the success of management tasks such as business and IT alignment, portfolio planning, and organisational governance (Lagerstorm, 2010; Winter & Fischer, 2007). It improves organisational efficiency and effectiveness in respect to an organisation's use of IT systems to achieve business objectives (Plummer & McCoy, 2006). EA also helps architects and managers to better understand the relevance, impact, and potential of new technologies for their organisations (Land, et al., 2009).

Organisations face many changes such as innovations, new technologies, and new business models (Land, et al., 2009). Business and IT trends confront organisations with critical questions regarding the relevance of these trends, and how they will change the organisations and their EAs (Land, et al., 2009; Short, 2013). In a recent Forrester Research survey, current business and IT trends such as pervasive business intelligence, networked business, virtual enterprises, service-orientation, and cloud computing are considered challenges for EA as a practice (McKendrick, 2010). EA needs to evolve and accommodate these emerging trends (Jung, 2009; McKendrick, 2010; Postina, et al., 2010; Sampaio, 2010; Sousa, Lima, Sampaio, & Pereira, 2009) so that it can accurately represent emerging trends, concepts, and relationships.

This thesis distinguishes between two levels of EA-related change: changes in architectural descriptions, and representational (content) changes. Architectural descriptions are the "vehicle" for building architectural representations (Martin, Purao, & Robertson, 2009). This thesis focuses on EA evolution (changes in architectural descriptions) that results from an introduction of new business or IT trends that brings new concepts, elements, or new ways of thinking to an organisation. New business and IT trends often include (1) changes to architectural elements and their relationships, and (2) changes to representational changes (content or instances) such as changes of applications and processes details (see Figure 1.1). However, this thesis does not consider representational changes because

(1) other studies have adequately addressed them (e.g. see Buckl, Ernst, Matthes, & Schweda, 2009; Farwick, Schweda, Breu, Voges, & Hanschke, 2012).

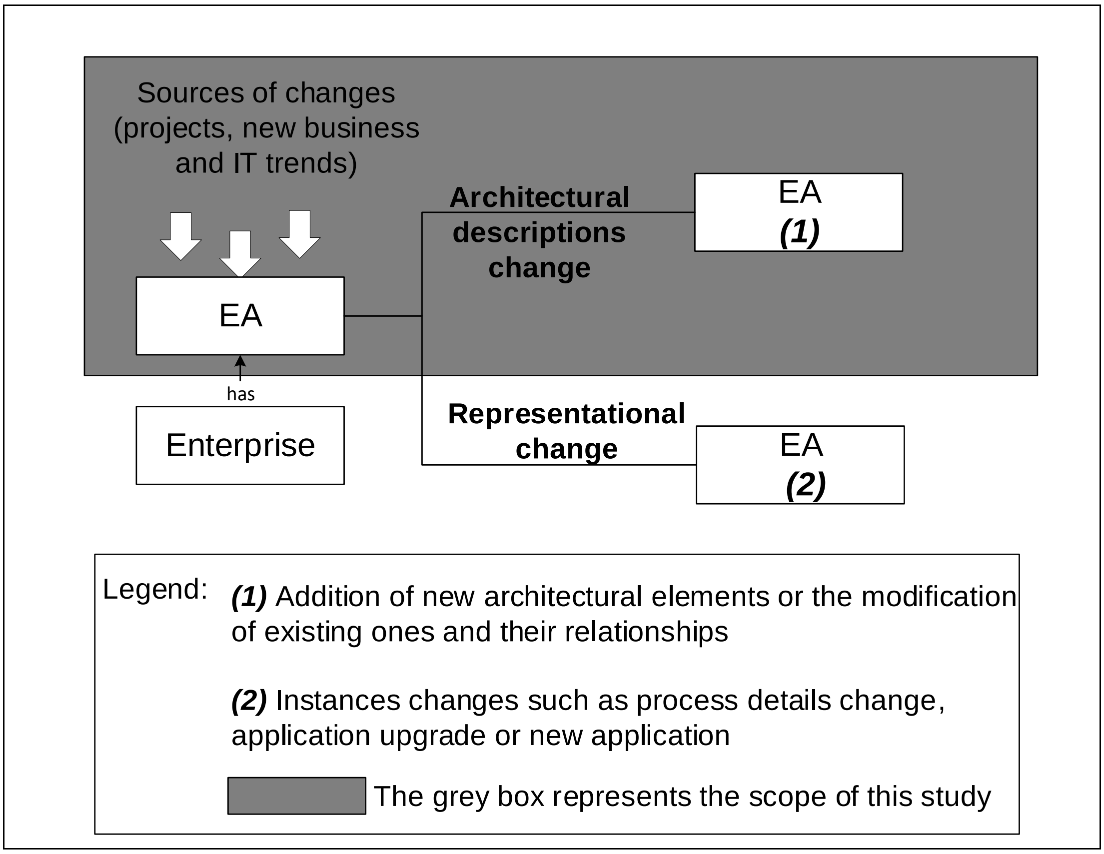


Figure 1.1 EA evolution (thesis's scope)

EA is challenged by emerging business and IT capabilities (Jung, 2009; Mens, Magee, & Rumpe, 2010; Roth, Hauder, Farwick, Breu, & Matthes, 2013) such as SOA and cloud computing (McKendrick, 2010). Thus, enterprise architects have to be aware of the impact of these emerging trends on EA (Roth, et al., 2013) and have to consider the practical means and mechanisms to competently integrate an emerging IT phenomenon into their architectures (Raj & Periasamy, 2011).

This thesis focuses on SOA as a major paradigm impacting EA and requiring EA's evolution because this thesis defines SOA as "an architectural style that supports service orientation, and service orientation is a way of thinking in terms of services and service-based development and the outcomes of services" (The Open Group, 2010).

There is a high level of interest in SOA. Major organisations have invested in SOA as a key approach to obtaining organisational agility and managing rapid changes (Chen, Kazman, & Perry, 2010). Chen et al. (2010) estimate that seventy percents of Fortune 500 companies have started enterprise SOA initiatives in recent years. Moreover, SOA integration into EA has been a challenging topic (Banerjee & Aziz, 2007; Correia & Silva, 2007; Dico, 2012; Infosys, 2009; Postina, et al., 2010; Shankararaman & Kazmi, 2011; Sharma, 2013 ; Sweeney, 2010; Varnus & Panaich, 2009). Service-orientation is among the top issues that organisations are trying to address using EA (Varnus & Panaich, 2009). Indeed, many studies explicitly argue that EA needs to evolve to address and integrate SOA (Khoshnevis, Aliee, & Jamshidi, 2009; Postina, et al., 2010; Sanders, Hamilton, & MacDonald, 2008; Sharma, 2013 ; Viering, Legner, & Ahlemann, 2009). EA needs to integrate SOA in order to properly describe current organisations. Kistasamy, Van der Merwe, and De La Harpe (2012) argue that, although both EA and SOA have been in the industry for a long period, EA longer than SOA, there is a lack of understanding of the relationship between them, which has resulted in a marginal realisation of their combined benefits (Kistasamy, et al., 2012). Dico (2012) states that SOA integration into EA needs more attention. The majority of EA programs are limited in both EA and SOA practices, and are not comprehensive enough to deal with and manage the associated complexities. These EA programs also suffer from the inability to leverage EA and SOA (Dico, 2012).

1.3 Motivation and Research Questions

Organisations have made substantial efforts to produce and use architectural models (Sampaio, 2010; Sousa, et al., 2009). The dynamic nature of organisations requires EA to evolve. In order to develop architectures that reflect the system nature of an organisation, frameworks, methodologies, and terminology used in developing EA need to take into account that organisations are adaptive systems of systems (Sampaio, 2010; Sousa, et al., 2009).

EA development is not a single activity that leads to static descriptions of an organisation. Indeed, static descriptions would obstruct the process of change. Rather, EA development changes with the evolution of the organisation and its strategy (Shah & Golder, 2011). EA needs to change over time along organisational change to provide value for stakeholders. Many of today's organisations have to confront the challenge of EA evolution. If EA evolution is not managed, EA is likely to evolve in an uncontrolled manner. EA models will become out-dated, if an organisation evolves to accommodate changes in isolation of EA (Lucke, Krell, & Lechner, 2010; Mens, et al., 2010).

Among the identified critical issues of EA management is the low quality level of EA models in terms of actuality, consistency, and completeness (Roth, et al., 2013). Despite the importance of keeping EA models evolving and up-to-date, many organisations still struggle to do so, which reduces EA's value and causes it to become a significant impediment for further EA activities (Roth, et al., 2013; Sousa, et al., 2009). Additionally, the value of these models has reduced because their value is tied to their accuracy, adequacy, and ability to convey the intended message to targeted stakeholders (Sampaio, 2010).

EA misrepresentation and occasionally even failures result when EA evolutions are not managed and aligned (Martin, et al., 2009). Related architecture models must be accurately and traceably linked to their implementation in order to manage the complexity, development, and maintenance of evolving systems. Any changes to the implementation have to be reflected back in the architecture in order to keep EA evolving and correctly describing the organisation (Mens, et al., 2010). In order to keep EA models aligned with reality, enterprise architects have to be aware of changes affecting the enterprise and its EA (Roth, et al., 2013). As a discipline, EA faces the challenge of responding to emerging business and IT capabilities and embedding them in existing EA frameworks in order to truly represent current organisational elements and their relationships (Jung, 2009; Mens, et al., 2010; Roth, et al., 2013). EA architectural elements and their relationships may require changes due to a new emerging trend. For example, new architectural elements and relationships need to be considered when new paradigms that change the enterprise emerge, such as Service-oriented Architecture (SOA) (Banerjee & Aziz, 2007; Infosys, 2009; Postina, et al., 2010; Varnus & Panaich, 2009). The architectural elements are the elements that enclose and describe an organisation's business, people, and technology (more specifically, its strategies, business principles, stakeholders, locations, functions, activities, processes, products, information, applications, systems, infrastructure, and so on) (Schekkerman, 2004, p. 22).

Unfortunately, despite the importance of continually evolving EA, few studies have discussed how EA actually evolves. The identified limited studies have either focused on the representational changes of EA, such as changes to applications (e.g. see Buckl, Ernst, Matthes, & Schweda, 2009; Farwick, et al., 2012), or provided examples of EA evolution (SOA integration into EA) without considering the underlying process of evolution or what may impact the evolution (e.g. see Banerjee & Aziz, 2007; Correia & Silva, 2007; Postina, et al., 2010; Shankararaman & Kazmi, 2011; Sharma, 2013). The existing approaches do not clearly consider external influences on EA planning process or changing conditions in an organisation's environment (Saat, et al., 2009).

Consequently, a thorough examination of EA evolution due to emerging business and IT trends such as SOA is needed (Correia & Silva, 2007; Khoshnevis, et al., 2009; Viering, et al., 2009). Legner and Heutschi (2007) argue that more work should be conducted on EA models and architecture management, which include service-based concepts. Postina, et al. (2010) also conclude that EA frameworks and languages have inadequately evolved to address SOA elements and viewpoints of SOA stakeholders. Further, Viering et al. (2009), after performing a literature survey, argue that further research is needed in order to understand EA evolution in response to SOA's emergence (Viering, et al., 2009).

Despite the remarkable need for understanding EA evolution and in particular EA evolution due to SOA introduction in an organisation, the previous paragraphs showed a lack of EA evolution studies that examine how EA evolves and, in particular, how EA evolves after SOA is introduced. As such, this thesis explores Research Question 1 (RQ1):

RQ1: *How does EA evolve as a result of the introduction of SOA?*

The literature contains diverse approaches for integrating SOA into EA. For instance, one study compared five widely used EA frameworks and showed that EAs have evolved to integrate SOA in different ways (Alwadain, Fielt, Korthaus, & Rosemann, 2011, 2013a). Additionally, Traverson (2008) argues that no clear consensus on an integration strategy of services and EA has yet been achieved. Thus, it is important to comprehend the factors that may influence EA evolution and in particular, SOA's integration into EA. As such, this thesis explores Research Question 2 (RQ2):

RQ2: *What are the factors that influence EA evolution as a result of the introduction of SOA?*

In summary, these research questions were developed in particular to address the lack of empirical studies on EA evolution which is supported by Viering et al.'s (2009) call for more research on how SOA changes EA, and others' arguments about the need to integrate SOA into EA to continually evolve EA (Khoshnevis, et al., 2009; Postina, et al., 2010; Sanders, et al., 2008; Sharma, 2013).

1.4 Theoretical Foundation

In order to add a unique, theoretical contribution to the body of EA knowledge, this thesis needs to be positioned from two perspectives: the domain (EA as a discipline/practice) and its theoretical basis (i.e., theory building or theory testing).

First, this thesis positions EA evolution as a complex, organisational level phenomenon and an important aspect of EA management (Winter, Buckl, Matthes, & Schweda, 2010). EA is still a young domain that presents many challenges for researchers. Moreover, EA studies often lack sound theoretical foundations (Schmidt & Buxmann, 2011; Simon, Fischbach, & Schoder, 2013). In particular, this thesis argues that there is lack of studies and theories that describe or explain how organisations evolve their EAs.

Second, research studies typically address theory building (i.e., discovery, description, mapping, and relationship building), theory testing, and theory extension (Handfield & Melnyk, 1998). As such, given the infancy stage of EA domain and the lack of relevant theory, this thesis builds theory.

Lynham (2000) argues for theory building (1) to advance professionalism in and maturity of a given field and (2) to help bridge the gap between research and practice. For the purposes of this thesis, we use Lynham's (2000, p. 222) definition for theory; that is, "a coherent description, explanation, and representation of observed or experienced phenomena". Unfortunately, the time and resources constraints associated with PhDs limit this thesis from addressing the stages after theory building: the theory testing and refinement stages.

Gregor (2006) classifies information systems theories into five types: (1) theory for analysing, (2) theory for explaining, (3) theory for predicting, (4) theory for explaining and predicting, and (5) theory for design and action. This thesis concerns itself with the first two types (see Table 1.1). First, it develops a theory that provides an analytical lens of how EA evolves, and describes the evolution process. Second, it explains how EA evolution outcomes are generated by providing potential generative mechanisms (factors) that may influence EA evolution outcomes

Table 1.1 Theory classification (Gregor, 2006)

Theory type	Distinguishing attributes	This thesis
Analysis	The theory does not extend beyond analysis and description. No causal relationships among phenomena are specified and no predictions are made.	This thesis builds a theory that provides analysis (analytical lens) of EA evolution and describes how EA evolves.
Explanation	The theory provides explanations, but does not aim to predict with any precision. There are no testable propositions.	This thesis provides explanation of EA evolution outcomes but does not aim to predict.
Prediction	The theory provides predictions and has testable propositions, but does not have well-developed justificatory causal explanations.	Not in the scope of this thesis.
Explanation and prediction (EP)	Provides predictions and has both testable propositions and causal explanations.	Not in the scope of this thesis.
Design and action	The theory gives explicit prescriptions (e.g., methods, techniques of form and function) for constructing an artefact.	Not in the scope of this thesis.

In order to theorise about EA evolution, this thesis recognises both the inherent complexity and the temporality of EA evolution and the need for an analytical lens for understanding its evolution. The study investigates EA evolution using the morphogenetic theory (Archer, 1995) to comprehend the evolution process through SOA introduction. This thesis adopts the morphogenetic theory, because it considers an explicit temporal dimension to study change, which is fitting for an investigation of EA evolution. It provides a useful conceptualisation approach examining organisational changes, particularly those involving technology (Mutch, 2010). It analytically represents EA evolution through the interactions between structure (in this thesis EA) and action (SOA introduction) and their operations over different time periods using three analytical phases: structural conditioning (T1), social interaction (between T2 and T3), and structural elaboration (T4) as shown in Figure 1.2.

Building on the morphogenetic theory, the elaboration (EA evolution outcomes) is the result of the interplay between the action-formation mechanisms of the interaction (SOA introduction) and the conditional generative mechanisms of the conditioning phase. Figure 1.1 shows the EA evolution process, which has three phases: (1) architectural conditioning (an organisation's EA), (2) architectural interaction (e.g., SOA introduction), and (3) architectural elaboration (EA evolution outcomes).

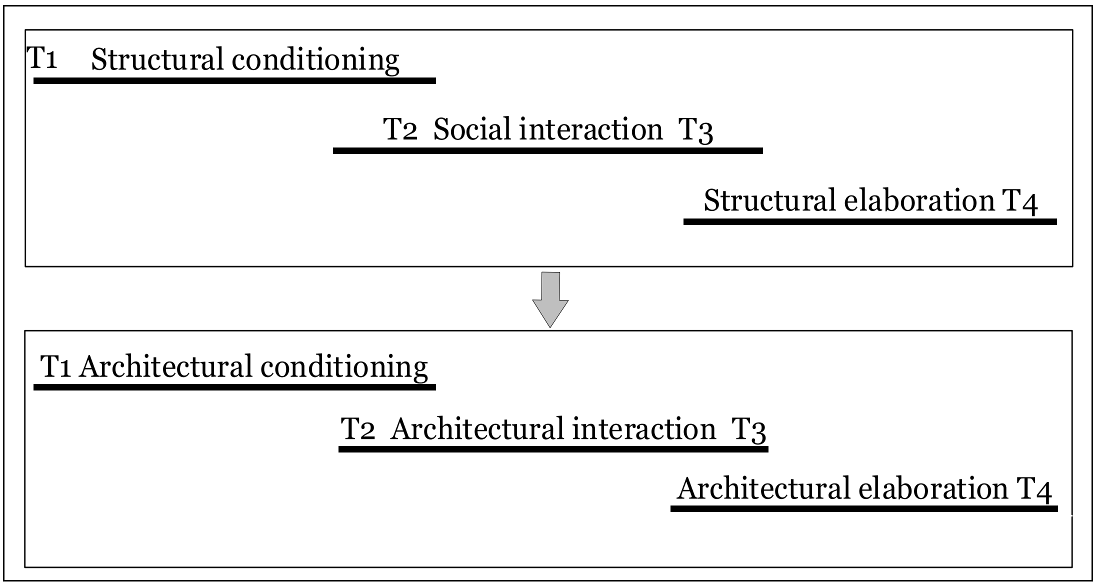


Figure 1.2 Mapping between the morphogenetic theory and EA evolution

By using Archer's (1995) morphogenetic theory, the study is able to take an overarching view to examine the complex interaction between the structure (EA) and the action (SOA's introduction) to analytically separate the interaction into three phases and to identify relevant generative mechanisms (causal powers) that affect EA evolution (SOA integration into EA).

1.5 Philosophical Foundations

The morphogenetic theory is built on critical realism (CR) as a philosophical foundations (Archer, 1995; Bhaskar, 1975; Bhaskar, 1998; Danermark, Ekström, Jakobsen, & Karlsson, 2002; Hedström & Swedberg, 1998). Critical realism is recognised as a viable philosophical paradigm for conducting social science research. CR-based studies offer opportunities to investigate complex organisational issues in a holistic manner (Wynn & Williams, 2012). A critical realism-based study explains a given set of outcomes by uncovering the hypothesised existence of mechanisms that, once actualised (activated), could have produced these outcomes (Bhaskar, 1998; Wynn & Williams, 2012). Given empirical evidence regarding a central phenomenon and context, CR endeavours to find the answer to the question: What must reality have been like in order for this outcome to have occurred? CR researchers aim to find out the mechanisms that surface from the components of interacting structures to produce the outcomes (Mingers, 2004; Sayer, 1992; Wynn & Williams, 2012). This thesis aims to find answers to EA's evolution outcomes and discover the mechanisms (factors) that interact to produce these evolution outcomes.

Further, in order to guide the overall conduct of this thesis and to uncover the mechanisms that surface from the interaction between EA and SOA introduction and thus produce different integration outcomes, an iterative five-stage critical realist methodological framework (Danermark, et al., 2002) was employed. Figure 1.3 represents the high level of use of this critical realist methodological framework in this thesis. Chapter 3 examines its use in more detail.

Stage	Purpose
1. Description	to prepare a description of the phenomenon under investigation
2. Analytical resolution	to differentiate various components of the phenomenon and set up its tentative boundaries.
3. Abduction / theoretical redescription	to redescribe the different components and their relationships using a theoretical framework.
4. Retroduction	to identify the causal powers (generative mechanisms) that may interact to generate the outcomes.
5. Concretization and contextualisation	to examine how these structures, generative mechanisms and outcomes manifest themselves in a given context.

Figure 1.3 Employed methodological framework (Danermark, et al., 2002)

1.6 Research Methodology

Given this thesis's explorative nature and EA evolution's complexity, a qualitative approach was undertaken to collect and analyse the empirical data for this thesis (namely, explorative interviews followed by multiple case studies—detailed justifications are provided in Chapter 3*)*. In general, qualitative studies are appropriate for conducting critical realism-based studies (Danermark, et al., 2002; Sayer, 1992). In particular, the case study method is an appropriate strategy of enquiry, because it is highly compatible with critical realism's underlying ontological position. It is considered suitable for exploring the interaction between structures and actions in a given context to discover causal mechanisms (Easton, 2010; Mingers, 2004; Wynn & Williams, 2012; Wynn Jr & Williams, 2008). It is congruent with the adopted theory (Volkoff, Strong, & Elmes, 2007) and is appropriate for investigating new phenomenon (Yin, 2009).

The research design has multiple phases. First, a literature review was conducted to understand the research context, its theoretical foundations and to identify research questions. Next, based on the identified literature, an a-priori research model was developed using Archer's morphogenetic theory (Archer, 1982; Archer, 1995). The morphogenetic theory was used as a lens to facilitate EA evolution conceptualisation using its temporal dimension. Then, an explorative semi-structured interview phase was carried out to further enrich the understanding of the research context and extend the a-priori model. Semi-structured interviews were used to explore EA evolution with every participant in detail, and to develop an understanding of the relevant aspects as seen from each participant's independent perspective. Then, the a-priori model was examined in two case studies to further explore the developed model in specific contexts. The case studies provide (1) the required depth to understand EA evolution in a specific context through the intensive collected evidence and (2) the triangulation of evidence sources.

1.7 Contributions

This thesis addresses EA evolution in light of changing concepts and approaches in the business and IT domains. In particular, it brings together two broad areas of study: EA and SOA. In general, this research delivers a deeper description of EA evolution and explanation of its evolution outcomes that allow for further examination of EA evolution. It builds a foundation to address EA evolution due to the emergence of, for example, cloud computing or enterprise mobility.

This thesis makes several theoretical contributions. First, the study develops the first theoretical model that describes EA evolution (analysis theory) and explains EA evolution outcomes (explanation theory). Second, it identifies and classifies nine generative mechanisms (factors) that influence EA evolution outcomes. Third, it further identifies and classifies EA evolution outcomes into five levels (see Figure 1.4).

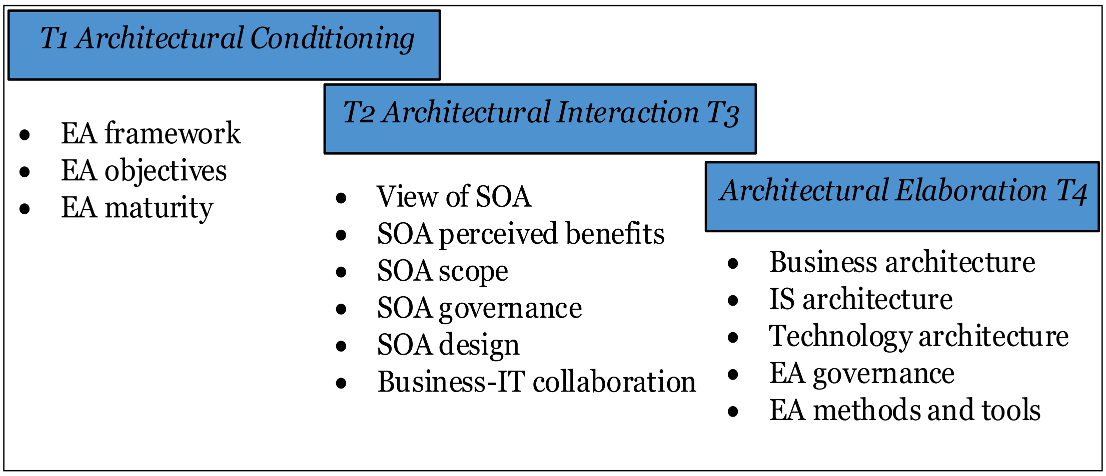


Figure 1.4 This thesis's theoretical model

Moreover, the study provides relevant practical contributions. The developed model, being the first theoretical study, explicitly unveils the complexity of the EA evolution process. It provides practitioners with an analytical model to identify how EA framework and practice need to evolve when new paradigms and trends emerge and what the required changes are. It further provides a sketch of the relevant aspects (conditional and action-formation generative mechanisms) that practitioners should consider to effectively manage EA evolution.

1.8 Thesis Structure

This chapter presents the introduction of the thesis. The rest of the thesis is organised across nine chapters. Chapter 2 covers EA, EA evolution SOA, and SOA's integration into EA. Chapter 3 presents the research approach and the methodology underpinning the study. It introduces critical realism and argues for its suitability to investigate this thesis's research problem. It also outlines the four stages of the research design process, which aligns with the adopted critical realist methodological framework.

Chapter 4 develops the a-priori model for the study using Archer's (1995) morphogenetic theory. The model is built using the relevant information discussed in Chapter 2 and the three analytical phases of Archer's theory: architectural conditioning (pre-existing EA), architectural interaction (SOA's introduction), and architectural elaboration (EA evolution outcomes), respectively.

Chapter 5 presents the interview findings. These findings refine and extend the a-priori model. Chapters 6 and 7 introduce the findings of the two case studies, while Chapter 8 presents the cross-case analysis of the two cases. Chapter 9 discusses the overall findings. Chapter 10 concludes the thesis and summarises the thesis: it examines key contributions for theory and practice, the study's limitations, and ideas for future research.

1.9 Chapter Summary

This chapter overviews the significance of evolving EA in response to emergent business and IT trends such as SOA that bring new concepts and relationships when introduced into organisations. Despite the significance of understanding how EA evolves (in particular after introducing SOA), there is a lack of empirical studies that address EA evolution. Thus, this thesis develops a theoretical model that describes EA evolution and explains its outcomes. For this purpose, and due to the inherent complexity and the temporality of EA evolution and the need for an analytical lens to understand its evolution, this thesis uses Archer's (1995) morphogenetic theory to (1) examine the overall EA evolution process and (2) identify the mechanisms that affect EA evolution. This thesis adopts critical realism as its underlying philosophical foundation, because the morphogenetic theory is based on critical realism. A qualitative research approach is used due to the explorative nature of this thesis and its congruency with the adopted theory and its underlying philosophical foundations. This thesis contributes to wider IS knowledge by developing a theoretical model that describes and explains how EA evolves. It contributes to the EA domain by identifying multiple generative mechanisms that influence EA evolution. It also identifies and classifies multiple levels of EA evolution. Table 1.2 summarises the research gap, research problem, research questions, research objectives, the study's methodological approach, and its key outcomes and findings.

Table 1.2 Summary of chapter one

Aspect	Summary
Research context	EA evolution due to emerging business and IT trends
Research questions	• How does EA evolve as a result of SOA introduction? • What are the factors that influence EA evolution as a result of SOA introduction?
Gaps to be addressed	• The lack of empirical studies that investigate EA dynamics. • Existing literature suggests the lack of empirical studies that examine EA evolution due to emerging trends and, in particular, the emergence of SOA.
Research objectives	The development of a theoretical model that describes and explains how EA evolves.
Theoretical foundation	The morphogenetic theory (Archer 1995) is adopted to provide an analytical lens and a conceptualisation tool of EA evolution.
Philosophical foundation	Critical realism.
Methodological approach	A qualitative approach, combining explorative interviews and case studies. A critical realist

	methodological framework (iterative stages) is used to guide the conduct, the data collection and analysis of the study.
Key outcomes of research phases	• A-priori model of the literature using the morphogenetic theory. • Large volume of rich empirical data from the explorative interviews which are used to refine the a-priori model. • Large and rich empirical data of two case studies from numerous sources (e.g., interviews, documentations, online reports) to provide insights of EA evolution
Key findings	• A theoretical model that describes EA evolution and explains EA evolution outcomes • Identification of range of influencers (three conditional generative mechanisms and six action-formation mechanisms) • Identification of multiple levels of EA evolution (evolution outcomes)

Chapter 2: Literature Review

2.1 Introduction

This chapter reviews the relevant literature related to enterprise architecture (EA), its evolution and Service-oriented Architecture (SOA). This literature review defines the research context and its boundaries, and develops a standpoint from which to understand EA evolution and specifically SOA's integration into EA. In this respect, it identifies factors and themes that may affect this SOA integration into EA. Moreover, it compares various integration approaches and classifies them.

A limited number of studies were found that discuss SOA's integration into EA. This result compares with Viering et al.'s (2009) finding that most SOA studies do no not relate SOA to EA. In addition, Joachim (2011) concludes that SOA studies are rare in the top IS journals. A few relevant articles were identified through IEEEXplore, SpringerLink, ScienceDirect, EBSCO Host-Business Source Elite, and AIS Electronic Library. Based on the Go Backward and Go Forward searching strategy (Webster & Watson, 2002), the search was extended later using Google scholar.

The chapter progresses as follows. Section 2.2 overviews EA, its development, EA framework structures, and EA maturity models. Section 2.3 introduces EA evolution. Section 2.4 introduces SOA, SOA's diverse perspectives, SOA's perceived benefits, and its implementation scope. The remaining sections compare examples of SOA integration into EA. Section 2.5 covers SOA's integration into the Zachman Framework. Section 2.6 covers SOA's integration into five other widely used EA frameworks, and Section 2.7 presents other studies that suggested approaches of SOA integration into EA. Section 2.8 summarises the chapter.

2.2 Enterprise Architecture

This section introduces EA. It covers EA definitions and The Institute of Electrical and Electronics Engineers (IEEE) standards for architectural descriptions, and introduces EA frameworks that are discussed later in this chapter. It also discusses how EA maturity influences SOA integration into EA.

2.2.1 EA Definition

Architecture is a widely used concept that is used to denote human-made, abstract, and natural things. Architecture is the design of any type of structure. It can be conceptual, physical, real, or virtual (Perko, 2008). More formally, in this thesis, architecture is defined as the

> *"fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution" (ISO/IEC/IEEE 42010, 2011, p. 2).*

Organisations are systems, and also systems of systems that comprise many interrelated elements and relationships. An organisation's architecture is commonly referred to as the enterprise architecture (Lankhorst, 2005). EA is defined varyingly. For example, Lankhorst (2005, p. 3) defines enterprise architecture as: "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organisational structure, business processes, information systems, and infrastructure". Schekkerman (2005, p. 18), on the other hand, defines EA as:

> *a complete expression of the enterprise; a master plan which "acts as a collaboration force" between aspects of business planning such as goals, visions, strategies and governance principles; aspects of business operations such as business terms, organisation structures, processes and data; aspects of automation such as information systems and databases; and the enabling technological infrastructure of the business such as computers, operating systems and networks.*

EA describes and models elements of organisations, and shows how they are organised and connected and how they function as a whole. EA is not a physical artefact in itself; rather, it produces the artefacts (e.g., models) that illustrate an organisation's present and desired future structure (Seppänen, 2008). In general, EA should be organised in a manner that explains an organisation's structure and behaviour. It specifies the elements that constitute the organisation.

Further, EA elements and artefacts need to be applicable for a broad range of organisations and government agencies (Winter & Fischer, 2007). Yet, little consensus exists about EA terminologies, concepts, approaches, and outcomes, which has created confusion in the EA discipline (Luo, 2006; Mykhashchuk, Buckl, Dierl, & Schweda, 2011).

2.2.2 Enterprise Architecture's Development

Enterprise architecture development is a continuous process that involves developing, revising, enforcing, applying, and disseminating results. This process should align with developments in an organisation's external and internal environments, which includes both its strategy and its operational activities (Land, et al., 2009). EA is often complemented by an EA lifecycle methodology that guides the process of developing and maintaining architectures and architectural descriptions (Jung, 2009). EA changes over time to represent the system of interest and provide value for stakeholders. Many organisations struggle, due to complexity, when transitioning from their current to their desired EA as shown in Figure 2.1 (Buckl, Ernst, Matthes, & Schweda, 2009). Creating a roadmap for transitioning EA has to include a multitude of projects that change the architecture. Currently, most EA transformation approaches do not consider architecture roadmaps that present EA evolution over a certain period of time (Buckl, Ernst, Matthes, & Schweda, 2009).

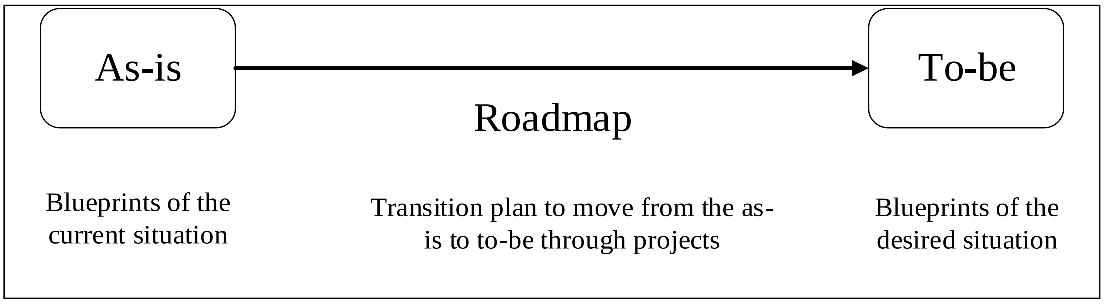


Figure 2.1 EA transformation process

2.2.3 EA Layers

Due to the wide range of relevant elements of EA, EA may cover a large number of business and IT artefacts. Thus, in order to reduce EA's complexity, EA frameworks are often divided into sub-architectures (layers).

EA is usually divided into several sub-architectures, such as business architecture, information architecture, application architecture, and technology architecture (Braun & Winter, 2007; Correia & Silva, 2007). Based on the multi-layer structure, EA can be described as the view that represents all combined artefacts and their relationships throughout all different layers (Braun & Winter, 2007). For example, Winter and Fischer (2007) analyse frequently used frameworks and identify the following EA layers:

- Business architecture: represents an organisation's essential administration from a strategy perspective. Artefacts depicted on this layer typically are value networks, targeted markets, organisational goals, and strategic projects.
- Process architecture: represents an organisation's business processes. Artefacts depicted on this layer are usually business processes, organisational units, internal and external business services, responsibilities, and performance indicators.
- Integration architecture: represents the fundamental administration of information system components. Examples of artefacts depicted on this layer are enterprise services, application clusters, and integration systems.
- Software architecture: represents the essential administration of software artefacts such as software services and data structures.
- IT (or infrastructure) architecture: represents the fundamental administration of hardware and networks.

Other EA frameworks have considered business, information systems (information, applications), and technology to be EA layers. They are widely accepted and used in the enterprise architecture discipline (Joachim, Beimborn, Schlosser, & Weitzel, 2011; Lankhorst, 2004; Pulkkinen, 2006). Furthermore, a widely used EA (Infosys, 2009), the Open Group Architecture Framework (TOGAF), uses a similar structure: business, information systems, and technology (The Open Group, 2009).

2.2.4 EA Objectives

Although a plethora of different EA frameworks have been developed (Berrisford & Lankhorst, 2009; Buckl, et al., 2010; Sowa & Zachman, 1992; Stein, Lauer, & Ivanov, 2008; The Open Group, 2009d; The UK Department of Defence, 2010b; US Department of Defence, 2009), they all have a set of shared objectives.

By depicting fundamental artefacts of an enterprise and their dependencies and by providing different analyses in different settings, EA is critical to organisational management and engineering (Braun & Winter, 2007; Bucher, Fischer, Kurpjuweit, & Winter, 2006). EA defines how information technology is associated with organisational business processes and outcomes, and describes relationships between technical and organisational elements (Weerakkody, Janssen, & Hjort-Madsen, 2007). Thus, EA is considered as a means to address the complexity of contemporary organisations (Seppänen, 2008; Zachman, 1987).

EAs are often used to provide an inclusive descriptive overview of organisations and to govern and direct IT and business decisions (Ekstedt et al., 2004; Foorthuis, Hofman, Brinkkemper, & Bos, 2009). EAs improve the understanding of an organisation's business and information systems landscape and support holistic decision-making (Franke et al., 2009). At a high level of abstraction, EA is a communication method between stakeholders (Chen, Doumeingts, & Vernadat, 2008). It facilitates the communication of an organisation's essential constituents to different stakeholders by allowing different viewpoints and alternative levels of abstraction along the artefact development lifecycle (Buckl, et al., 2010). Further, EAs, as a central repository reflecting organisational elements and relationships, have become essential input for developing required information systems (Khoshnevis, et al., 2009).

EA models are maps with information about an organisation's current (as-is) state and strategies for future (to-be) directions. Therefore, maintaining a high-quality EA model allows organisations to react quickly to new demands and to evaluate potential future directions. In particular, EA models can assist decision-making on issues such as IT-business alignment, IT investment, and IT systems quality assessment (Lindström, Johnson, Johansson, Ekstedt, & Simonsson, 2006). In contrast to many traditional architecture approaches such as IS architecture and software architecture, EA clearly includes pure business artefacts and thus provides a better chance to align IT and business more effectively.

EA can support IT-business alignment, project portfolio planning, business process redesign, quality management, sourcing decisions, and IT service management by using different methods of analysis such as dependency analysis, cost-benefit analysis, complexity analysis, and interface analysis. EA benefits organisations in many ways, such as by aligning business and IT, improving organisational communications and information sharing, and reducing IT complexity (Tamm, Seddon, Shanks, & Reynolds, 2011).

EA has two major functions. The first function is to provide a complete and comprehensible descriptive overview of an organisation to decision makers or stakeholders. Such a function builds the foundations for making high-level management decisions (Ekstedt, et al., 2004; Johnson, Ekstedt, Silva, & Plazaola, 2004). The second function is to provide "a prescriptive framework" that directs and governs the development of an organisation's IT and business domains (Foorthuis, et al., 2009).

2.2.5 IEEE Standard 1471-2000

The IEEE Standard 1471-2000, IEEE Recommended Practice for Architectural Description, introduces and provides a conceptual framework for architectural description. The standard demonstrates terms and concepts concerning the content and the use of architectural descriptions. The objective of this standard is to organise and facilitate the communication of architectures by standardising architectural description elements and practices. Figure 2.2 represents the standard's conceptual framework for architectural description. In this framework, a system has an architecture that can be described by an architectural description. The architectural description can include many views, viewpoints, and models.

Definitions of the conceptual model's key elements are listed below (IEEE, 2000):

- Architectural description: "a collection of products to document an architecture". It includes views, viewpoints, and models.
- Architecture: "the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution".
- Stakeholder: "an individual, team, or organisation (or classes thereof) with interests in, or concerns relative to, a system".
- View: "a representation of a whole system from the perspective of a related set of concerns".
- Viewpoint: "a specification of the conventions for constructing and using a view".

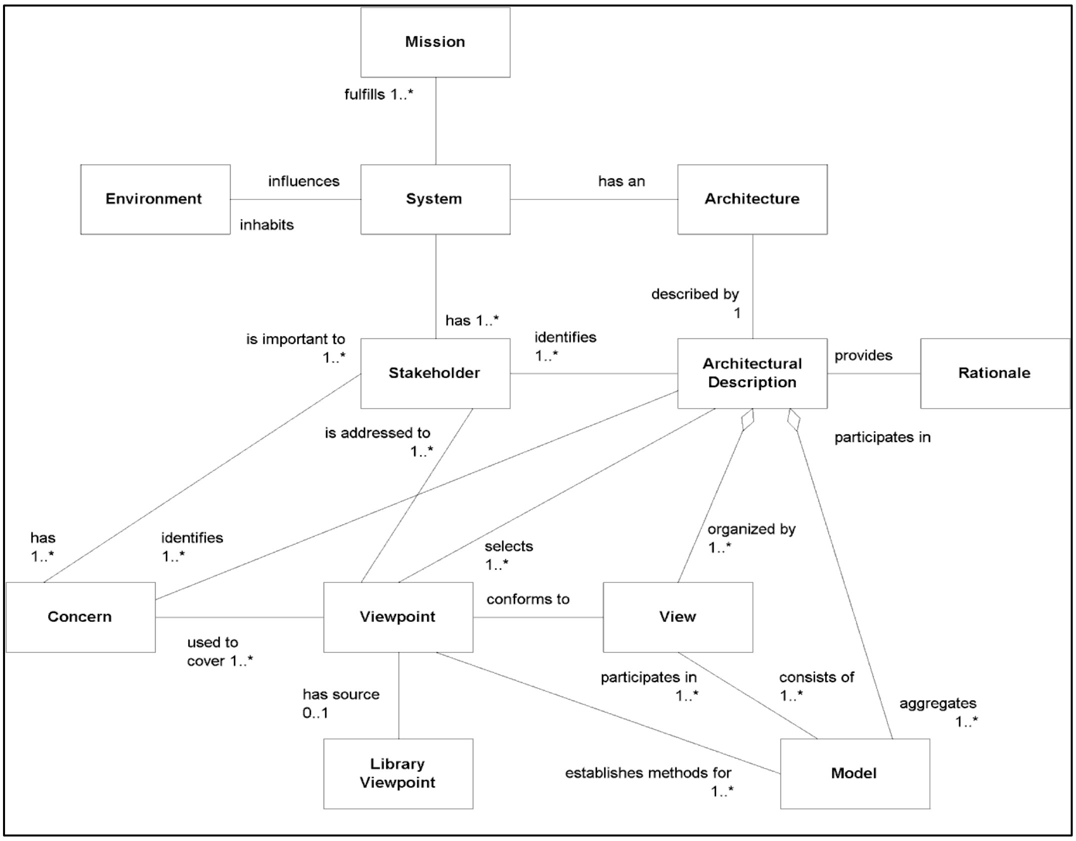


Figure 2.2 Conceptual model for architectural description, IEEE Standard 1471-2000

2.2.6 Enterprise Architecture Frameworks

There are different architectural frameworks in use today. They may have some overlap or address similar concerns. However, some frameworks have been designed to address specific needs or views. These frameworks differ by the stakeholders they address and the issues that concern their "world" (Urbaczewski & Mrdalj, 2006). Goethals (2003), in a comprehensive review of EA frameworks in the literature, classified EA frameworks into classic enterprise architecture frameworks (e.g., the Zachman Framework and the 4+1 View Model) and federated enterprise architecture frameworks (e.g., the Federal Enterprise Architecture Framework, the Department of Defence Architecture Framework, and the Treasury Enterprise Architecture Framework). Urbaczewski and Mrdalj (2006) discuss and compare five EA frameworks: the Zachman Framework, the Federal Enterprise Architecture Framework, the Department of Defence Architecture Framework, the Treasury Enterprise Architecture Framework, and TOGAF. According to a recent survey, TOGAF and the Zachman Framework are the two most widely used frameworks (Varnus & Panaich, 2009). Widely used EA frameworks are discussed in Sections 2.5 and 2.6

2.2.7 Enterprise Architecture Maturity

EA is developed to manage the whole organisational domains: business and IT. However, often, EA's architecture function has tackled only limited parts of organisations, primarily the technology aspects (Turner, Gøtze, & Bernus, 2010). High EA maturity increases EA's value and scope. However, reaching such a level requires planning and effort. In their early stage, organisations normally adopt EA to help them to standardise their technical platforms and infrastructure. Later, organisations may extend their EA to include business architecture (see Figure 2.3).

Research suggests that organisations with mature EA make synergies between EA components and processes to achieve business value (Espinosa, Boh, & DeLone, 2011). Thus, organisations need to incorporate business and IT architectures into their EA scopes in order to realise more value from EA (Perko, 2008). In their study, Schmidt and Buxmann (2011) report that EA was considered an important practice by all interviewees for their organisations. Yet, there was a high variety of EA maturity due to variations in adopted EA methodologies, objectives and EA implementations

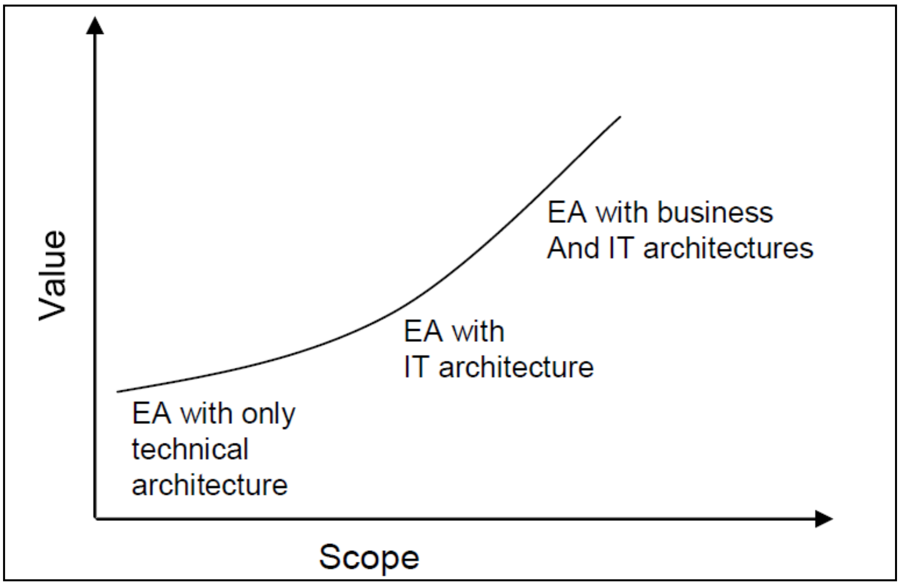


Figure 2.3 EA scope and value (Perko, 2008)

In relation to SOA, an organisation's EA must support SOA development efforts by identifying, classifying, and managing services in a way that is commensurate with the organisation's mission (Brooks, 2009). Antikainen and Pekkola (2009) show that using EA helped business people to gain better awareness of their organisation's architecture and of SOA. O'Brien (2009) highlights the need for business enterprise architecture to drive the SOA initiative in order to deliver an organisation's required set of services. EA should act as a blueprint for the SOA initiative as a starting point for various types of SOA projects. He identifies several factors that might impact an organisation's SOA project, such as the availability of a detailed business enterprise architecture and the skill level of the architects and developers designing and implementing the SOA. In addition, (Kokko, Antikainen, & Systa, 2009) note that, in the organisations they interviewed, the major obstacles for SOA adoption was a low maturity of EA and the absence of business process models.

Several EA maturity frameworks have been proposed and the following paragraphs discuss some well-known EA maturity frameworks.

2.2.7.1 Ross and Beath's (2006) EA Maturity Model

Ross and Beath (2006) developed an EA maturity model that describes how organisations generate value by increasing their EA's maturity level. Ross and Beath's (2006) model states that organisations evolve through four

EA development stages (see Figure 2.4). Each consecutive stage increases the value of IT to the organisation and improves its effectiveness. In a earlier version of the model, Ross (2003) identified four stages of increasing IT architecture maturity. Consequently, Ross and Beath (2006) extended Ross's (2003) previous work to the context of enterprise architecture. Venkatesh et al. (2007) uses the extended model to investigate the architectural evolution of the Veterans' Health Administration.

	Business Silos	Standardised Technology	Optimised Core	Business Modularity
IT Capability	Local IT Applications	Shared Technical Platforms	Organisation-wide processes and data	Plug-and-Play business process modules
Business Objectives	ROI of Local Business Initiatives	Reduced IT Cost	Cost and Quality of Business Operations	Speed to market; strategic agility
Funding Priorities	Individual Applications	Shared Infrastructure Services	Enterprise Applications	Reusable Business Process Components
Key Management Capability	Technology enabled change management	Design and update of standards / funding shared services	Core enterprise process definition and measurement	Management of reusable business processes
Who Defines Applications	Local Business Leaders	IT and Business Unit Leaders	Senior Management and Process Leaders	IT, Business and Industry Leaders
Key IT Governance Issues	Measuring and Communicating Value	Establishing local, regional and global responsibilities	Aligning process priorities with architecture objectives	Defining, sourcing and funding business modules
Strategic Implications	Local / Functional Optimisation	IT Efficiency	Business operational efficiency	Strategic agility

Figure 2.4 EA maturity model (Ross & Beath, 2006)

2.2.7.2 The U.S. Governance Accountability Office EAMMF

The U.S. Governance Accountability Office (GAO) developed the EA management maturity framework (EAMMF) as a benchmarking tool for planning and evaluating enterprise architecture efforts (see Figure 2.5). The last publicly available version (EAMMF v2) includes three interrelated components: (1) seven stages of management maturity, (2) four representations of management attributes that are critical to the success of any program or organisational initiative, and (3) many EA management elements that are at the heart of an EA program (The US GAO, 2010).

Maturity stages are classified into seven stages: (1) creating EA awareness, (2) establishing EA institutional commitment and direction, (3) creating the management foundation for EA development and use, (4) developing initial EA versions, (5) completing and using an initial EA version for targeted results, (6) expanding and evolving the EA and its use for institutional transformation, and (7) continuously improving the EA and its use to achieve corporate optimisation. The four representations of management attributes are: (1) EA management action representation, (2) EA functional area representation, (3) office of management and budget capability area representation, and (4) EA enabler representation. The 59 core elements are the EA structures, activities, practices, and conditions that, when used based on the circumstances of each organisation and the declared purpose of its EA program, can help that organisation to move to progressively higher states of EA maturity and thus increase its chances of achieving EA's value (The US GAO, 2010) .

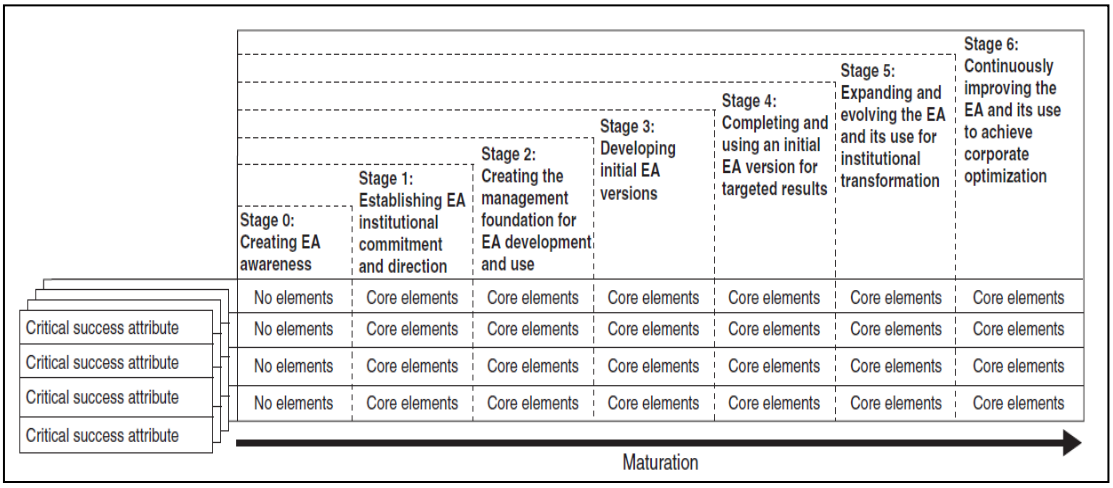


Figure 2.5 EAMMF overview (The US GAO, 2010)

2.2.7.3 The Office of Management and Budget Assessment Framework

The Office of Management and Budget (OMB) developed the Enterprise Architecture Assessment Framework (EAAF) to evaluate the capability and effectiveness of agencies' EA programs. Each criterion consists of five performance levels. Assessment criteria are grouped into three capability areas: completion, use, and results (The US OMB, 2009).

2.2.7.4 Extended Enterprise Architecture Maturity Model (E2AMM)

The Extended Enterprise Architecture Maturity Model (E2AMM) has been developed to provide a model to assess enterprise architecture maturity on five levels: no enterprise architecture (level 0); initial enterprise architecture (level 1); under development (level 2); defined (level 3); managed (level 4); or optimised (level 5) (IFEAD, 2004). This model attempts to prescribe a path for architectural improvements in multiple dimensions (see Table 2.1).

Table 2.1 E2AMM

E2AMM dimensions
• Business & technology strategy alignment
• Enterprise involvement
• Management involvement
• Business units involvement
• Enterprise architecture program office
• Enterprise architecture developments
• Enterprise architecture results
• Strategic governance
• Enterprise program management
• Holistic enterprise architecture
• Enterprise budget & procurement strategy

2.2.7.5 Dynamic Architecture (DyA) Maturity Matrix

DyA was introduced as an architecture maturity model for enterprise architecture. The model differs from the existing maturity models in that it departs from the standard five-stage approach. It identifies 18 factors, called key focus areas, which are applicable to improving architectural practice. Each key focus area has its own number of specific maturity levels (see Table 2.2). The overall maturity of an organisation is determined as a combination of the maturity levels of these key focus areas (Steenbergen, Berg, & Brinkkemper, 2007).

Table 2.2 The DyA maturity matrix

Scale Area	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Development of architecture		A			B			C						
Use of architecture			A			B				C				
Alignment with business		A				B				C				
Alignment with the development process			A				B		C					
Alignment with operations					A			B			C			
Relation to the as-is state					A				B					
Roles and responsibilities				A		B					C			
Coordination of developments							A			B				
Monitoring				A		B		C		D				
Quality management								A		B			C	
Maintenance of the architectural process							A		B		C			
Maintenance of architectural deliverables					A			B					C	
Commitment and motivation		A					B		C					
Architecture roles and training				A		B			C			D		
Use of an architectural method				A						B				C
Consultation			A		B				C					
Architectural tools							A				B			C
Budgeting and planning				A							B		C	

2.2.7.6 National Association of State Chief Information Officers (NASCIO)

The NASCIO Enterprise Architecture Maturity Model provides a path for architectural and procedural improvements in and across organisations. As the architecture matures, effectiveness, predictability, and process controls also improve. At a high level, the components of the model include architecture governance, business architecture, and technology architecture. They are mapped across six stages of maturity that closely conform to the Software Engineering Institute (SEI) capability maturity model. Each level is defined and contains statements that are indicative of an EA program at that level (NASCIO, 2003). Table 2.3 shows the maturity stages and their descriptions.

Table 2.3 NASCIO maturity stages

Maturity level	Description
EA level 0 – No program	No documented architectural framework exists at this level of maturity.
EA Level 1 – Informal program	The base architecture framework and processes are performed informally.
EA Level 2 - Repeatable program	The base architecture and standards have been identified and are being tracked and verified. The program processes are repeatable and reusable templates are starting to be developed. Standards and requirements

	have been agreed on.
EA Level 3 - Well-defined program	The enterprise architecture framework is well defined. Processes are documented across the organisation. Governance committee are defined. Business and IT support and participate in EA activities.
EA Level 4 - Managed program	At this point, performance metrics are collected, analysed, and acted on. The metrics are used to predict performance and provide better understanding of EA processes and capabilities.
EA Level 5 - Continuously improving vital program	The processes are mature; targets have been set for effectiveness and efficiency based on business and technical goals. There are ongoing refinements and improvements based on the understanding of the impact changes have to these processes.

2.2.7.7 The Adopted EA Maturity Model

This section discusses the previously introduced EA maturity models in order to select one for this thesis. Ross and Beath's maturity model differs from the other maturity models in terms of its variables and maturity stages. Both the U.S. GAO EAMMF and the OMB FEAF originated in the U.S. public sector. The OMB FEAF has only three dimensions, and the GAO EAMMF, being rather complex, has extra dimensions that are specifically designed for U.S. government agencies. The DyA matrix model uses a different method from the other maturity models to measure maturity on each dimension. Therefore, these EA models are not adopted in this thesis as standalone models. The E2AAM has multiple variables that shared within other maturity models, and it does not provide a description of how to measure maturity on the given dimensions.

On the other hand, there are significant similarities between these models. In a similar comparison, Lagerstrom, Sommestad, Buschle, and Ekstedt (2011) conclude that there is an overlap between most of EA models in terms of maturity dimensions. They use dimensions that are aligned with the NASCIO maturity model, which the Open Group considers a good example of EA maturity models, and that can be used to assess government and private EA maturity (The Open Group, 2009e). The NASCIO maturity model conforms to the well-known maturity model SEI CMM (NASCIO, 2003) and is widely used (Gosselt, 2012). Thus, this thesis adapts the

NASCIO maturity model guided by the most used EA dimensions (Lagerstrom, et al., 2011; Schmidt & Buxmann, 2011), which Table 2.4 shows. The items of the maturity dimensions are presented in Appendix (B) in the case study protocol.

Table 2.4 Adopted maturity model (based on NASCIO)

Adopted dimensions	Description
EA governance	EA governance assesses the maturity of EA policies, procedures, and guidelines that have to be complied with in order to achieve better outcomes.
EA planning	EA planning dimension is to assess the maturity of the process and activities of developing enterprise architecture.
EA team	This dimension assesses the maturity of EA team.
EA documentation	EA documentation assesses the maturity on EA documentation practices level. It includes the assessment of documentation process, consistency, use of standards and use of EA repository.
EA evaluation	EA evaluation dimension assesses the maturity of EA at the level of maintenance and update of EA artefacts and processes.
EA business support	Business Support evaluates the awareness of EA and the level of business support and engagement with the EA program.

2.3 Enterprise Architecture Evolution

Enterprise architecture captures the fundamental elements of organisation and their relationships to enable organisational analysis and planning. As the elements and their relationships change over time, EA management becomes progressively more complex. Transformations provide the mechanisms through which architectures evolve (Martin, et al., 2009). For example, strategic change in an organisation can lead to evolutionary changes in its enterprise architecture. Further, the merging of two organisations may require the integration of their existing EAs into a new joint EA. Moreover, for an organisation evolving from a traditional to a virtual enterprise the concept of a customer may undergo significant change and the consequence of this change, required in EA, may be overlooked (Shah & Golder, 2011).

Yet, an analysis of existing EA methods shows that the complexity of EA dynamics has not been sufficiently addressed (Saat, et al., 2009) and that the current EA frameworks and models do not consider organisations' dynamic nature (Sousa, et al., 2009). Existent studies on EA planning do not broadly consider dynamic aspects such as the interdependencies, volatilities, and impacts of changes (Saat, et al., 2009). Findings from EA practices study demonstrate that current EA planning processes do not adequately cover EA dynamic aspects (Aier, Gleichauf, Saat, & Winter, 2009).

Most of the previous EA-related studies have focused on EA's static characteristics such as EA frameworks, EA design, and EA modelling (Haki, Legner, & Ahlemann, 2012; Kappelman, McGinnis, Pettite, & Sidorova, 2008; Simon, et al., 2013; Winter, et al., 2010). On the other hand, few studies have addressed EA's dynamic aspects (Aier, et al., 2009; Buckl, Ernst, Matthes, & Schweda, 2009; Lucke, et al., 2010; Saat, et al., 2009). Dynamic EA aspects are challenging and far less explored research area (Lucke, et al., 2010; Simon, et al., 2013). Hardly any empirical work has investigated how organisations migrate from one EA to the next after they implement a new strategy, and very few investigate the necessary changes required for EA (Shah & Golder, 2011). EA's evolution must be traced and more work is needed to understand the practice of architectural evolution (Martin, et al., 2009).

A recent survey of EA literature suggested that more EA work should focus on architectural planning, the integration of EA management into organisations, and deeper examination EA's lifecycle aspects (Simon, et al., 2013). Temporal aspects of EA are also research issues (Aier, et al., 2009; Simon, et al., 2013). The rapidly changing conditions in the technology and business fields are a significant research issue for EA and, thus, architects have to deal with organisational dynamics and constraints (Lucke, et al., 2010). While there is some research on aspects of EA changes (the temporal dimension), it focuses on applications or processes landscapes changes only, and does not satisfy EA's holistic scope requirements (Saat, et al., 2009). Moreover, it does not consider the impact of emerging trends that may require EA evolution, such as SOA (Banerjee & Aziz, 2007; Infosys, 2009; Postina, et al., 2010; Varnus & Panaich, 2009).

Managing EA evolution is a key challenge for contemporary organisations. Present approaches to manage this challenge concentrate on EA plans, which represent proposed future states of the architecture. Nonetheless, these plans disregard the role of the information technology (IT) project, which actually transforms an organisation's current EA to its planned EA (Buckl, Ernst, Matthes, & Schweda, 2009). Managing EA evolution requires understanding the motivations and the mechanisms behind it (Martin, et al., 2009).

A major problem in enterprise architecture management is to maintain EA models and to keep them up-to-date and of satisfactory quality (Farwick, et al., 2012). Developing an EA is not a single event that produces static descriptions of an organisation—such descriptions would only subsequently hinder change. Rather, EA has to evolve in parallel with the evolution of the organisation and its strategy (Shah & Golder, 2011). In order to develop architectures and models that reflect organisations' systems nature, the frameworks, methodologies, and terminology used in developing EA need to consider that organisations are adaptive systems of systems (Sampaio, 2010; Sousa, et al., 2009). When organisational elements change, their architectural descriptions need to co-evolve in order to sustain the system's relevance and overall value (Harmon, 2005).

EA architectural descriptions (meta) and architectural representation (content) all are evolvable; but, when they are evolved, they are done so for different purposes. EA needs to match organisational changes on both representational (non-meta) and architectural descriptions (meta) levels. Architectural descriptions are the vehicle for building architectural representations (Martin, et al., 2009). Figure 2.6 differentiates the representational from the architectural descriptions level. The abstract (meta-model) represents the architectural descriptions that are used to guide the representational level design.

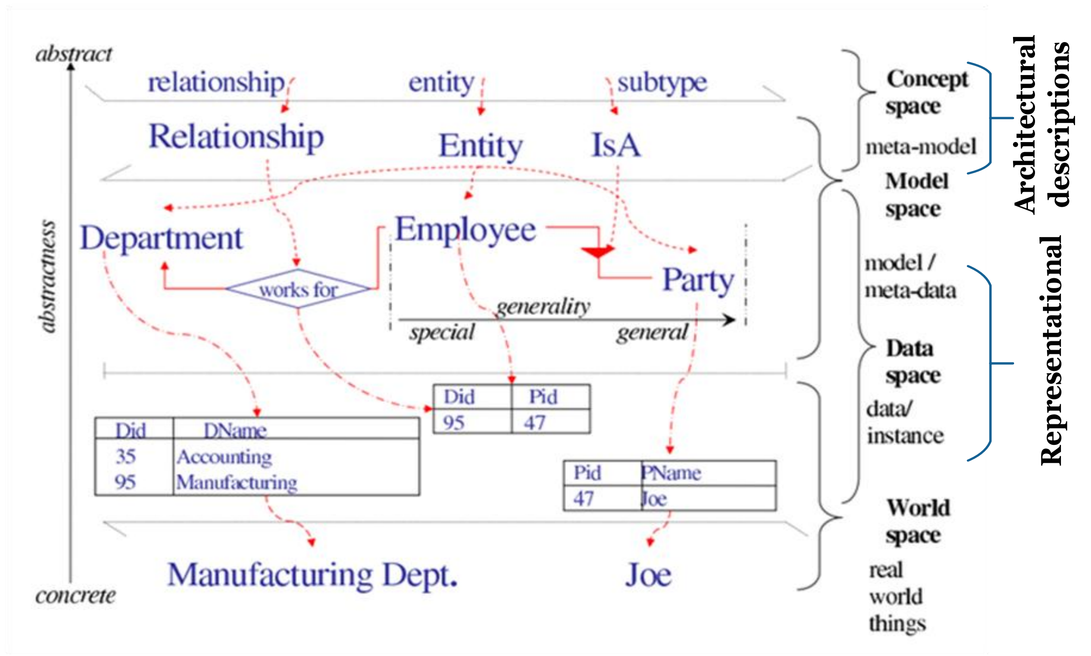


Figure 2.6 Representational vs. architectural descriptions ((Martin, et al., 2009)

Representational level evolution entails keeping EA models up-to-date with instance changes such as changes to business processes or applications data (Martin, et al., 2009; Shah & Golder, 2011). EA's architectural descriptions evolution concerns EA's meta-model. It includes managing the elements, concepts, or properties of a system, their relationships, and the principles of system design (Martin, et al., 2009).

Meta and non-meta aspects of the target architecture need to be distinguished during EA evolution in order to align transformations and to build artefacts that appropriately represent the system of interest and afford value for stakeholders (Martin, et al., 2009). The EA literature contains few studies that provide guidance about EA's representational level evolution. For example, (Buckl, Ernst, Matthes, & Schweda, 2009) propose an information model for modelling EA's transformation process, which is concerned with the managed evolution of the application landscape. (Farwick, et al., 2012) discuss events that may cause changes to existing EA elements due to projects implementations. Yet, most of existing EA planning approaches do not clearly consider external influences on the planning process or changing conditions in an organisation's environment (Saat, et al., 2009).

Thus, organisations need to consider both representational and architectural descriptions when transforming their EA in order to keep it relevant and updated (Buckl, Ernst, Matthes, & Schweda, 2009).

EA elements and their relationships do not evolve accidentally. Rather, they are changed by actions taken by actors, such as projects, initiatives (Farwick, et al., 2012), and new business and IT trends (e.g., SOA) (Banerjee & Aziz, 2007; Infosys, 2009; Postina, et al., 2010; Varnus & Panaich, 2009). Current business and IT trends such as SOA and cloud computing are challenges for EA (McKendrick, 2010). The emergence of SOA requires more than the representational maintenance of EA instances, and necessitates changes to underlying EA architectural descriptions (Correia & Silva, 2007; Khoshnevis, et al., 2009).

This thesis defines SOA as "an architectural style that supports service orientation, and service orientation is a way of thinking in terms of services and service-based development and the outcomes of services" (The Open Group, 2010).

SOA presents a new way of thinking to organisations which requires existing EA frameworks to evolve (Knippel & Skytte, 2007). EA and SOA share many similarities and even overlap in terms of concepts, activities, and outcomes. They deal with similar architectural domains, they intend to align business and IT, and they both require similar strategies and planning activities (Ibrahim & Long, 2007). They both have been abstracted away from technology to address information systems' complexity and business changes. They share the same goals of creating agile and cost-effective organisations. They also promote improved interoperability and better alignment of business and IT (Seppänen, 2008). Thus, SOA and EA should be seen as complementary to each other rather than as alternatives (Knippel & Skytte, 2007; Seppänen, 2008) in order to increase their values.

Due to the rising growth of SOA adoption, consistent EA management has become fundamental. That is SOA's business-oriented design requires the alignment of IS architecture with business architecture in order to effortlessly align information system design with business requirements (Legner & Heutschi, 2007). Correia and Silva (2007) argue that the service concept has become as significant as the other core concepts in EA, such as data, functions, and locations. Due to the lack of an explicit representation of services in existing EA frameworks, Correia and Silva (2007) emphasise the need to capture services on EA frameworks because services play a functional and structural role in organisations. EA frameworks have to evolve to be able to match organisations' actual representation, and therefore the service concept has to be considered as an important element of current EAs. Correia and Silva (2007) stress the importance of an integrated and cohesive vision of services in enterprise architecture in order to increase organisations' agility.

Enterprise architects have to adapt their methodologies and concepts in order to manage the complexity of service architecture (Postina, et al., 2010). SOA requires an intensive enterprise re-engineering effort, which affects the different EA layers such as process, applications and, infrastructure. Grigoriu (2007) argues that SOA implementation does not achieve its objectives outside the context of EA development because SOA relies on EA "as-is" models and artefacts. EA is needed to plan the journey, and an EA team should embrace SOA to increase SOA's visibility and impact (Paras, DeBoever, & Westbrock, 2007). Organisations need an EA with supportive methods and tools in order to implement an architecture based on SOA, and to take advantage of the new capabilities that SOA offers (Perko, 2008). Service-orientation descriptions can be employed for the semantic integration of both the dynamic and static aspects of EA frameworks (Gustas, 2007).

2.4 **Service-oriented Architecture**

This section introduces Service-oriented Architecture (SOA). It covers SOA definitions, SOA reference architecture, and services. It also covers three aspects that are found in the literature to impact SOA's integration into EA: the view of SOA, SOA's perceived benefits, and SOA's scope.

SOA has recently gained popularity in both academia and practice. It is based on recognised concepts such as loose coupling, composition and coordination of building blocks, and complexity reduction (Schroth, 2007). Academics and practitioners have defined the term varyingly. However, in general, SOA is defined from two perspectives: a narrow technical and a broader business/managerial view. Although most SOA's definitions are predominantly technical, recent publications have taken a broader, business perspective (Lee, Shim, & Kim, 2010). From a technical perspective, SOA is a software architecture approach in which the basic design and development components are services (Kumar, Dakshinamoorthy, & Krishnan, 2007) .

On the other hand, from a broader business perspective, SOA is a paradigm for structuring an organisation's business in the form of services, which accordingly drive the IT architecture (Engels et al., 2008). It provides a new way for organisational design that covers both an organisation's business and technical aspects. The SOA concept includes the understanding of business capabilities as services (e.g., payment, fraud detection) down to the technical implementation of encapsulated software capabilities in terms of Web services (Lee, et al., 2010; Stein, et al., 2008).

Moreover, SOA provides a framework to assist communication and interaction between services. Services are advertised by service providers with related service-level agreements in service registries to be accessed and utilised by consumers (Luthria & Rabhi, 2009). Thus, there are three key players in SOA: service providers, service consumers, and the agencies that help consumers find services (Erl, 2005; Luthria & Rabhi, 2009; Papazoglou, 2003), which Figure 2.7 shows.

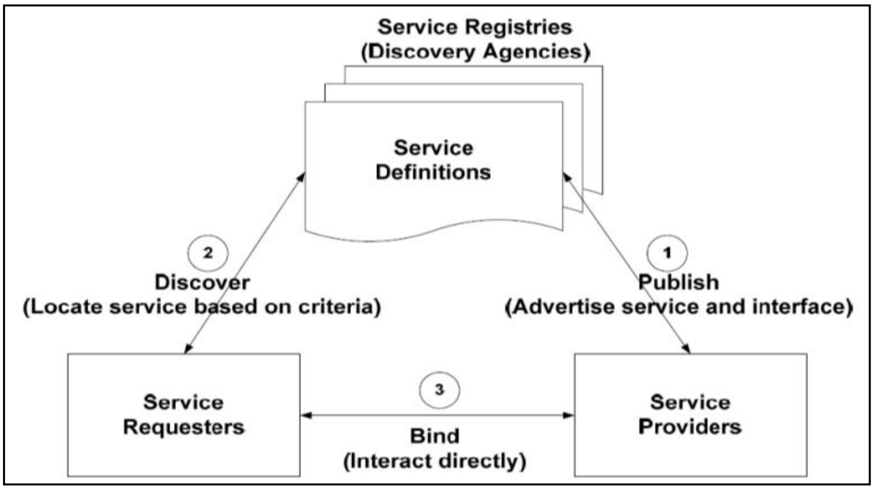


Figure 2.7 Service-oriented Architecture (Luthria & Rabhi, 2009)

Papazoglou (2003) claims that traditional SOA does not cover the overarching issues such as service management, security, coordination, and other concerns that apply to service architecture components. Therefore, the concept of Extended Service-oriented Architecture (ESOA) was introduced to address such concerns. ESOA is illustrated as a pyramid of architectural layers comprising the basic service layer at the bottom, the service composition layer in the middle, and the service management layer at the top. Services from the basic service layer are composed to form a particular composite service. The composition layer encompasses the required functionality for aggregating multiple services such as monitoring and conformance, and non-functional attributes such as security, reliability, and performance. At the top of the pyramid, ESOA provides the necessary operation management functionalities for system integrity and market management functionalities to support marketplace functions (Luthria & Rabhi, 2009; Papazoglou, 2003).

SOA's technical principles are open standards for interoperability, loose coupling, reuse of services, and dynamic orchestration of services. When adopting SOA, organisations hope to leverage existing systems by abstracting the functionalities of applications into services that can be rapidly orchestrated into new business solutions (Chen, et al., 2010). Further, loose coupling of services, modularity, and services reuse are attractive; these factors signify the flexibility to develop, upgrade, replace, or substitute services without affecting an organisation's operation, and the flexibility to effortlessly change a service's suppliers, which leads to cost savings. In addition, interoperability via open standards improves the integration of both internal and external resources and, consequently, leads to more potential cost savings. Therefore, SOA's ability to dynamically configure both internal and external resources is anticipated to improve organisations' competitive advantage in service innovation (Chen, et al., 2010).

In addition, SOA is claimed to be the most capable approach for IT-business alignment, which is an enormous issue that has been a top concern of CIOs for the last two decades (Chen, et al., 2010). Services are becoming fundamental building blocks of contemporary enterprise architectures. EAs' characteristics are changing because the focus is shifting from applications toward services. Consequently, new challenges are arising, such as the level of granularity, the formalisation of interface descriptions, and an increasing number of service interdependencies (Postina, et al., 2010).

2.4.1 SOA Reference Architecture

SOA reference architecture's objective is to offer a guideline for establishing and evaluating SOA's architecture. In addition, it provides insights for integrating SOA's fundamental components into SOA layers (Arsanjani, Zhang, Ellis, Allam, & Channabasavaiah, 2007; The Open Group, 2009f). There are many different SOA reference architectures. Among them is a well-known reference architecture that was developed by SOA experts at IBM and adopted by The Open Group (see Figure 2.8). This reference architecture is used as an enabler to achieve SOA's value propositions.

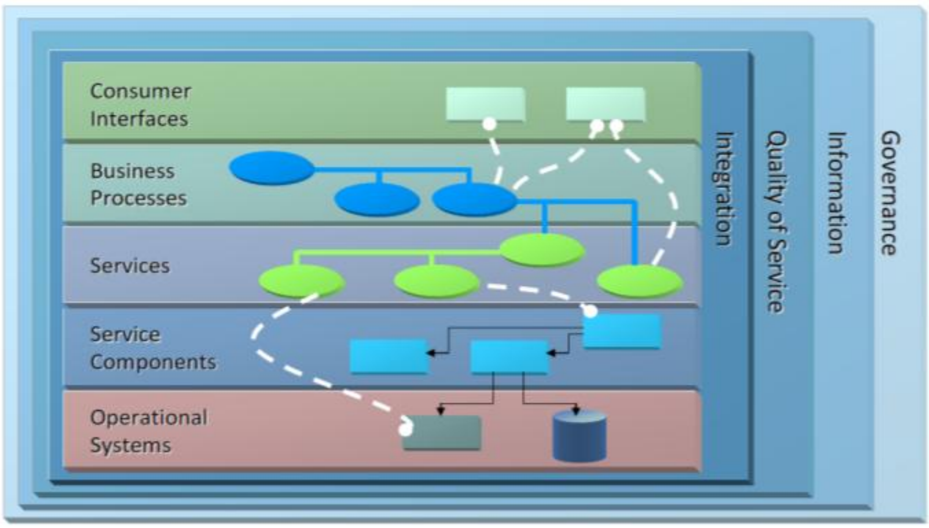


Figure 2.8 SOA reference architecture (The Open Group, 2009f)

First, the operational systems layer captures existing and new infrastructure needed to support SOA. It includes the required infrastructure to run SOA, physical and operational systems components, application assets, infrastructure services, and other composed or orchestrated services. Second, the service component layer contains software components to implement or realise services. It links the service contract to its implementation in the first layer. Third, the service layer, which contains all SOA services, includes the service description, runtime contact description, and service dependencies. Figure 2.9 further elaborates on the service layer. It represents a middleware view and classification of services in the SOA reference architecture. Fourth, the business process layer includes the service composition and orchestration. Finally, the consumer layer is responsible for the provision of capabilities to end users through channels and portals (Arsanjani, et al., 2007; The Open Group, 2009f).

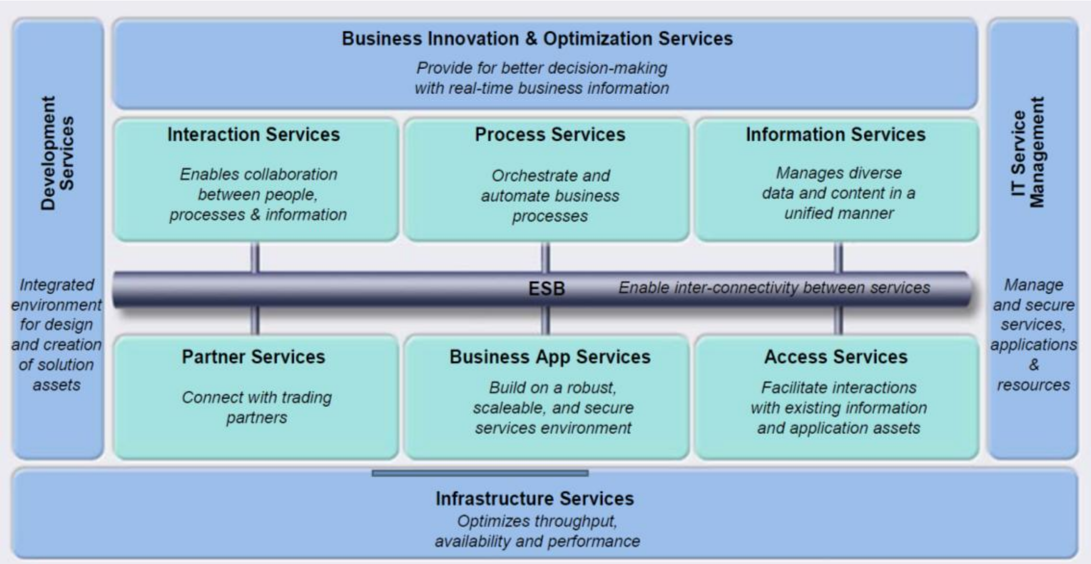


Figure 2.9 Middleware view of the SOA reference architecture (The Open Group, 2009f)

The OASIS SOA Reference Model, which was introduced by the public standardisation body OASIS (2006), describes core SOA elements such as service, contract, policy, service function, service provider, service consumer, service description, service interface, and the relationships between these elements. This model is used to understand the significant elements in a service-oriented environment and to develop consistent standards and specifications that support the environment.

Further, Everware-CBDI Inc. has introduced CBDI Service Architecture and Engineering™ (CBDI-SAE™), which is an SOA meta-model that includes a taxonomy of all terms used (Everware-CBDI Inc, 2009). It provides elements for describing a business-oriented SOA independent of technology and services implementation. However, the CBDI-SAE™ has not been integrated with any existing enterprise architecture framework (Stein, et al., 2008). Nonetheless, the meta-model has a first-class service element that executives relate to very well, and contains some elements logically associated with it such as service interface, service dependency, service domain, participant, and service classification. Service specification, which is used in the software realm, includes the elements that describe a service, such as service definition, service operation, service level agreement (SLA), versioning, service state, and policy.

2.4.2 Services

A service is "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description" (OASIS, 2006). Further, a service is a business function implemented in software that has a formal advertised interface. Services embody full business functions and are designed to be reused and involved in transactions at the application and organisation and across organisational levels (Papazoglou, 2003).

From a business perspective, a service is defined as an interaction between a provider and a client that creates and involves value (Aier & Gleichauf, 2009). Business participants may perceive a service as a unit of transaction described by a contract and fulfilled by the infrastructure. The service's semantics and presumptions are expressed from business experience that determines the perspective (Perrey & Lycett, 2003). However, from an IT perspective, a service is defined as a loosely coupled, discoverable component that has a published interface (Chen, 2008; Perrey & Lycett, 2003). Aier and Gleichauf (2009) state that services are mainly discussed from a technical viewpoint in SOA literature. Although no agreement has been achieved on the definition or conceptual boundaries of the service notion (Aier & Gleichauf, 2009), service-orientation is embraced by both the IT and business communities (Chen, 2008). The business and IT perspectives are not contradictory if the infrastructure and technical services are structured to provide and support business services.

Nurcan and Schmidt (2009) argue that services have become favourable modules in enterprise architecture. Services are used to define the interaction between business and IT. The term is used to describe the interaction elements and the possible aggregations.

Services are classified differently in the literature. For example, Jung (2009) presents a service taxonomy (see Figure 2.10), which includes process services, business services, application services, and infrastructure services. A business service represents business logic (Jung, 2009) and it is a self-contained, independent unit (Banerjee & Aziz, 2007). On the other hand, an application service represents a specific technical functionality and provides reusable technical functions. An infrastructure service provides non-business functionality (Jung, 2009). Further, a process service is coarse-grained and composed of other services (Jung, 2009). Although business processes are not services in their own right, it might sometimes be justifiable to provide a service interface for a business process; for example, to make a process available to other business units inside the organisation (Schulte, Kadner, Repp, & Steinmetz, 2009).

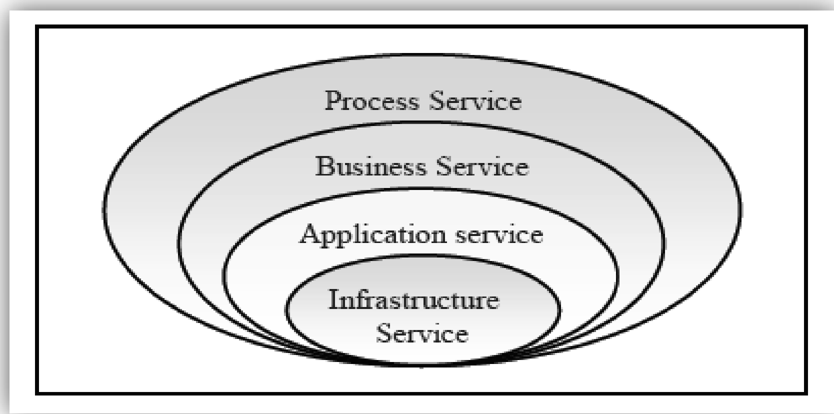


Figure 2.10 Service taxonomy (Jung, 2009)

In addition, Nurcan and Schmidt (2009) suggest that services are an essential part of a service system and are divided into different types (e.g., business services, software services, and technology services). A business service is a service that directly supports business processes. An application service is an encapsulated unit of software that has a published interface. A technology service is a hardware-related service such as storage and infrastructure services. Schulte, et al. (2009) describe different levels of services still:

- Technical services, which are on the infrastructure level and are highly reusable.

- Composite services, which unite functionalities of two or more technical services and may represent direct business value. In addition, they might be a sub-process of a business process; therefore, it is important that these services are effortlessly discovered.
- Business processes, which can be complex and comprise composite or technical services. Although business processes are not services on their own, it might be sometimes justifiable to offer a service interface for a certain business process.
- Public service, which is a service that is made available to consumers outside an organisation's boundary. They could be technical, composite, or even a complete business process.

2.4.3 View of SOA

Understanding and perception of IT phenomena are widely discussed in IS literature. For example, Reich and Benbasat (1996) investigate the understanding of business and IT objectives by managers in organisations. Salmans, Kappelman, and Pavur (2009) investigate the perception of EA by IT professionals. Holland and Light (2001) measure the strategic use of IT, which is concerned with the measurement of how IT is perceived throughout organisations, and how important the IT function is in businesses.

There are many SOA definitions that do not usually view SOA in the same way. Indeed, standardisation organisations, academics, and leading vendors all define the term in different ways (Ren & Lyytinen, 2008). For example, IBM defines SOA as "a business-centric IT architectural approach that supports integrating business as a linked, repeatable business task, or service" (Ren & Lyytinen, 2008). OASIS (2006) defines SOA as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains". Papazoglou (2003) defines SOA as a way to reorganise a collection of formerly siloed software applications into an interlinked set of services. Each service can be accessed through a standard interface using messaging protocols. Engels et al. (2008) define SOA as a paradigm for structuring an organisation's business in the form of services that accordingly steer the IT architecture. Noran and Bernus (2009) believe that SOA is an architectural style that emphasises service concept and service consumers as a foundation to structure the functionality of an entire business.

This list is by no means exhaustive. However, they are included here to illustrate the diversity in how SOA is defined, which might influence the way it is perceived, and consequently the way in which it is integrated into EA. Viering, Legner and Ahlemann (2009) highlight the need for a clear, generic SOA definition accompanied by typologies and taxonomies that distinguish SOA and service designs.

Erl (2005) argues that SOA appears to be a confusing term. There is no shared or common understanding of SOA concepts between technology staff and managers (Gulledge & Deller, 2009). Viering et al.'s (2009) findings, which indicate that understanding of SOA is still immature and under discussion, support this claim. Luthria and Rabhi (2009) state that most organisations adopting SOA do not understand the business potential of SOA and thus focus on technical implementation issues instead of broader business considerations. Becker, Buxmann, and Widjaja (2009) found that a clear understanding of SOA is important to achieve SOA benefits. Developing an organisation-wide perception of SOA is a success factor of SOA implementation. SOA implementation success can be achieved through (1) obtaining a shared understanding of organisation-wide SOA including management and IT as well as (2) viewing SOA as a business paradigm and not a technology (Lee, et al., 2010).

Initially, SOA has been focusing on the design of one domain of EA, the application architecture. SOA is a software architecture where the basic components of design and development are services (Kumar, et al., 2007). A technical view of SOA would limit SOA capabilities and ignore the value of much broader service and SOA concepts. Viering et al. (2009) found that SOA is seen as a pure IT approach and an architectural style that uses services as key artefacts. They found that, even when viewing SOA as an architectural style, there are diverse and different opinions about SOA design principles and characteristics that influence SOA implementation.

Lately, SOA has evolved to represent other domains of EA such as the business architecture. SOA now defines services that represent business architectures (Kistasamy, van der Merwe, & De La Harpe, 2010), which extends SOA beyond technical architecture. Recent publications have taken a broader, business-based viewpoint of the concept (Lee, et al., 2010). Organisations in different industries are eagerly pursuing SOA not just as an architectural style but also as a business strategy (Chen, et al., 2010; Shan & Hua, 2006). SOA is described as a business paradigm that addresses and integrates both business and IT (Joachim, Beimborn, Hoberg, & Schlosser, 2009). From a broader business perspective, Engels et al. (2008) define SOA as a paradigm for structuring organisations in terms of services, which accordingly drives IT architecture. A business-centric SOA maps business functions to technical applications and infrastructure in order to facilitate the automation of business rules and to align business and IT (Luthria & Rabhi, 2009; Perrey & Lycett, 2003). Hirschheim et al. (2010) further emphasise that SOA drives the structure and design of business architecture by identifying higher-level services to steer the definition and development of lower-level services. They conclude that several organisations have started to adopt a pure service-oriented perspective with a vision of becoming a service-oriented enterprise.

Hirschheim et al. (2010) classify the view of SOA, one dimension of SOA maturity, into five maturity stages: fine-grained service components, emerged software architecture, business process support, enterprise service architecture, and adaptive architecture. They conclude that how organisations view SOA is a critical issue for its implementation. They found that organisations perceive SOA as a technology, and therefore most SOA implementations have focused on IT architecture rather than using SOA as a transformational strategic tool. They also argue that many organisations will go through an IT-driven SOA adoption before evolving to a higher maturity stage with a greater business orientation.

This thesis adopts the classification of the view of SOA that are described in two studies (see (Hirschheim, et al., 2010; Welke, Hirschheim, &

Schwarz, 2011). They are used in Chapter 4 to develop the a prior model of this thesis. These levels are shown in Table 2.5.

Table 2.5 Classification of the view of SOA

View of SOA	Description
Fine-grained software components	• Finer-grained software components, largely using web services and wrapping fine-grained existing functionalities • Loosely defined services
Emerged software architecture	• Organisations define and create more and finer-grained services • They employ common service definitions and some form of registry to manage service policies • Services are still defined relative to internal IT needs • Use of enterprise service bus.
Support of business processes	• Firms move beyond the simple, IT-focused management of services toward actual definitions of new services driven by business requirements • Service definition is now directly tied to business requirements capture • Business service definition methods become more important.
Enterprise service architecture	• Redesign of business processes to achieve organisation’s agility through services • Define services in terms of business needs in advance of their use in processes.
Adaptive architecture	• Firms define, develop, and implement business processes that are themselves services. This leads to the creation, in conjunction with senior managers, of truly adaptive enterprise service architecture. • The process of optimisation moves outside the firm along value-chain lines that extend to upstream suppliers and downstream clients

2.4.4 SOA Perceived Benefits

This section introduces SOA perceived benefits as a factor that influences SOA implementation and thus its relation to EA. The value of introducing a SOA for an organisation has been discussed increasingly in past years. Perceived benefits have been discussed widely and have been considered to have an effect on organisations’ intent to adopt. Chwelos, Benbasat, and Dexter (2001) acknowledge that some attributes of a particular technology, such as perceived benefits, will vary across organisations. Parr &

Shanks (2000) note that ERP implementations vary significantly in motivation, and that this variance affect the scope, design, and approach of ERP implementations.

Luthria and Rabhi (2009) argue that organisations adopt SOA based on a certain perceived value. Joachim et al. (2009) state that perceived benefits is a factor that influences SOA adoption. Perko (2008) argues that SOA offers different types of benefits on different levels. He found that organisations adopt SOA for different purposes such as strategy execution, business processes integration and improvement, and IT standardisation.

SOA, as a technical architecture, supports the easier integration of application and business processes. Further, Welke et al. (2011) found that using SOA as an IT design principle leads to many opportunities at the technical level. It affords organisations the ability to reuse IT assets, which contributes to reducing IT development costs and decreasing development time. SOA's potential to reduce implementation and maintenance costs drives SOA adoption from an IT perspective. Through the decomposition of existing applications, the use of an enterprise service bus (ESB) to decouple services, the reuse of services, and the elimination of point-to-point connections, SOA leads to decreases in implementation and maintenance costs (Joachim, et al., 2009).

On the other hand, moving toward a service-oriented organisation is a major reason to adopt SOA from a business point of view. It improves IT flexibility, which consequently improves organisations' responses to environmental changes (Joachim, et al., 2009).

Findings from multiple case studies suggest that organisations seem to choose focused areas for SOA adoption. For example, SOA can be adopted to standardise integration infrastructure, to decouple application domains, and/or to achieve flexible business process integration. Each focus area is distinguished by a set of related benefits and objectives from the organisation's perspective. Moreover, the focus area has considerable implications for the chosen architectural principles and SOA implementation measures (Legner & Heutschi, 2007). Further, Kohlmann et al. (2010) argue that the design of SOA building blocks differs depending on the SOA implementation goal.

Lee et al. (2010) note that setting clear SOA goals based on business value is a critical success factor in SOA implementation. Further, Welke et al. (2011) classify SOA drivers into IT-related drivers and enterprise-related drivers. The IT drivers are infrastructure efficiency, reuse, and application/data composition and integration. The enterprise drivers are business analytics and processes, organisational flexibility and agility, and enterprise transformation. They even classify the drivers into five maturity stages: promise of reuse, standardisation of data and resources, business process redesign, agility and flexibility, and autonomic systems. The primary difference among these SOA stages is the degree to which they relate to IT needs or higher-level business concerns.

Yoon and Carter (2007), in their review of five SOA cases, found that both business and IT benefits drove SOA implementation. A business benefit includes business agility improvement, cost reduction, and accurate data. IT benefits incorporate IT scalability improvement, efficient application development, and the reduction of IT maintenance costs (see Table 2.6).

Table 2.6 SOA benefits (Yoon and Carter, 2007)

Business benefits	IT benefits
Improving business agility	Improving IT scalability
Lowering costs	Efficient application development
Getting timely accurate data	Decreasing IT maintenance costs
Improving customer service	Integrating systems or applications
Increasing business efficiency	Providing timely accurate data

Additionally, Mueller, Viering, Ahlemann, and Riempp (2007) have developed an SOA economic potential model, which illustrates the relationships between SOA's characteristics (such as open standards, modularity and loose coupling) and SOA benefits. They found that SOA's benefits are currently mainly driven by operational and IT infrastructural improvements. Few organisations have also realised strategic benefits from SOA; for example, the integration of business partners through SOA implementations. Table 2.7 classifies SOA benefits.

Table 2.7 SOA benefits (Mueller, Viering, Ahlemann, & Riempp (2007)

SOA impact layers	SOA benefits
IT Infrastructure benefits	Better business/IS alignment
	Better assets utilization
	Reduce redundancy
	Reduce maintenance and operations costs
Operational benefits	Reduce maintenance and operations costs
	Increased availability of information
	Increased customer satisfaction
	Business process improvement
Strategic benefits	Improved inter-organisational coordination and communication
	Individualisation of services and products
	Reduced time to market

Further, Becker et al. (2009) investigate SOA's benefits. They situate the benefits they found into three separate levels: (see also Table 2.8).

- IT-level benefits: reuse, facilitation of software development, IT-landscape consolidation, and management of IT complexity.
- Process-level benefits: process optimisation, improved information quality and availability, and simplified third party integration.
- Strategy-level benefits: agility, strengthening B/IT alignment, and the enablement of new functionality and business models.

Table 2.8 SOA benefits (Becker et al., 2009)

SOA impact layers	SOA benefits
IT level	Reuse
	Facilitation of software development
	IT-landscape consolidation (harmonisation)
	Management of IT complexity
	Facilitation of maintenance
	Risk reduction
	Software (vender) independence
	Improved project management
Process level	Process optimisation
	Improved information quality and availability
	Simplified third party integration
Strategy level	Agility (flexibility and speed for business)
	Strengthening of IT/business alignment
	Enablement of new functionality

As such, we can conclude that SOA offers diverse benefits that may affect SOA adoption, and that these benefits can be classified into different levels (e.g., business and IT benefits; or strategic, process, and IT benefits). This thesis adopts and combines Becker et al.'s (2009) and Mueller, Viering, Ahlemann, and Riempp's (2007) classifications (see Table 2.9).

Table 2.9 Combined and adapted SOA benefits

SOA impact layers	SOA benefits
IT level	Reuse
	Facilitation of software development
	Improved IT integration
	Reduce complexity
	Improved project management
	Better assets utilisation
Process (operational) Level	Increased availability of Information
	Reduce maintenance costs
	Increased customer satisfaction
	Business process improvement
Strategy level	Agility
	Business-IT alignment
	Enablement of new functionality
	Improve communication
	Reduce time to market

2.4.5 SOA Scope

This section introduces the third factor that may have an influence on SOA implementation and thus SOA's integration into EA. Campbell and Mohun (2007) present three different approaches for SOA adoption: project-by-project, portfolio, and enterprise level (see Figure 2.11). Organisations usually adopt the project-by-project based approach when there is no long-term vision and SOA benefits are mainly at the project level. Projects are done separately for service-oriented development and integration purposes. In the portfolio approach, organisations assess their portfolio projects and invest in potential projects for SOA to achieve benefits from SOA at the project and portfolio levels, and to prepare the organisation for SOA adoption at the enterprise level. In the enterprise-level approach, organisations focus on business processes. At this level, the organisation has long-term goals to transform itself into a service-oriented organisation. This adoption approach has the potential to fully realise the maximum benefits of SOA.

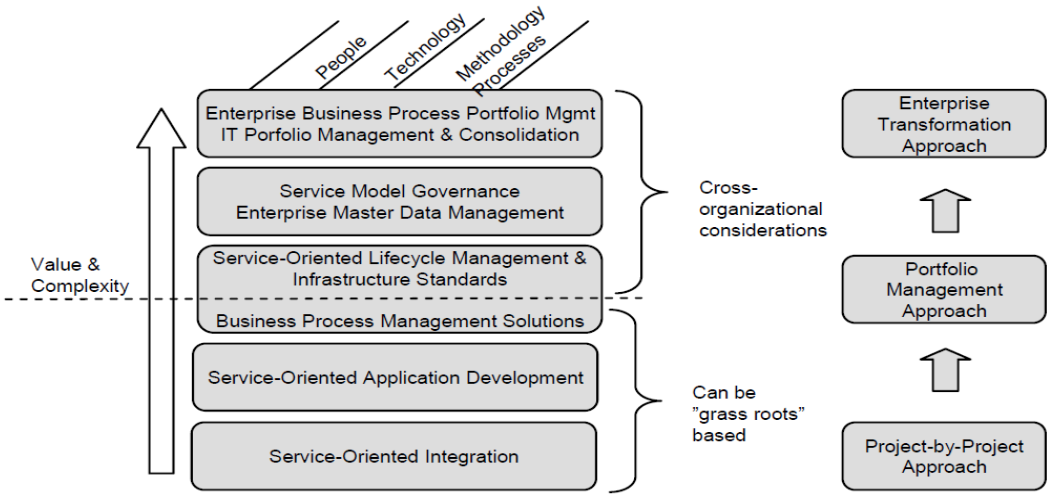


Figure 2.11 Value and scope of SOA adoption (Campbell and Mohun, 2007)

Findings from multiple case studies suggest that organisations seem to choose focused areas for SOA adoption. For example, SOA can be adopted to standardise integration infrastructure, to decouple application domains, and/or to achieve flexible business process integration. Each focus area is distinguished by a set of related benefits and objectives. Moreover, the focus area has considerable implications on the chosen architectural principles and SOA implementation measures (Legner & Heutschi, 2007).

O'Brien (2009) claims that there are different types of SOA projects, and organisations may undertake one or more projects. Each project type may require different methods and tools to determine the project's scope and activities. Further, for every project type, there are significant technical and organisational dimensions that need to be considered. Although SOA scope determination is a critical task, there is a very little published work about estimating the scope of SOA implementation. An understanding of the domain (which includes things such as system complexity, service complexity, process complexity of new services needed, enabling technology, applicable standards, and number of data elements) is required in order to understand what needs to be done and to estimate SOA's scope (O'Brien, 2009).

Shah and Kalin (2007) identify various approaches (models) that organisations have adopted to migrate SOA. They have come across ad hoc and program-based (organic and strategic) approaches. Ad hoc SOA adoption is the project-level adoption of service-oriented technologies on a tactical basis or for a specific need. There is no plan or central coordination. On the other hand, program-based adoption controls SOA's evolution through an overarching organisational strategy and goals. This model offers a holistic view of SOA and addressing organisational business and technology dimensions. The program-based model is further divided into organic and strategic. They differ in the way they are initiated and the pace of adoption.

Hassanzadeh and Namdarian (2010) argue that SOA adoption processes have four levels: intra-department, inter-department, inter-business, and enterprise. At the intra-department level of adoption, smaller SOA rollouts, proof of concept projects, and integration projects are undertaken. At the inter-department level, departments in a business unit are SOA enabled and use services to interact with each other. At the inter-business level, SOA is enabled at the business units' level and the interactions across business units are through services. At the enterprise level, a highly evolved stage of SOA adoption is reached where the whole organisation is designed in terms of services. Kokko et al. (2009) found that the Finnish organisations they examined adopted SOA based on an IT-driven approach. These organisations started testing SOA principles in small-scale projects and, later, some moved toward more-comprehensive SOA adoption. However, most SOA were based on prototypes or small-scale systems, although organisations with these SOA had been involved with SOA for at least four years.

In conclusion, SOA's scope is suggested to influence SOA's implementation and, consequently, SOA's integration into EA. If SOA is adopted at a smaller scope (small project), then it's expected to have a minimal affect on EA and, therefore, may not even require a significantly huge collaboration between SOA and EA. On the other hand, the larger the SOA scope (e.g., portfolio or enterprise level), the more SOA and EA are expected to integrate and collaborate. Therefore, in this thesis, the scope of SOA (e.g., project, portfolio, enterprise) is employed to study its impact on SOA and EA integration (see Table 2.10).

Table 2.10 SOA scope

SOA scope	Characteristics
Project level	• Little involvement from business side • Scope is limited to individual projects • SOA adoption is based on specific need or tactical basis • No plan or central coordination • The technologies are applied inconsistently, leads to increase of bad SOA practices • Could lead to more complexity (introduces new silos).
Portfolio level	• Portfolio projects guided by plans include integration, consolidation and mainframe migration • Standardisation on SOA platform • Building foundation technologies that can be used for successive SOA projects • Incremental benefits will be achieved through SOA on successive projects.
Enterprise level	• BPM/process automation • Application consolidation • Aligned with business strategy • High involvement of business • Provides active service portfolio management • Promotes SOA best practices • Adopt new technology paradigm consistently across the enterprise as well as increase SOA ROI • The future state includes not only the technology, but also the organisational and process changes as well • The roadmap of strategic SOA is based on a comprehensive assessment of the enterprise and defines SOA projects over a three- to five-year timeframe • Requires organisational alignments.

2.5 SOA's Integration into the Zachman Framework

This section introduces the Zachman Framework and the approaches used to integrate SOA into it[1].

The Zachman Framework was introduced initially to cover only organisations' information systems architecture (Zachman, 1987). Later, the framework was extended to address those aspects that were only loosely covered in the previous version (Sowa & Zachman, 1992). The Zachman

[1] This work was published in Alwadain, Korthaus, Fielt, and Rosemann (2010)

Framework is the first and probably the best-known EA framework. It has been widely used and incorporated into various other frameworks (Riempp & Gieffers-Ankel, 2007; Tang, Han & Chen, 2004; Traverson, 2008; Urbaczewski & Mrdalj, 2006).

The Zachman Framework is arranged in a matrix-like structure (see Figure 2.12) It is a logical structure for organising and classifying an organisation's components that are important to its stakeholders and the development of enterprise systems. The rows of the framework represent six different perspectives on organisations: scope (planner), enterprise model (owner), system model (designer), technology model (builder), detailed representation (sub-contractor), and functioning systems. The columns (abstractions) of the framework represent different ways to describe the real world: data, function, network, people, time, and motivation (Sowa & Zachman, 1992; Zachman, 1987).

The framework shows how the different constructs fit together. In other words, it is a means of viewing a system from many different viewpoints and illustrating how they are connected (Sowa & Zachman, 1992).

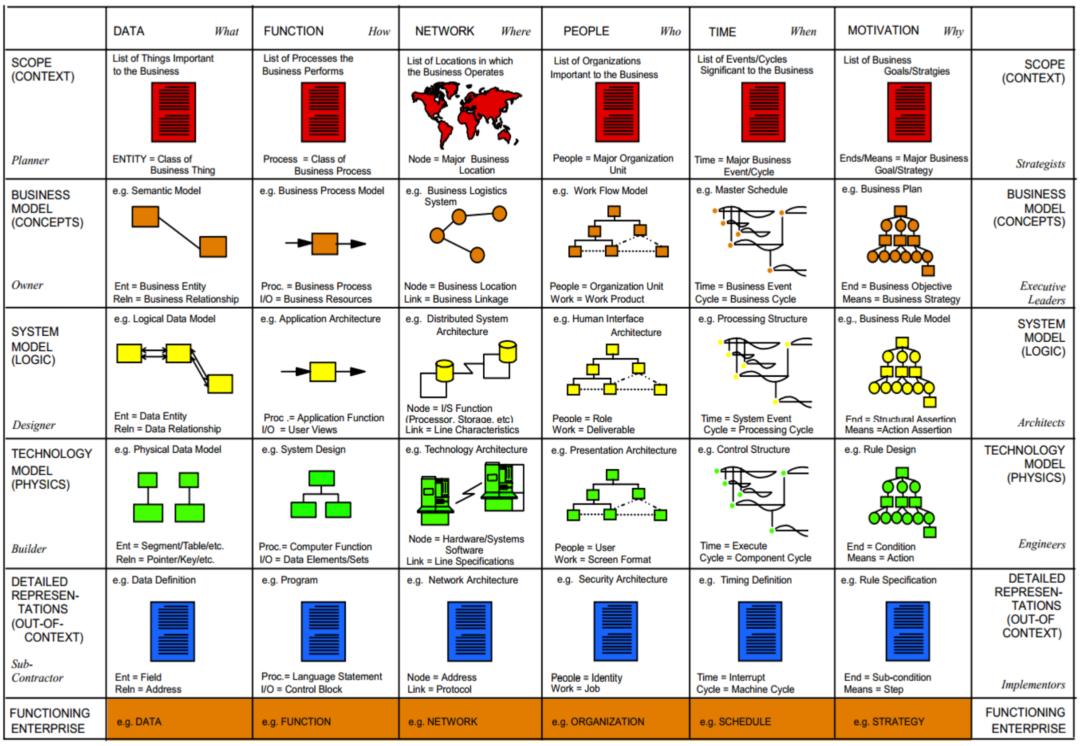


Figure 2.12 Zachman Framework

To integrate SOA into the Zachman Framework, different approaches are possible, including adding a new column or a new row, or integrating SOA in multiple (or particular) cells.

Some approaches have been identified that propose different ways of integrating SOA into the Zachman Framework. Three studies integrate SOA/services into the Zachman Framework as a new column. One study integrates SOA into multiple squares, while another one integrates SOA into an existing column of the Zachman Framework. Section 2.5.1 discusses each approach in detail.

2.5.1 Type One: SOA in A New Column

Correia and Silva (2007) introduced the service concept to EA. They argue that service is a key concept similar to the other core concepts in EA such as data, function, and location. They claim that an integrated and cohesive vision of services in EA is required in order to enhance an organisation's agility. They argue that the service concept is still ambiguous and has multiple semantic meanings and different levels of abstraction, such as business services, technical services, and web services. Correia and Silva (2007, p. 161) perceive a service as "a unit of work done by a service provider to achieve desired end results for a service consumer". Additionally, they see services as a means of linking architectural elements to achieve coherence and flexibility in the operation of these elements.

Due to the lack of service representation in existing EA frameworks, Correia and Silva (2007) propose adding a new column to the Zachman Framework to incorporate the service view. The new column is entitled "whence" and provides information about the source of the service, the service requester, and how it is represented in different views (Table 2.11). The representation of the service changes from each perspective and the models become increasingly fine-grained as we navigate from the higher to the lower rows. For example, at the top, the planner perspective is concerned with strategic planning and organisations' mission. At the level of the owner's perspective, the major concerns are services provided to customers and services requested from partners. The designer perspective is concerned with customising services to stakeholders through market segmentation. The builder perspective focuses on the technical conditions of service availability and systems integration. Finally, the subcontractor perspective focuses on service availability guarantees for the technological components.

Table 2.11 Zachman Framework and SOA: first approach

	What	How	Where	Who	When	Why	Cell Example	Whence Provider/ consumer	Whence Service
Scope (Planner)							Strategic definition of core business	Industry	Business Model
Business Model (Owner)							Definition of core services	Major suppliers, partners and customers	Business outsourcing, Partnership contracts with SLA
System Model (Designer)							Market Segmentation	B2B, B2C, B2E	Customisatio n
Technology Model (Builder)							Systems Integration	CRM, ERP, SRM	SOAP, web services, XML
Detailed Representati ons (Subcontract or)							Pay as you go, IT Outsourcing	Software and Hardware constructor	Support and maintenance
Functioning Enterprise									

Secondly, Khoshnevis, et al. (2009) argue that SOA artefacts are not explicitly included in the Zachman Framework. Thus, in order to provide the framework with the needed capabilities to represent SOA artefacts, they propose a model-driven approach to extend it. The authors argue that the service artefacts have to be represented at all five perspectives (planner to sub-contractor) in the Zachman Framework (Table 2.12). Khoshnevis, et al. (2009) also propose a method for modelling the service column (except in the first and the sixth rows). They argue that the first perspective is not a model, but rather a list of things described in a natural language, and that the sixth row is not a perspective and represent the actual deployed components of an organisation. To model the other perspectives, Khoshnevis, et al. (2009) chose a computation-independent model for the second perspective, a platform-independent model for the third perspective, a platform-specific model for the fourth perspective, and code for the fifth perspective.

Table 2.12 Zachman Framework and SOA: second approach

	Data	Function	Network	People	Time	Motivation	What Services
Scope (Planner)							List of business Services
Business Model (Owner)							Business Service Model
System Model (Designer)							Logical System Service Model
Technology Model (Builder)							Physical System Service Model
Detailed Representations (Subcontractor)							Service Implementation
Functioning Enterprise							Functioning Service Oriented Enterprise

Thirdly, Scheithauer, Augustin, and Wirtz (2009) used the Zachman Framework to classify service description notations in the service ecosystems context. Such an approach facilitates the identification of service description notations for each perspective. In their approach, they added a new column to the Zachman Framework (Table 2.13). From the planner perspective, service properties have a strategic semantic service purpose and a list of important properties. From the owner perspective, service proposition value and owner's requirements in regard to the service are represented. From the designer's perspective, a complete service model that is formal and technologically independent is represented. From the builder's perspective, concrete technology properties, such as web services modelling ontology, are adopted. From the subcontractor's perspective, functionality properties such as WSDL and quality of service properties are represented. In the last row, the implemented service description is represented.

Table 2.13 Zachman Framework and SOA: third approach

	Data	Function	Network	People	Time	Motivation	Service
Scope (Planner)							List of important properties
Business Model (Owner)							Value proposition
System Model							Service

(Designer)							Model
Technology Model (Builder)							Service Profile
Detailed Representations (Subcontractor)							Service component
Functioning Enterprise							Service

In all three approaches, SOA/service is integrated into the Zachman Framework by the addition of a new column. Moreover, in all three approaches, SOA or services are considered an essential part of the framework and as important as the other aspects such as data and network. SOA or services are a concern for all the stakeholders in these approaches. SOA or service elements are aggregated into the original perspectives (views) of the Zachman Framework by the addition of a new square (model) to the end of each perspective. However, none of these approaches discuss how the elements of the new column (SOA/service) are associated with the original elements of the Zachman Framework. Further, in contrast to the second approach, the first and the third approaches do not even have a meta-model that explains the relationships between their new column elements.

2.5.2 Type Two: SOA in Nine Squares

Schmelzer (2006) agrees that there are different views of SOA. For example, SOA may be perceived as a form of application architecture while, in other contexts, SOA may be seen as representing an area of concern as broad as enterprise architecture. He believes that disagreement about the concept is caused by unawareness that there are multiple viewpoints. Therefore, in order to understand the relationships between different viewpoints of SOA and to make sense of them, Schmelzer (2006) proposes using the Zachman Framework and attempts to tailor it to accommodate SOA. He suggests that SOA's initial logical position is the application architecture square at the intersection of the "system model" row and the "function" column. However, SOA is not just an approach dealing with applications and functions of the system. Processes are composed of services and business processes are exposed as services in SOA. In addition, SOA also influences information sharing and representation, and the way a network deals with applications (Schmelzer, 2006; Seppänen, 2008). Therefore, SOA goes beyond the application architecture square and affects all the eight neighbouring squares on the Zachman Framework (see Table 2.14). As a result of mapping SOA to the Zachman Framework, architects have a clear understanding of the relationships between the various SOA components (Schmelzer, 2006).

Table 2.14 Zachman Framework and SOA: multiple squares approach

	Data	Function	Network	People	Time	Motivation
Scope (Planner)						
Business Model (Owner)	SOA	SOA	SOA			
System Model (Designer)	SOA	SOA	SOA			
Technology Model (Builder)	SOA	SOA	SOA			
Detailed Representations (Subcontractor)						
Functioning Enterprise						

In this integration type, SOA is positioned in nine squares in the Zachman Framework (i.e., SOA and the elements of the nine squares share the same position in the framework). Further, no meta-model is provided as to how SOA elements and the original elements of the cells are supposed to be integrated or modelled. However, Schmelzer (2006) states that IT assets are represented as services on the application architecture square. He also explains how processes and services are associated. He generally argues that SOA affects all eight neighbouring squares of the application architecture square.

Further, only three perspectives—the owner, designer, and builder—are concerned with SOA. In other words, SOA is not a concern for the planner and the sub-contractor. In other words, SOA does not have any strategies, goals, or objectives that are important to the planner. Regarding the affected columns, SOA is only considered part of the Zachman Framework's data, function, and network aspects.

2.5.3 Type Three: SOA in the Network Column

Laplante, Zhang, and Voas (2008) use the Zachman Framework to clarify the differences between SOA and software as a service (SaaS). They defined SOA as an architectural strategy intended to change the way internal systems are built and the way systems interact. In SOA, reusable services are the essential elements of the software system. SOA is used to enable the publishing, discovery, and use of these services. Services interact through well-defined interfaces and protocols and can be further used to build new software components, which can be published as a new service (Laplante et al., 2008).

Laplante, et al. (2008) argue that SOA fits into the network (where) column in the Zachman Framework (see Table 2.15), because SOA focuses on connections among its elements in the bigger picture. From the planner's perspective, the SOA network model is a list of possible services to be used in a software system under development. From the owner's perspective, SOA constitutes a collection of existing services to be used in the system. At the designer level, SOA represents an architectural model specifying interaction patterns between service components. The builder's perspective depicts the identification and selection of necessary technology (e.g., web services) to realise the interaction model. At the subcontractor level, the concern is about the list of languages, protocols, and services used. Finally, at the functioning system level, the main concerns are the management and monitoring of all collaboration and communication among services and service components.

Table 2.15 Zachman Framework and SOA: existing column approach

	Data	Function	Network	People	Time	Motivation
Scope (Planner)			SOA			
Business Model (Owner)						
System Model (Designer)						
Technology Model (Builder)						
Detailed Representations (Subcontractor)						
Functioning Enterprise						

In this integration type, SOA is positioned in the Zachman Framework's network column. SOA elements and the elements of the network column occupy the same position. However, neither details nor a meta-model are given to explain the relationships between those elements. In this integration type, SOA is considered a part of all the framework's six perspectives.

2.5.4 Comparison

In order to provide an overview and compare the commonalities and differences of the five approaches discussed in this paper, a set of comparison criteria was selected. To this end, this thesis draws from Jamshidi, Sharifi, and Mansour's (2008) evaluation criteria to compare SOA integration types into Zachman. In addition, this thesis also adopts some metrics from Franke, et al. (2009) who compare different enterprise architecture frameworks in terms of architecture governance and modelling concepts. However, the limited amount of conceptual background information provided by some of the studies discussed in this chapter restricted the comparison's comprehensiveness.

Table 2.16 Comparison of SOA integration into Zachman Framework

Approaches	Perspectives (viewpoints) affected by SOA					Aspects (abstraction) affected							Perspective		Focus On		Type Of integration				Meta-model
	Planner	Owner	Designer	Builder	Sub-Contractor	Data	Function	Network	People	Time	Motivation	Service	Business	Technical	SOA	Services	New Column	Multiple cells	On Network Column	Relationship with ZF cells	SOA/Service
1	+	+	+	+	+	-	-	-	-	-	-	+	+	+		+	+			-	-
2	+	+	+	+	+	-	-	-	-	-	-	+	+	+	+	+	+			-	+
3	+	+	+	+	+	-	-	-	-	-	-	+	+	+		+	+			-	-
4	-	+	+	+	-	+	+	+	-	-	-	_	+	+	+			+		-	_
5	+	+	+	+	+	-	-	+	-	-	-	_		+	+				+	-	-

In Table 2.16, rows 1-5 refer to the five approaches presented earlier in the sequence of their appearance. The first column shows which perspectives in the Zachman Framework are affected by the respective SOA integration approach. This column shows that, in all five approaches, SOA is a concern for the owner, designer, and builder perspectives, while four approaches consider SOA as a concern for all the perspectives. Column two depicts which aspects of the Zachman Framework are affected by the different integration types. Since the first three approaches are examples of the integration type that adds a new column to the existing framework, they only affect the added aspect of “service”. As Section 4.2 describes, the fourth approach is based on an integration type that superimposes SOA on nine models (squares) in the Zachman Framework and thereby affects three aspects; namely, data, function, and network, whereas the last approach is restricted to the network aspect. The third column reflects whether the respective approach takes on a business-oriented or technical view (or both). Only the last approach seems to neglect the business perspective completely. The fourth column distinguishes between the concepts of SOA on the one hand and service on the other. Some approaches target both concepts, while others focus on either the former or the latter. The second-to-last column shows the classification of the integration type chosen in the respective approach. The last column details whether the approach comes with a meta-model that defines the SOA/service concepts and their relationships, and whether it offers information about the relationships with the original Zachman Framework elements.

2.6 SOA’s Integration into Other EA Frameworks

This section examines and compares how SOA elements are integrated in the selected EA frameworks in terms of the completeness of their integration and their relative position in the EA layers (viewpoints)[2].

First, EA frameworks in this thesis’s scope were identified. Second, each framework was reviewed to determine how it integrates SOA elements. Third, the identified elements were grouped into generic categories based on

[2] This work was published in Alwadain, et al. (2011)

similarities. In the fourth step, the elements across the different frameworks were compared.

First, the criteria for selecting EA frameworks were (1) that the frameworks were widely adopted and popular and (2) that they supported SOA elements. A survey by Infosys (2009) concluded that TOGAF, Zachman, and FEAF were the most commonly adopted EA frameworks. In another survey, Varnus and Panaich (2009) found that TOGAF, Zachman, FEAF, DoDAF, Gartner, and MODAF were the most used frameworks (Varnus & Panaich, 2009). Leist and Zellner (2006) claim that ARIS, DoDAF, FAEAF, MDA, TEAF, TOGAF, and Zachman were popular and widely discussed in EA literature. In addition, Winter and Fischer (2007) argue that TOGAF, FEAF, and ARIS were broadly employed EA frameworks. The EA frameworks also needed to have integrated SOA elements and provided documentations and/or meta-models unfolding how SOA elements are integrated in their EAs; see, for example, (Berrisford & Lankhorst, 2009; Federal CIO Council, 2008; US Department of Defence, 2009).

TOGAF, FEAF, DoDAF, and MoDAF met these conditions; thus, they are incorporated in this thesis. ArchiMate is also included in this thesis because it was developed as a service-oriented EA framework and modelling language (Berrisford & Lankhorst, 2009; The Open Group, 2009a). There are, however, other EA frameworks that also met the criteria. For instance, the Zachman Framework was investigated in Section 2.5. The Gartner framework was excluded, because it was not entirely accessible due to commercial restrictions (Franke, et al., 2009). ARIS was excluded. While Stein, et al. (2008) show how SOA modelling is enabled in ARIS, they do not show how SOA elements fit into the original structure (viewpoints) of the framework.

Second, the SOA elements in the chosen frameworks were identified. When SOA elements were spotted explicitly in a framework's meta-models, then that was regarded as solid evidence that SOA had explicitly been integrated into it ("++"designates this in Table 2.17). Similarly, if SOA elements were discovered explicitly in the textual documentation of the frameworks, then this was considered solid evidence ("+" designates this in

Table 2.17). The identification for each framework is described explicitly to make this process clear and verifiable (Section 2.6.6 discusses the results of the comparison).

During analysing how the various EA frameworks had integrated SOA elements, it was observed that some of the SOA elements represented in the frameworks had conceptual or terminological variations. In order to be able to overview and compare the approaches without losing detail, the original terms were kept as used in the respective frameworks, but were grouped together in more generic categories based on the key elements of the OASIS reference model mentioned in Section 2.4.1. The categories are services, actors (e.g., service providers and consumers), service interfaces, service contracts, and others for elements that could not be grouped together (e.g., service description, service policy, and service function).

2.6.1 ArchiMate

ArchiMate is an EA framework and modelling language (The Open Group, 2009a). The ArchiMate framework is a reference taxonomy scheme for architecture concepts, models, viewpoints and views (The Open Group, 2012a). It was introduced with a focus on service orientation and it defines the following three layers (viewpoints): business, application, and technology. It models the global structure in each layer, the main artefacts, components, and dependencies between them, and the relationships between the layers (Iacob, Jonkers, Lankhorst, & Steen, 2007).

Iacob et al. (2007) show how business processes, applications, and infrastructure can be modelled using a “generic service” concept in ArchiMate. They argue that services have different natures and granularity. Further, services can be provided by organisations to their clients, by applications to support business processes or by infrastructure and hardware to support applications. In addition, they claim that service orientation results in a layered view of enterprise architecture, where the service is one of the major connecting pins between layers. They distinguish between three main layers: business, application, and technology (Lankhorst, 2005; Lankhorst, 2004; Steen, Strating, Lankhorst, ter Doest, & Iacob, 2005). On the business layer, business processes are exposed as business services. On the application layer, application services support business processes. On the technology layer, infrastructure services such as data storage and communication services are designed to support the various application functions (see Figure 2.13).

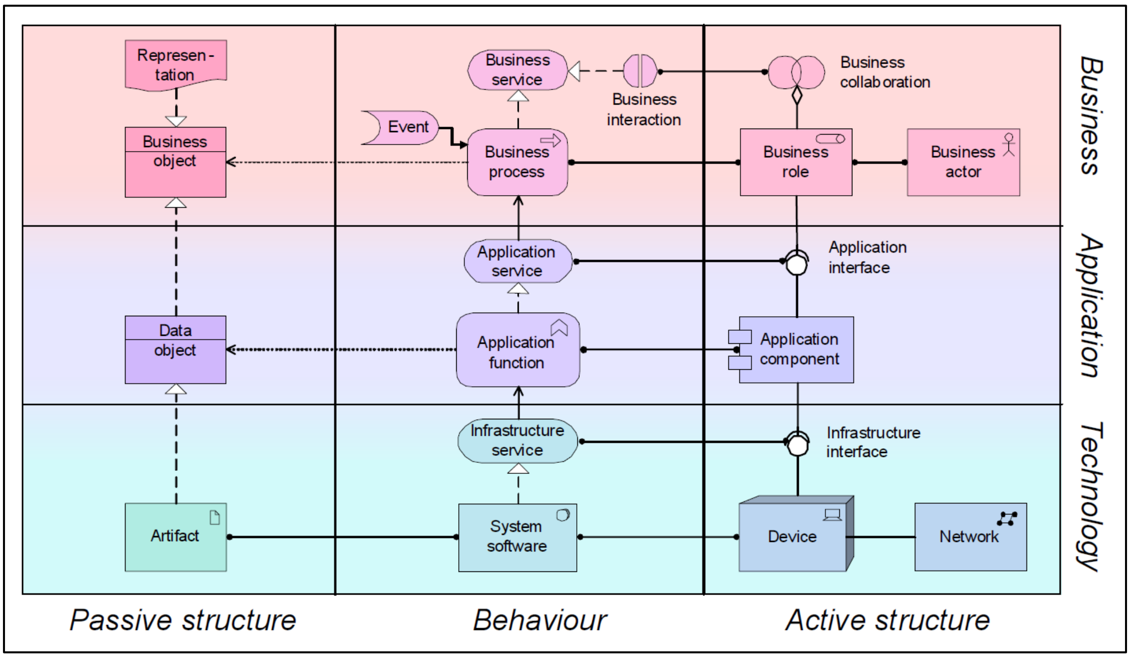


Figure 2.13 ArchiMate (Lankhorst, 2005)

Further, the Open Group had adopted ArchiMate for modelling enterprise architecture (The Open Group, 2009a). In order to identify SOA elements in ArchiMate, this thesis examined ArchiMate specifications (The Open Group, 2009a). First, in the business layer, a product that is defined as a coherent collection of services, a business service, a business interface, and a contract was identified in the meta-model provided. However, the contract is associated with the product (collection of services), not the business service. Service level agreement (SLA) and quality of service (QoS) were recognised in the text of ArchiMate documentations as part of the contract. Secondly, in the application layer, an application service and an application interface were found in the meta-model. Thirdly, in the technology layer, an infrastructure service and an infrastructure interface were recognised in the meta-model.

2.6.2 The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework (TOGAF) was developed by the Open Group in 1995. TOGAF is considered an industry EA framework. The architecture development method (ADM) illustrates the ten different phases of EA development (see Figure 2.14). The ADM techniques and guidelines are provided to support the ADM's application and to deal with different scenarios and different process styles. The content framework provides a conceptual meta-model for describing architectural artefacts. It is not considered compulsory and could be combined with other meta-models. The enterprise continuum is a virtual repository to maintain architectural assets such as models and architectural descriptions. The TOGAF reference models are divided into the TOGAF technical reference model and the integrated information infrastructure reference model. The architecture capability framework is a set of the required skills, roles, and responsibilities to establish and operate an EA. TOGAF divides enterprise architecture into three closely interrelated architectures: business architecture, information systems architecture (application and data), and technology architecture (Buckl, Ernst, Matthes, Ramacher, & Schweda, 2009; The Open Group, 2009d).

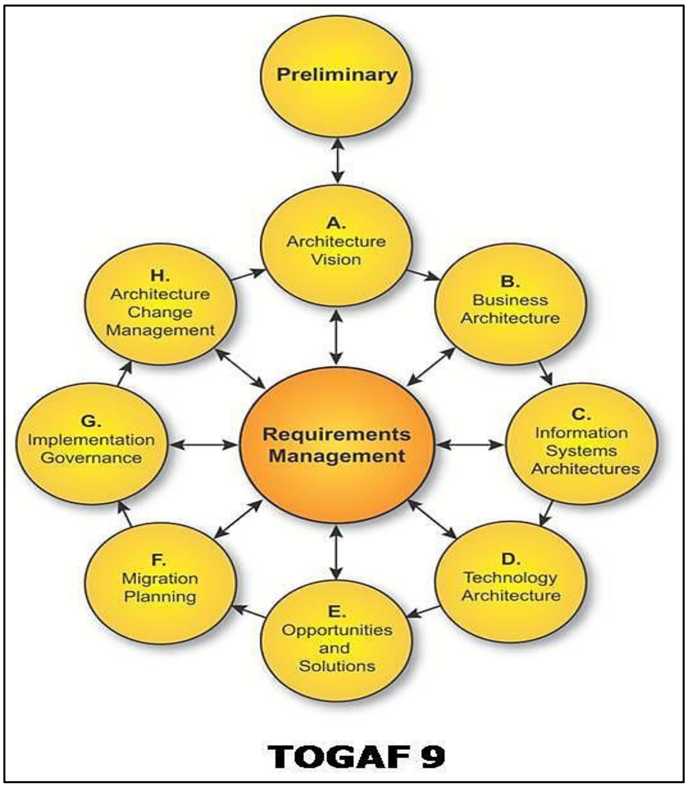


Figure 2.14 TOGAF ADM

SOA elements in TOGAF are represented in its meta-model and explained further in its documentation (The Open Group, 2009b, 2009d). First, in the business architecture, a business service, a contract, service quality, and a measure, which links objectives to business services, were identified in the meta-model. A service-level agreement (SLA) is also found in the documentation. In the application architecture, the meta-model has an information system service, and, in the technology architecture, the meta-model identifies a platform. TOGAF's documentation identifies a service interface, service attributes, and a service policy as part of all three business, application, and technology architectures.

2.6.3 The Federal Enterprise Architecture Framework (FEAF)

The Federal Enterprise Architecture Framework was developed by the U.S. Federal Chief Information Officers Council. The FEAF was introduced to comply with the Clinger-Cohen Act, to develop and maintain integrated systems architectures, and to promote and organise federal information sharing across U.S. Federal Government agencies (Urbaczewski & Mrdalj, 2006). It has five reference models: the performance reference model (PRM), the business reference model (BRM), the service component reference model (SCRM), the technical reference model (TRM), and the data reference model (DRM). The reference models are designed to standardise terms and definitions in EA contexts and to facilitate sharing and collaboration across the U.S. Federal Government (Federal CIO Council, 2008). In order to identify SOA elements in the FEAF, its SOA practical guide was investigated (Federal CIO Council, 2008). However, the framework provides no meta-model. Thus, the documentation itself was used to evaluate the extent to which SOA elements and artefacts are integrated in the framework. The service model is apparent in the BRM and the SCRM. The documentation recognises the elements and artefacts of service portfolio, business service, IT service, quality of service, SLA, service contract, service consumer, and service provider.

2.6.4 The Department of Defence Architecture Framework (DoDAF)

DoDAF V2.0 is a comprehensive, overarching framework and conceptual model that enables the development of architectures to assist Department of Defence (DoD) managers to make more-effective key decisions. The vision for DoDAF is for it to provide a comprehensive set of architecture concepts, methods, and best practices to facilitate architecture development; to define and institutionalise the Internet-centric data strategy and Internet-centric services strategy of the DoD; and to define, describe, and develop services through the SOA's introduction. DoDAF V2.0 has different viewpoints: systems viewpoint (SV), service viewpoint (SvcV), data and information viewpoint (DIV), operational viewpoint (OV), standards viewpoint (StdV), capability viewpoint (CV), project viewpoint (PV), and all viewpoints (AV) (US Department of Defence, 2009). The DoDAF documentation was reviewed to identify SOA elements. In the DoDAF generic meta-model, a service (including business and software services), service description, a service port, and a service performer (both a service consumer and a service provider) were found. The DoDAF documentation contains service function, service contract (SLA is part of it), service policy, QoS, and service channel. The main viewpoint with SOA elements is the service viewpoint (SvcV). However, these elements may appear in some other viewpoints, such as all viewpoints and capability viewpoint, when mapping services to capabilities.

2.6.5 The Ministry of Defence Architecture Framework (MODAF)

The MODAF is an EA framework developed by the U.K. Ministry of Defence to support its planning and management activities. The MODAF provides a consistent set of rules and templates, known as views, which provide a textual and graphic visualisation of an area of the organisation. Each view provides a different perspective on the business to match different stakeholder interests. The views are divided into seven categories: strategic views, operational views, service-oriented views, systems views, acquisition views, technical views, and all views. The MODAF offers a meta-model that defines the relationship between all the data in all the views (The UK Department of Defence, 2010a). The service-oriented viewpoint includes seven views to provide a perspective that enables service specification, behaviour, and policies. The views do not focus on the detailed implementation of services, but on requirements that the services fulfil. A service, service interface, SLA, service policy, service function, service attributes (description), and service consumer were identified in the MODAF's models as SOA-related elements (The UK Department of Defence, 2010b).

2.6.6 Comparison

The five selected EA frameworks were compared in order to see the commonalities and differences between the SOA elements they cover and the position of these elements in each framework's layers (viewpoints) (see Table 2.17). The table presents the names of the original SOA elements and their grouping. Table 2.17's columns represent the five frameworks and only the layers (viewpoints) that have SOA elements because some frameworks represent SOA elements in one layer (e.g. DoDAF). The rows show how the identified SOA elements are grouped according to the OASIS SOA's reference architecture.

Table 2.17 SOA elements in EA frameworks

EA frameworks		ArchiMate			TOGAF 9			FEAF		DoDAF v2.0	MODAF
SOA elements \ Layers (views)		Business	Application	Technology	Business	Information Systems	Technology	BRM	SCRM	Services View	Service-Oriented Viewpoint
Services category	Service									++	++
	Business service	++			++			+		+	
	Application service		++							+	
	Is service					++					
	Enterprise service								+		
	Infrastructure Service			++							
	Platform service						++				
Actors category	Actor	++			+						
	Service consumer							+	+		++
	Service provider							+	+		
	Performer									++	
Interfaces category	Business interface	++									
	Application interface		++								
	Infrastructure interface			++							

EA frameworks		**ArchiMate**			**TOGAF 9**			**FEAF**		**DoDAF v2.0**	**MODAF**
	Service port									++	
	Service interface				+	+	+	+	+		++
Contracts category	Contract	++			++			+	+	+	
	Sla	++			+			+	+	+	++
	Service conditions	++								+	
	Qos	+			++			+	+	+	
Others category	Product	++									
	Measure				++						
	Service description				+	+	+			++	++
	Service policy				+	+	+			+	++
	Service channel									++	
	Service function									+	++

The first category is services. The service is found in all the frameworks; however, its details differ substantially in the details. For example, a generic service element is identified in the meta-models of DoDAF V2.0 and MODAF, while a business service is recognised in the meta-models of ArchiMate and TOGAF and in the documentations of the FEAF and DoDAF V2.0. In addition, an application service is identified in the meta-models of ArchiMate and DoDAF V2.0. Further, an information system service is recognised in TOGAF's meta-models, and an enterprise service is identified in FEAF's documents. Further, an infrastructure service is found in ArchiMate's meta-model, while a platform service is identified in TOGAF's meta-model.

The second category is the actors. In ArchiMate's and TOGAF's meta-models, an actor is represented in the business layer. A service provider is found in the FEAF's documents in BRM and SCRM, while a performer, which could be a service provider or a consumer, is identified in the MODAF's meta-models.

The third category is service interfaces. In ArchiMate's meta-models, a business interface is found in the business layer, an application interface is found in the application layer, and an infrastructure interface is found in the technology layer. All these interfaces are linked to the services in the same layer. However, in the meta-model of DoDAF V2.0, it is called a service port, while, in the documents of TOGAF and the FEAF and in the MODAF's meta-models, it is called a service interface.

The fourth category is service contracts. A contract is recognised in ArchiMate's meta-models in the business layer, in TOGAF's meta-models in the business architecture, in FEAF's documents in BRM and SCRM, and in the DoDAF V2.0 documents. However, in ArchiMate, the contract is associated with the product, which is a collection of services. An SLA is identified in ArchiMate's meta-models in the business layer as part of the contract and also in TOGAF's document as part of the contract. It is also recognised in FEAF's documents in BRM and SCRM viewpoints and in DoDAF's services viewpoint. The SLA is found in the meta-models of the MODAF's service-oriented viewpoint. The service conditions element is identified in ArchiMate's meta-models as a part of the contract. In contrast, it is found in DoDAF's documents as part of the service description in the services view. However, it is not mentioned in other frameworks. The Quality of Service (QoS) is identified in ArchiMate's document in the business layer as part of the contract and TOGAF's meta-model in the business architecture. It is also recognised in FEAF's document in both BRM and SCRM viewpoints and in DoDAF's documents in the services view.

Finally, there are a couple of single elements that are not grouped. First, a product that is defined as a coherent collection of services accompanied by a contract is found only in ArchiMate's meta-model in the business layer. Next, a measure element, which links the objective and business service, is only identified in TOGAF's meta-models in business architecture. A service description is found in TOGAF's document in all three architectures in DoDAF's and MODAFs' meta-models. Next, a service policy is identified in TOGAF's document in all three layers, in DoDAF's documents, and in MODAF's meta-models. Then, a service channel, which is a logical or physical communication path between requisitions and services, is recognised only in DoDAF's meta-models. Finally, a service function is only found in DoDAF's documents and in MODAF's meta-models.

2.7 SOA's Integration into EA: Other Examples

This section introduces other studies that have integrated SOA into EA. The other literature findings suggest that there are various integration outcomes. Due to the fact that EA frameworks use different architectural layers, this thesis tried to find common layers of EA to use them to structure these findings. These EA layers—business, information systems (information, applications), and technology—are widely accepted and used in the enterprise architecture discipline (Joachim, et al., 2011; Lankhorst, 2004; Pulkkinen, 2006). Additionally, the Open Group Architecture Framework (TOGAF), a widely used EA (Infosys, 2009), uses a similar structure: business, information systems, and technology (The Open Group, 2009e). Therefore, this thesis adopts these three layers to structure the literature findings in this chapter and thereafter.

2.7.1 Business Architecture

This section presents the studies that integrate SOA into business architecture. This level of integration often implies that lower architectures (e.g., IS and infrastructure architectures) are (or will be) service-oriented. Examples of studies that discuss SOA integration into business architecture are presented in the following paragraphs.

First, some organisations have become a service-oriented enterprise (SOE) where services are the major component of all architectural layers. Intel is an example of such an organisation and its architectural vision is shown in Figure 2.15 (Hirschheim, et al., 2010).

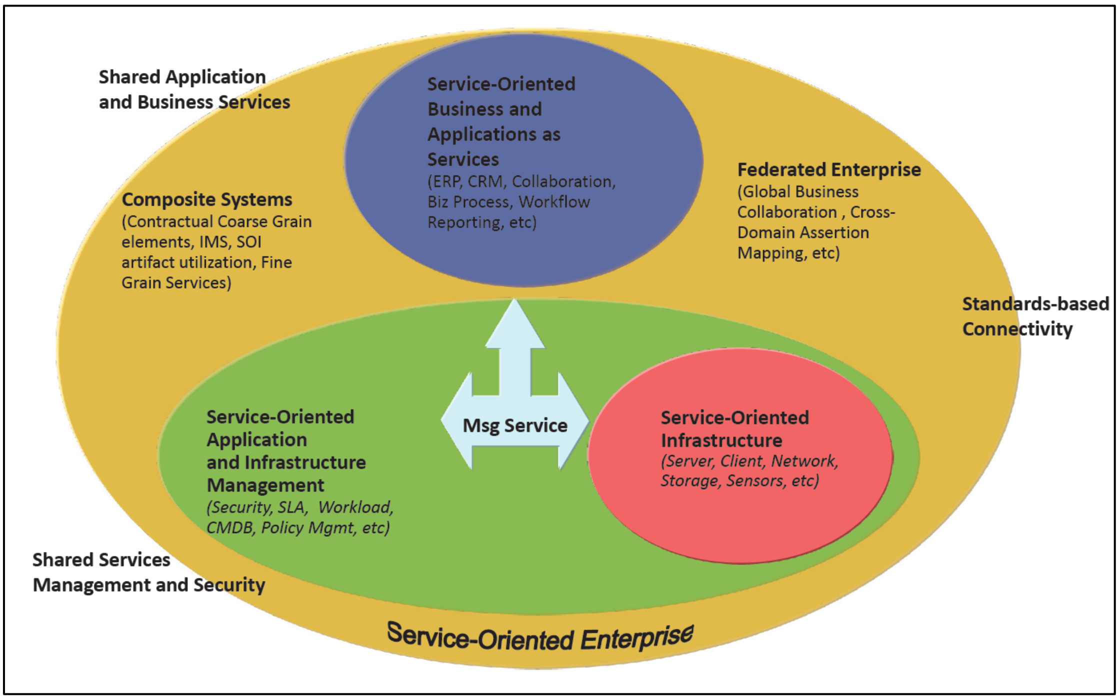


Figure 2.15 Service-oriented enterprise (Hirschheim, et al., 2010)

Second, Chen (2008) suggests a service engineering schematic that offers a multi-disciplinary approach to service engineering. One of the goals of the schematic is to provide an architectural blueprint for service systems development and service modification. The schematic has three layers: the business model layer, the business architecture layer, and the IT architecture layer, (see Figure 2.16). These layers denote the traditional separation of concerns concept. Service-oriented enterprise architecture is depicted in the middle column and positioned in a complex socio-technical system. Services are in a hierarchical order, from coarse-grained to fine-grained, in the three layers. Business services are represented in layer one, application and process services are in layer two, and infrastructure services are in layer three. Business services are considered from a customer-driven requirements or business goals perspectives. Process and application services are orchestrated and composed to deliver the business services. Infrastructure services are aggregated to support the application and process services. Interfaces in each layer are well designed and defined to allow services to interact.

Further, meta-data is used to manage the application portfolio, business processes, and services. In the meta-database, all services are registered to promote service reuse. Service requesters, providers, operations, discovery, service level management, and routing are managed in the meta-database.

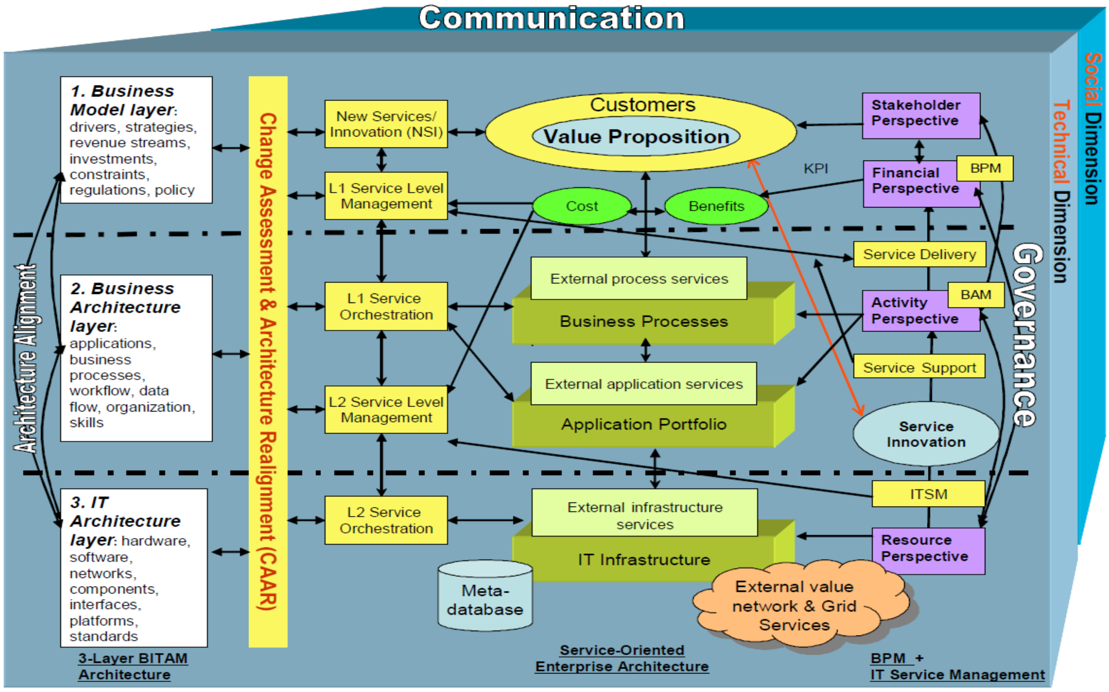


Figure 2.16 Service engineering schematic (Chen, 2008)

Third, a power-supply company in Germany partially service-oriented its business architecture in addition to its IS (Schelp & Aier, 2009). Its SOA initiative was driven by its IS department and limited to some business departments to reduce implementation time. Business owners and architects were involved in the initiative, with a focus on IS architecture management and business process management for selected business departments due to the lack of holistic EA. The identified architectural levels are business process architecture, business service architecture, and basic service architecture (software architecture).

Fourth, Butler (2007) discusses the CBDi-SAE Reference Framework, which is designed to provide a complete framework for all the essential components to assist the migration to and maintain service-oriented enterprises. It addresses, among multiple issues, the architecture of SOA artefacts. The reference framework includes five SOA views: business, specification, implementation, deployment, and technology (see Figure 2.17).

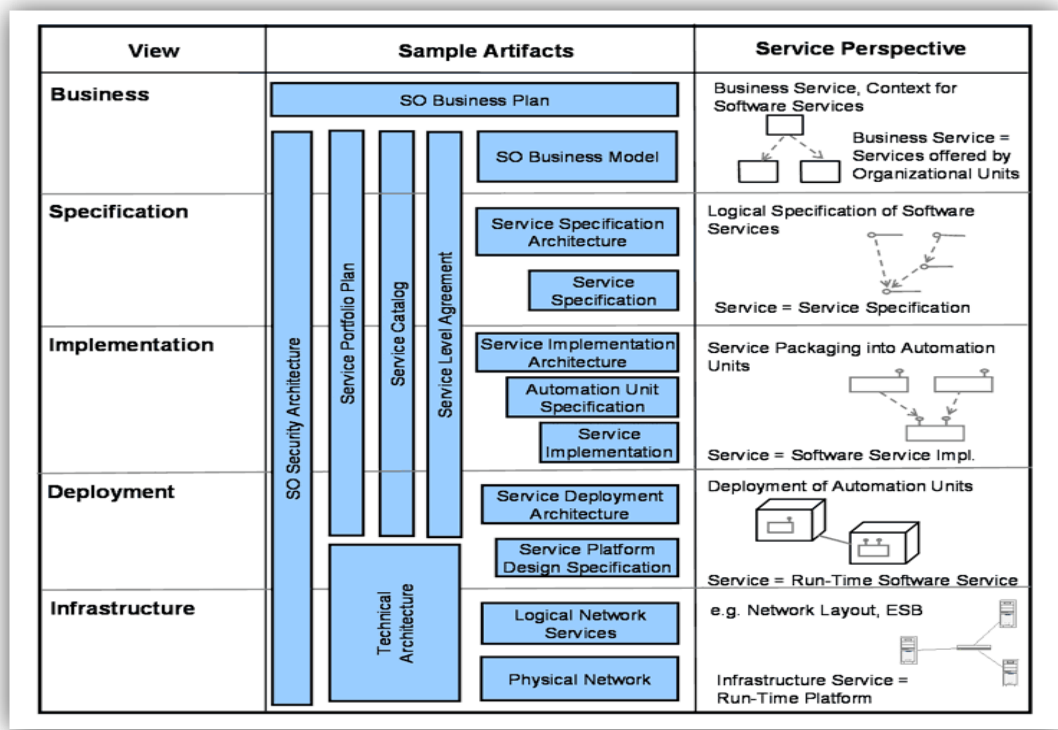


Figure 2.17 SOA views of CBDi-SAE (Butler 2007)

These views represent a level of abstraction and hold certain artefacts that are important to distinct stakeholders on each view. Further, these views are parts of the enterprise sub-architecture's "layers". The business view is part of the business layer. Specification, implementation, and deployment views are part of the software layer. Deployment and technology are part of the infrastructure layer. SOA artefacts are represented in these views to suit the relevant stakeholders. For example, the business view includes business services offered by business or organisational units. The specification view includes the necessary artefacts that assist architects in specifying functional and non-functional requirements of the applications and services and the dependencies between them. The main diagram in this view is the service dependency diagram, which shows the service architecture layers, the services on each layer, and the dependencies between them. Additionally, the implementation view has service implementation architecture as its primary artefact, which represents the structure of the software components that realise the services specified in the previous service specification architecture. Further, the deployment view represents the allocation of services and software components to platforms or network nodes. Finally, the technology (infrastructure) view represents infrastructure services, logical network layout, processing nodes, network nodes, services operations, governance policies, and communication channels between services. It also includes dependencies between technologies used to realise SOA. In addition to all these views, the multi-view is an overarching view that comprises service orientation, security architecture, service portfolio plan, service catalogue, and service-level agreement.

Fifth, Aier and Gleichauf (2009) propose three layers of enterprise architecture representing three sub architectures: service-oriented process architecture, service-oriented integration architecture, and service-oriented software architecture (see Figure 2.18).

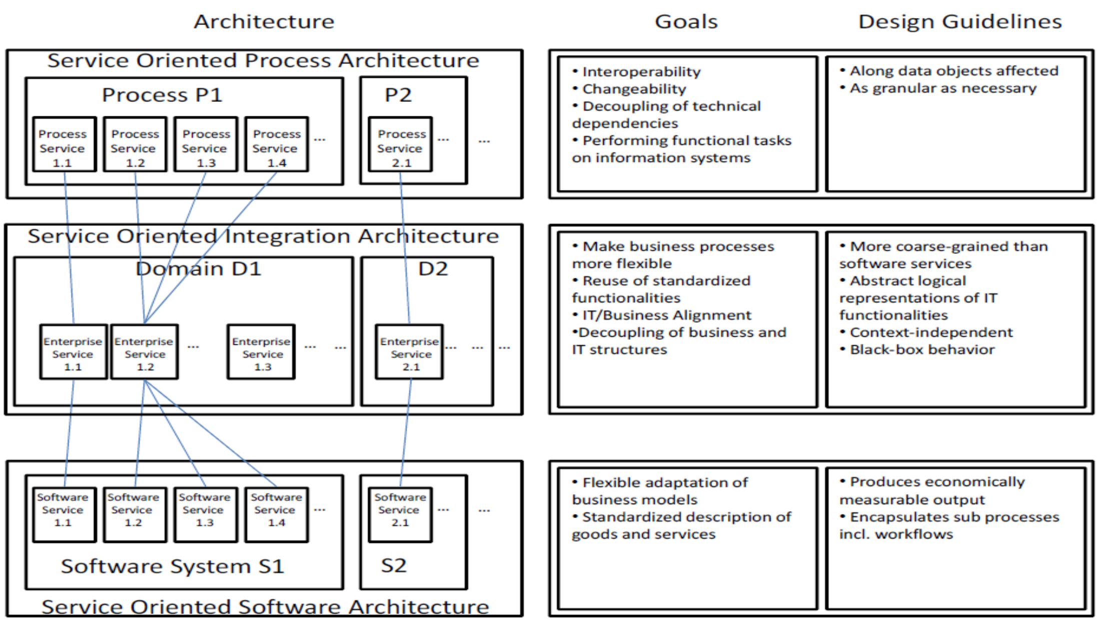


Figure 2.18 Three SOEA layers (Aier & Gleichauf, 2009)

Software services, which realise technical tasks, are represented on the software architecture layer, while enterprise services (business services), which encapsulate business functionalities, are represented on the integration architecture layer. Enterprise services support business processes that are positioned on the process architecture layer. Service-oriented software architecture aims for manage applications changeability and interoperability. Changeability and flexibility are achieved by decoupling and reducing technical dependencies. Thus, the main guidelines for designing software services are loose coupling and interoperability, which provide advantages during implementation. Further, service-oriented integration architecture's aims for infrastructure flexibility and standardised functionalities. Flexibility is achieved through the creation of enterprise services in order to orchestrate flexible business processes.

2.7.2 Information Systems Architecture

This section presents the studies that integrate SOA into information systems architecture. First, Schelp and Aier (2009) report the findings of introducing SOA into a bank in Switzerland. They distinguish between several architectural levels: business, application and integration, software component, and technical architecture. SOA was introduced to meet the integration complexity of more than 450 systems. The bank introduced the program, because of an increasing demand for application integration and the need to reduce the resulting integration complexity.

Second, Schelp and Aier (Schelp & Aier, 2009) also discuss a telecommunication service provider case in Germany that started its SOA initiative to reduce the complexity of its distributed application landscape. In regard to services integration, enterprise services are integrated in integration architecture, while basic services (software components) are integrated in software architecture.

Third, in Erl's (2005) enterprise model, the service layer is located between the business process layer and the application layer where most of SOA characteristics are prevalent. These characteristics are loose coupling, support service-oriented business modelling, agility, and layers of abstraction.

Erl (2005) argues that, in order to achieve one of SOA's most outstanding features, enterprise-wide loose coupling, a model needs to have separate layers of services. In other words, the service layers constitute a loosely coupled relationship between application and business logic. Consequently, this structure will allow the business processes to evolve without necessarily changing the technical level responsible for their implementation. Therefore, Erl (2005) divides the service layer into three layers of abstractions. The first layer, the application service layer, contains services that are designed to represent application logic. The second layer, the business layer, is represented above the application service layer. Services on this layer are designed to represent business logic. By adding this layer, an important characteristic of SOA, service-oriented business modelling, is supported. The third layer, the orchestration layer, acts as a controller and a centralised location for business and composition rules. This layer introduces the notion of process services that are capable of composing other services to complete a business process. The addition of the orchestration layer and the organised service abstraction significantly increase organisational agility.

Fourth, Engels and Assmann (2008) and Assmann and Engels (2008) propose integrating the services layer between business and IT architectures. Business architecture contains organisational goals, organisation, and processes. IT architecture contains software and infrastructure architectures. The historical gap between the business process layer and the IT application layer, caused by the design of IT systems as functional silos, is one of the main ideas behind SOA's emergence as a concept to bridge the gap. IT systems' granularity and aggregation of functions was technically driven in a silo system. Therefore, changes to business processes often led to an entirely different granularity and aggregation of IT functions. Thus, the service layer is introduced between business architecture and IT architecture to bridge the gap (see Figure 2.19). Bridging the gap eases implementing business processes with IT and services' reuse.

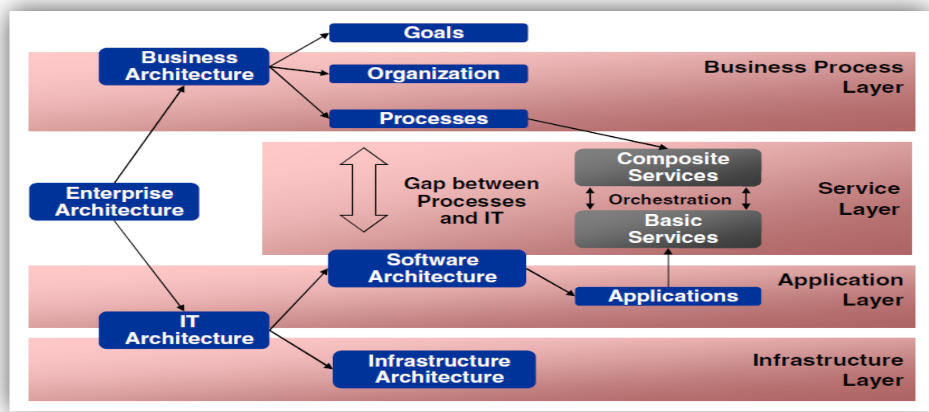


Figure 2.19 Service layer (Assmann & Engels, 2008)

2.7.3 Technology Architecture

This section presents the studies that integrate SOA into technology architecture. For example, the N.S.W Department of Lands adopted a service-oriented architecture approach, and service-oriented its technical architecture. Figure 2.20 shows the gradual process of service-orientation activities. First, it started with a focus on the Department of Lands' technical architecture when an enterprise service bus (ESB) first introduced in 2005. Then, the journey of SOA continued until 2007 to include the applications landscape Harris (2008). The four sub-architectures of the department's EA are: business, information, application, and technology (SOA) architectures.

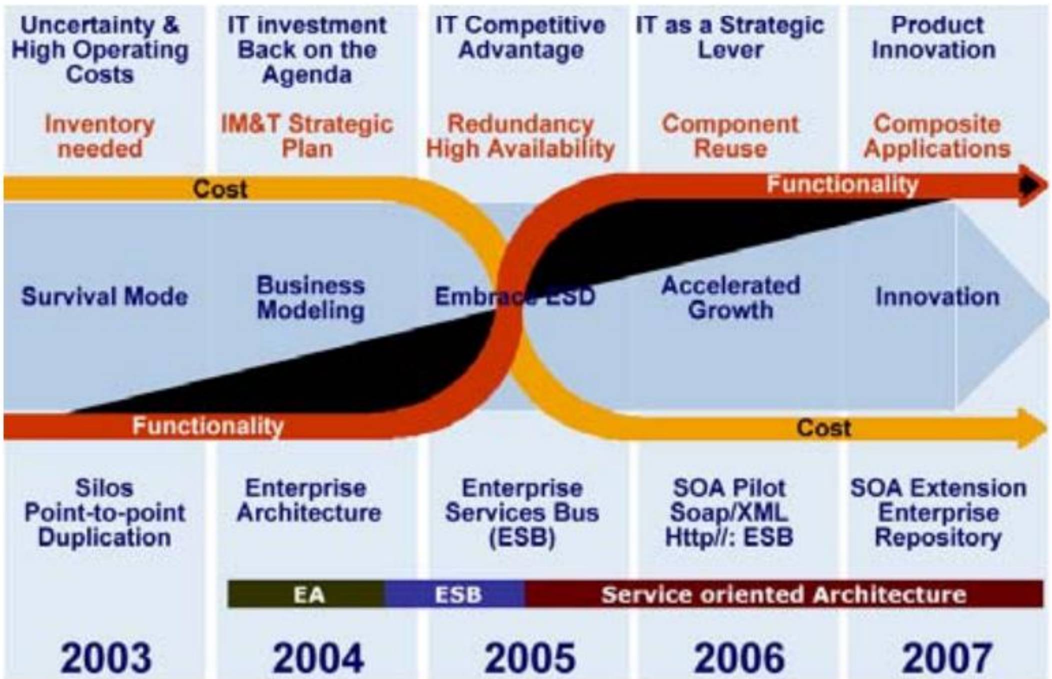


Figure 2.20 N.S.W Departments of Lands' SOA

A logistics operator in Finland adopted SOA during 2005; it was an IT-driven pilot in one business unit to test the feasibility of web service technology. The SOA adoption process started using a technical bottom-up approach, and SOA was almost equal to web services. Later, it was expanded into multiple projects to integrate the legacy system landscape, employing SOA technology mainly to expose legacy system services via an integration platform (Kokko, et al., 2009).

Further, a public sector organisation in Finland adopted a service platform that was complemented with a J2EE-based infrastructure platform to support XML and web service interfaces. It took an external vendor three years to build the platform. Later, the platform was expanded in iterative SOA projects (Kokko, et al., 2009).

2.8 Chapter Summary

This chapter introduces the phenomena under investigation (namely, enterprise architecture (Section 2.2), EA evolution (Section 2.3), Service-oriented Architecture (Section 2.4), and the factors (generative mechanisms) that might influence EA evolution, and serves to scope and ground this thesis. Section 2.3 shows the lack of empirical work addressing EA evolution. Moreover, Sections 2.5 to 2.7 cover this thesis's specific focus: SOA's integration into EA. It presents studies in which SOA has been integrated into EA. These studies are compared and classified where possible. The majority of these studies vary in their integration approaches, and do not offer a deeper understanding of the EA evolution process. This chapters ends by selecting the three widely used EA layers (business, information systems, and technology) to organise the outcomes of EA evolution due to the heterogeneous nature of EA frameworks, layers, and structures. This chapter informs Chapter 4, in which this thesis's a-priori model is developed.

Chapter 3: Research Methodology

3.1 Introduction

This chapter presents the research methodology adopted for this thesis and its underlying philosophical foundations. This thesis adopts a critical realist theory—Archer's (1995) morphogenetic theory—and uses a critical realist methodological framework to guide its overall conduct. Further, given the exploratory nature of the research questions, the complex nature of EA evolution, and the need for qualitative analysis to disclose the generative mechanisms, a multi-method qualitative approach was deemed appropriate. In particular, an explorative interview phase was conducted to explore EA evolution and to extend the a-priori model. Consequently, a multiple case study was conducted to further explore the developed a-priori model in real settings, as is suggested in the critical realist methodological framework.

The chapter progresses as follows. Section 3.2 presents the philosophical assumptions that underpinned this thesis's development, and Section 3.3 explicitly clarifies the chosen philosophical position. Section 3.4 outlines the critical realist methodological framework used to guide this thesis's process. Section 3.5 briefly presents and justifies the employed qualitative multi-method approach. This section argues that the qualitative multi-method approach is appropriate due to the complex nature of integrating SOA into EA and the lack of previous empirical studies that discuss the process. Section 3.6 details the research plan that comprises four main phases and is mapped to the adopted critical realist methodological framework. It also covers the details of the chosen research methods and their justifications. Section 3.7 addresses the reliability and validity measures employed to ensure the quality of this thesis, and Section 3.8 summarises the chapter.

3.2 Philosophical Foundations

Healy and Perry (2000) compare four philosophical paradigms based on three main elements: ontology, epistemology, and common methodologies (see Figure 3.1). More specifically, they provide an overview of the relationship that exists between these three elements in each paradigm. They explain that "ontology is the 'reality' that researchers investigate, epistemology is the relationship between that reality and the researcher, and methodology is the technique used by the researcher to investigate that reality" (2000, p. 119).

	Paradigm			
Element	**Positivism**	**Critical theory**	**Constructivism**	**Realism**
Ontology	Reality is real and apprehensible	"Virtual" reality shaped by social, economic, ethnic, political, cultural, and gender values, crystallised over time	Multiple local and specific "constructed" realities	Reality is "real" but only imperfectly and probabilistically apprehensible
Epistemology	*Objectivist*: findings true	*Subjectivist*: value mediated findings	*Subjectivist*: created findings	*Modified objectivist*: findings probably true
Common methodologies	*Experiments/surveys*: verification of hypotheses, chiefly quantitative methods	*Dialogic/dialectical*: researcher is a "transformative intellectual" who changes the social world within which participants live	*Hermeneutical/dialectical*: researcher is a "passionate participant" within the world being investigated	*Case studies/convergent interviewing*: triangulation, interpretation of research issues by qualitative and by some quantitative methods such as structural equation modelling

Figure 3.1 Scientific paradigms and their elements (Healy & Perry, 2000, p. 119)

Positivist research is grounded on the existence of a priori relationships in the investigated phenomenon and employs structured instrumentation. It is often concerned with empirically testing theories to verify or falsify theories. Positivism dominates in science because scientists quantitatively measure independent observations about a single comprehensible reality (Healy & Perry, 2000). Positivist research is widely adopted in the information systems discipline (Gable, 1994; Kaplan & Duchon, 1988). However, positivism approaches do not often consider contextual factors, which limits their contribution potential (Kaplan & Duchon, 1988).

Critical theory is concerned with critiquing existing social systems, contradictions, and/or changes (Orlikowski & Baroudi, 1991). Critical theory research aims to transform social, cultural, political, and gender values. Its studies are often long-term, and historical and/or ethnographic in nature. This paradigm is not suitable for business studies except when they aim to be "transformative intellectual"; in other words, when the objective is to liberate people from their historical emotional, mental, and social structures (Healy & Perry, 2000, p. 119).

The constructivist (interpretive) paradigm assumes that subjective meaning is created and associated with people as they interact with the world around them. The paradigm fundamentally emphasises the subjective meaning that participants assign to the investigated phenomenon (Orlikowski & Baroudi, 1991). According to Healy and Perry (2000, p. 120), this approach is not appropriate in business studies because it "excludes concerns about the important, and clearly 'real', economic and technological dimensions of business".

Finally, critical realism (CR) assumes an objective reality and that reality is stratified. It consists of structures and mechanisms that generate the observed events (Bhaskar, 1975; Healy & Perry, 2000; Tsang, 2013). Bhaskar (1975) has long advocated critical realism, and he has established a coherent critical realism language (Danermark, et al., 2002). Critical realism has attracted interest in recent years due to the criticisms and limitations of both positivism and constructivism (Danermark, et al., 2002; Wynn & Williams, 2012). Many scholars, including Archer (1995), Pawson and Tilley (1997), Sayer (1992), and Danermark, et al. (2002) have contributed to critical realism. Conceivably, the distinguishing feature of critical realism is its view of causality

In Information Systems, CR provides interesting views in examining contemporary IS-related issues. It offers a vigorous basis for using a variety of methods in order to achieve an improved understanding of the meaning and significance of information systems in the contemporary world (Mingers, Mutch, & Willcocks, 2013).

3.3 Critical Realism

This section provides an overview of critical realism and then discusses the critical realist methodological framework that guides this thesis.

3.3.1 Overview

One of CR's key assumptions is that *"the world exists independently of our knowledge of it"* (Sayer, 1992, p. 5). Or, put differently, CR *"claims that a world outside and independent of our conscious perception exists (reality)* *and that only some aspects of this world are objectively knowable via our senses"* (Johnston & Smith 2008, p. 28).

The critical realist perspective regards social systems as open systems that are subject to frequent change through interactions. It presumes *"that social science studies are conducted in open systems, that reality consists of different strata with emergent powers, that it has ontological depth and that facts are theory laden"* (Danermark, et al., 2002, p. 150). In the critical realism, reality is stratified into three nested domains: real, actual, and empirical (Bhaskar, 1975; Bhaskar, 1998; Mingers, 2004; Wynn & Williams, 2012).

The real domain includes mechanisms and structures that have causal powers to produce events and experiences (Mingers, 2004; Wynn & Williams, 2012). It is a reflection of complex interactions between open, dynamic systems where specific structures generate certain causal powers or tendencies (often called generative mechanisms) (Mingers, 2004). These generative mechanisms interact, and one may counterbalance another, causing the occurrence or absence of events (Mingers, 2004). The actual domain includes events that may or may not occur depending on the enactment of structures and generative mechanisms. The empirical domain includes events that are experienced or observed (Mingers, 2004; Wynn & Williams, 2012).

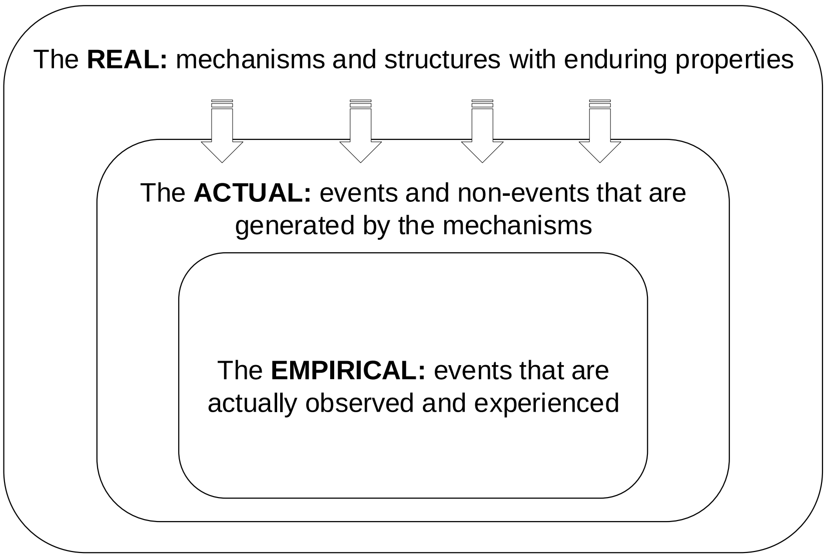


Figure 3.2 CR reality domains (Mingers, 2004).

Many scholars have highlighted the value of using critical realism to examine IS-related issues (Mingers, 2004; Mingers, et al., 2013; Smith, 2006; Volkoff, et al., 2007). For example, Mingers (2004, p. 97) emphasises that:

> *critical realism is important for IS because: (i) CR enables us to take a basically realist stance whilst accepting the major critiques of naive realism; (ii) it addresses both natural and social science and thus encompasses the main domains of IS; and (iii) does potentially fit well with the reality of IS as an applied discipline.*

CR-based studies offer opportunities to examine complex organisational phenomena in a comprehensive manner (Mingers, 2004). CR helps researchers comprehend and explain *"why things are as they are"* and *"to hypothesise the structures and mechanisms that shape observable events"* (Mingers, 2004, p. 100).

A CR study's rationale is to explicate a given set of outcomes by uncovering the hypothesised existence of mechanisms that, once activated, could have generated these outcomes (Bhaskar, 1998; Wynn & Williams, 2012). Given a set of empirical evidence regarding a central phenomenon and context, CR endeavours to answer the question: What must reality be like in order for this outcome to have happened? CR researchers aim to determine the mechanisms that surface from the components of interacting structures to produce the relevant outcomes (Sayer, 1992; Wynn & Williams, 2012).

This thesis uses critical realism mainly because the theory adopted, Archer's (1995) morphogenetic theory, is a theory developed based on critical realism's philosophical foundations. Additionally, in line with Mingers's (2004) statement cited above, this thesis argues that critical realism-based investigations facilitate the examination of EA evolution (1) to explain why EA evolution outcomes are as they are, and (2) to hypothesise the structures and mechanisms that shape these outcomes. Thus, this thesis uses Archer's morphogenetic theory to support the examination of EA evolution after SOA introduction. Chapter 4 justifies the use of this theory.

3.3.2 Iterative Five-Stage Critical Realist Framework

Given that this thesis uses critical realist theory to understand EA evolution, it adheres to critical realism foundations in applying a widely used CR methodological framework (Danermark, et al., 2002). Danermark, et al. (2002, p. 109) provide an iterative five-stage critical realist methodological framework in what is described as a structurable method for conducting critical realism research (Raduescu & Vessey, 2009) (see Figure 3.3). This thesis adopts the framework (1) to guide the overall conduct of the study and (2) to identify the relevant structures, mechanisms, and outcomes of EA evolution.

Stage	Purpose
1. Description	to prepare a description of the phenomenon under investigation
2. Analytical resolution	to differentiate various components of the phenomenon and set up its tentative boundaries.
3. Abduction / theoretical redescription	to redescribe the different components and their relationships using a theoretical framework.
4. Retroduction	to identify the causal powers (generative mechanisms) that may interact to generate the outcomes.
5. Concretisation and contextualisation	to examine how these structures, generative mechanisms and outcomes manifest themselves in a given context.

Figure 3.3 CR-based methodological framework (Danermark et al., 2002)

The following sections provide an overview of the iteratively employed five stages of the framework. The link between these five stages and the phases of the research plan are described in Section 3.5.

3.3.2.1 Description

The first stage, description, is the starting point for recognising an actual event or situation for further analysis (Danermark, et al., 2002; Raduescu & Vessey, 2009). Bygstad and Munkvold (2011, p. 5) define the identification and description of the events as "clusters of observations, which may have been made by the researcher or by the researcher's informants".

3.3.2.2 Analytical Resolution

The second stage, analytical resolution, includes the explication of the composite and the complex research issue by identifying relevant components, aspects, or dimensions (Danermark, et al., 2002; Raduescu & Vessey, 2009). These stage breakdowns research phenomena into their constituent parts. The key components are the objects of the case (e.g., agents, systems, and organisations) (Bygstad & Munkvold, 2011). For instance, Easton (2010) examines a customer relationship management (CRM) implementation case and identifies its main components: the company, the CRM vendor, the government knowledge transfer program, and the exchange relationship. The constituent parts of the phenomenon under investigation are the organisation, its EA, SOA introduction, and SOA/EA integration outcomes. They are the selected boundaries for this thesis.

3.3.2.3 Abduction/Theoretical Redescription

The third stage, abduction (theoretical re-description), reflects the selection of a theory or multiple theories as a lens through which to redescribe and examine a research problem. Danermark, et al. (2002, p. 205) define abduction as a process by which a "particular phenomenon or event is interpreted from a set of general ideas or concepts". It involves redescribing the components of interest from hypothetical conceptual frameworks by using theories about structures and relations (Danermark, et al., 2002; Raduescu & Vessey, 2009) that provide leverage for potential explanations (Wynn Jr & Williams, 2008). Abduction is:

> *used to propose likely theories (i.e., explanations) for actualities identified. In the movement from surface phenomena to a deeper, perhaps non-observable causal thing, the critical realist depends heavily on theory to propose possibilities. Such a perspective is consistent with a deep realism where explanation is not about prediction but about the steady unearthing of deeper levels of structures and mechanisms* (Dobson, Jackson, & Gengatharen, 2013, p. 7).

For example, Volkoff, et al. (2007) employs Archer's (1995) morphogenetic theory to explain the changes of organisational elements after an ERP implementation. Easton (2010) uses an economic-exchange model to redescribe the relationships of a customer relationship management (CRM) implementation case.

3.3.2.4 Retroduction

Stage four, retroduction (or the identification of candidate mechanisms) (Bygstad & Munkvold, 2011; Danermark, et al., 2002), is a "reconstruction of the basic conditions for anything to be what it is, or, to put it differently, it is by reasoning we can obtain knowledge of what properties are required for a phenomenon to exist" (Danermark, et al., 2002, p. 206).

Retroduction involves identifying and elaborating on the generative mechanisms that may interact to generate certain outcomes (Bygstad & Munkvold, 2011; Raduescu & Vessey, 2009; Wynn & Williams, 2012). The principle of retroduction from observed outcomes to the mechanisms behind them is based on CR's focus on explanation (Wynn Jr & Williams, 2008). In retroduction, an unexplained phenomenon is investigated to propose hypothetical mechanisms that, if they exist, offer explanations for that which is to be explained (Mingers, 2004).

3.3.2.5 Concretisation and Contextualisation

Stage five, concretisation and contextualisation, examines how structures and mechanisms manifest in practice (empirical situations). In this stage, the researcher studies the phenomenon in which mechanisms interact at different levels and in different situations (Danermark, et al., 2002). The objective is to interpret the meaning of the mechanisms in a given context and to offer explanations of the observed outcomes (Danermark, et al., 2002; Raduescu & Vessey, 2009). During this stage, the researcher elaborates on the explanatory power of the proposed mechanisms that are described through abduction and retroduction (stages three and four) (Danermark, et al., 2002).

The five stages are described in relation to their application in this thesis in Section 3.5.

3.4 Research Methods

This section introduces the adopted research methods and justifies their selection.

This thesis aims to ensure a rigorously valid understanding is reached regarding EA evolution. This thesis investigates EA evolution (SOA's integration into EA), and, in particular, the structures and their underlying generative mechanisms that generate observed outcomes (EA evolution outcomes). This thesis develops a theoretical model that describes the EA evolution process and explains EA evolution outcomes. It does not provide conceptual findings or recommendations for designing the new architectural elements themselves (nor the modifications of the existing ones), namely artefacts where design science method is appropriate (Hevner, March, Park, & Ram, 2004).

From a critical realist perspective, Sayer (1992, p. 179) states that "qualitative analysis of objects is required to disclose mechanisms". Based on (1) Sayer's (1992) argument, (2) the need to examine a contemporary socio-technical phenomenon (EA evolution) (Wynn & Williams, 2012), and (3) this thesis's exploratory nature, a multi-method qualitative research approach was chosen to investigate EA evolution. This thesis adopted an explorative interview method (Kvale, 1996; Mingers, 2001) and a case study research method (Easton, 2010; Tsang, 2013; Wynn & Williams, 2012; Wynn Jr & Williams, 2008; Yin, 2009) in a sequential manner, the case study being the dominant (Mingers, 2001).

This thesis employed a multi-method approach as suggested by Mingers (2001), Creswell (2003) and Wynn and Williams (2012). Their basic argument for using this approach is that a research study is often not a single event. Rather, it is a process that proceeds through a number of phases. Each phase poses different tasks and issues for the researcher(s) (Mingers, 2001). Creswell (2003) suggests that methods could be combined in a sequential procedure when the researcher's intention is to expand or elaborate on the findings of one method with another method. A researcher could combine multiple research methods such as surveys, interviews, experiments, and case studies (Mingers, 2001). Wynn and Williams (2012) suggest that a methodological triangulation could be achieved through integrating multiple qualitative or multiple quantitative methods that increase the understanding of a certain phenomenon.

3.4.1 Explorative interviews

The interview is an important data-gathering method in qualitative research (Myers & Newman, 2007). It remains the most widely used data gathering method in organisational qualitative research (King, 2004). It is flexible and can be used alone or with other data gathering methods. It is also capable of gathering data of great depth (King, 2004). Kvale (1996) argues that, recently, qualitative interview research is increasingly being used as a ***research method in its own right***. The objective of the qualitative research interview is to understand the research topic from the perspective of interviewees.

Qualitative interview research can be divided into many types. Some of the common interview types are structured interviews, semi-structured interviews, and group interviews (Fontana & Frey, 2000; Myers & Newman, 2007).

- A structured interview is where the interview guide is prepared in advance. All the respondents are asked the same pre-established series of questions. Therefore, the variation is limited in such interviews except in the open-ended questions that are infrequently used. This kind of interview is often used in surveys where the interview could be conducted by many people.
- A semi-structured interview uses a high-level interview guide. Some questions to guide the interview may be prepared beforehand. However, variation is needed in such interviews. It provides a greater breadth of data than the other types. The interview is conducted by the researcher or a member of the research team. This type is widely used in qualitative IS research.
- Group interviews are where many people are interviewed at once by one or more interviewers in structured or unstructured settings. They have usually been associated with marketing research under the name focus groups.

The explorative (semi-structured) interview research method was adopted to enrich the understanding of EA evolution and extend the a-priori model. Creswell (2003) notes that interviews are appropriate to use if the researcher wants to understand the phenomenon in question, if little research has been done, and if the researcher does not know all the important variables to examine.

3.4.2 Case Study Research Method

The case study research method originated in the social science disciplines. It is one of the most extensively used qualitative research methods in information systems research (Benbasat, Goldstein, & Mead, 1987). It investigates an existing phenomenon in its real setting. Yin (2003, p. 13) defines the case study as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly defined".

It is an appropriate method to investigate emerging phenomena in which few previous studies have been conducted. Indeed, Eisenhardt (1989) highlights that the case study method is "especially appropriate in new topic areas" or in areas in which few studies have been conducted (Benbasat et al., 1987). It allows the investigation of IS-related issues in their real settings (Gable, 1994). Benbasat, et al. (1987) believe that case studies are "well-suited to capturing the knowledge of practitioners and developing theories from it". The case study method provides means to answer "how" and "why" questions in order to understand the nature and complexity of phenomena.

The case study method is well suited to conduct critical realist research when studying contemporary socio-technical phenomenon (EA evolution) to uncover the causal mechanisms that generate evolution outcomes (Wynn & Williams, 2012).

Case studies reflect a wide variety of designs (Jensen & Rodgers, 2001; Yin, 2003; Yin, 2009). Yin (2009) classifies case studies into three types: exploratory, explanatory, and descriptive. Exploratory case studies are fundamentally used to answer "what" questions. They aim to develop a relevant hypothesis for further investigation (Tellis, 1997; Yin, 2003). Explanatory case studies are designed to answer "how" and "why" questions. They aim to describe how a phenomenon has occurred, and to establish links between variables or events. Descriptive case studies are generally used to answer "what" questions in the form of "how many" or "how much". They aim to help the researcher gain a rich description and understanding of the phenomenon under investigation (Yin, 2003).

The employed case study is both exploratory and explanatory. It is an exploratory case because it explores SOA's integration into EA. It is also an explanatory study because it uses Archer's (2011, 1982, 1995) morphogenetic theory to explain how SOA integration within EA outcomes have been generated in a specific context. Chapter 4 describes Archer's theory in more detail.

This thesis describes EA evolution and provides plausible explanation of EA evolution outcomes. The case study research method is suitable to answer "how" and "why" questions (Gable, 1994; Yin, 2009), which match this thesis's questions related to EA evolution..

More specifically, the case study method was selected because (1) EA evolution is a new phenomenon and little is known about it, (2) EA evolution is a broad and complex issue, and (3) EA evolution cannot be studied outside the context in which it occurs. Furthermore, the case study method is consistent with critical realism and enables the investigation of the proposed mechanisms and outcomes in their real settings (Easton, 2010; Mingers, 2001; Tsang, 2013; Wynn & Williams, 2012; Wynn Jr & Williams, 2008). It is also useful to explore the developed research model and its generative mechanisms in different contexts.

3.5 **Research Plan**

The research plan is the structure of the research: it is a blueprint that displays the arrangement for the collection, measurement, and analysis of data in a manner that aims to combine research purpose and relevance (Gable, 1994). Figure 3.4 presents this thesis's research plan.

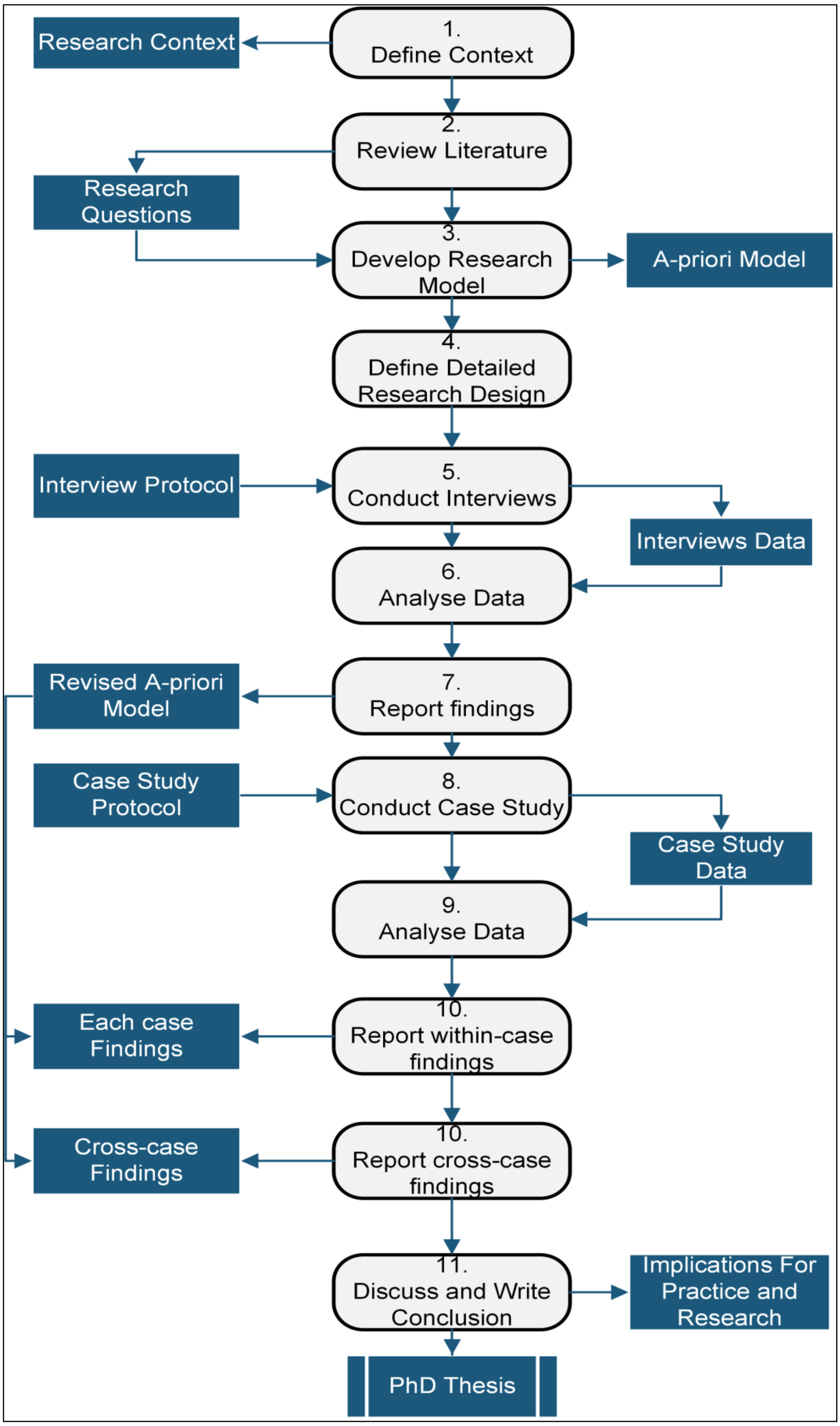


Figure 3.4 Research plan

This thesis's research plan was conducted in alignment with the adopted CR methodological framework described in Section 3.3.2. The research plan phases were aligned with the stages of the methodological research framework (see Table 3.1).

Table 3.1 Mapping between the research plan's phases and the adopted CR framework stages

CR explanatory stages / Research plan phases	Description	Analytical resolution	Theoretical redescription	Retroduction	Contextualisation
Literature review	(shaded)	(shaded)		(shaded)	
Model development			(shaded)		
Explorative interviews		(shaded)		(shaded)	
Multiple-case study					(shaded)

3.5.1 Context and Literature Review Phase

The literature review was undertaken to identify research questions and to determine the research context.

The literature review was conducted across both academic and practitioner information sources to identify the research context, understand the topic under investigation, and identify gaps and research questions. For this thesis, literature about enterprise architecture (EA), EA evolution, Service-oriented Architecture (SOA), and SOA's integration into EA were reviewed (see Chapter 2 for details and results). Following the critical realist framework by Danermark, et al. (2002) described above, the literature review was conducted in light of CR's description, analytical resolution, and retroduction stages. The literature review was conducted to describe SOA's integration into EA (description stage), pinpoint the relevant aspects of it (analytical resolution), and identify potential generative mechanisms (retroduction stage). This phase outcomes provided an initial description of SOA's integration into EA, its boundaries and potential generative mechanisms related to EA evolution.

3.5.2 The Model Development Phase

This phase was also conducted to theorise about SOA's integration into EA (the theoretical redescription). In this phase, candidate theories were identified and a theory to re-describe the components and aspects that were identified in the previous phase were selected. Archer's (1995) morphogenetic theory was adopted to re-describe the literature review findings related to SOA's integration into EA. This phase activities resulted in the development of this thesis's a-priori model (see Chapter 4 for further details).

3.5.3 Explorative Interviews Phase

An explorative semi-structured interviews research method (Kvale, 1996; Mingers, 2001) was selected to refine and extend the a-priori model. Semi-structured interviews were used because they permit an in-depth exploration of the research issues with every participant and to develop an understanding of EA evolution as seen from the independent perspective of each participant. They were conducted to explore SOA's introduction into EA to enrich the understanding of the research problem and to identify the scope of what should be investigated in the following multiple case study phase.

Snowball (chain) sampling, an approach for locating information-rich key informants, was used in this phase (Patton, 1990). It was used to identify participants that lead to other participants (Myers & Newman, 2007; Patton, 2001). Twenty interviews with two types of informants were included in this thesis (EA practitioners, such as chief enterprise architects, and EA consultants) because they held the knowledge and the expertise required for this thesis. Raadt and Vliet (2009) note that enterprise architects are experienced employees who are often highly valued for their knowledge about organisations' structures, systems, processes, and technology. Participants were selected based on their roles (i.e., EA-related) and their involvement in SOA introduction projects. EA consultants were included in the sample because they usually have a broader perspective (e.g., they are often involved in multiple SOA projects) and knowledge that isn't constrained to a single organisation (Chapter 5 details participant information).

An interview protocol is particularly valuable for exploratory studies for several reasons. The protocol encourages researchers to consider the objectives of their study in advance. It also guides data collection endeavour (Yin, 2003). Thus, an interview protocol (see Appendix A) was developed prior to commencing the interview phase to facilitate the comprehensive exploration of EA settings, SOA implementation, and SOA's integration into EA. Research questions, the a-priori model, and the literature review were the primary inputs that helped to formulate the interview protocol. The protocol consists of multiple sections with questions about demographic information, EA, SOA, and SOA's integration into EA.

3.5.4 Interview Data Analysis

The qualitative data analysis process includes analysing textual data, coding concepts in the text, and then categorising the codes into themes. Miles and Huberman (1994, p. 56) define analysing qualitative data as "review[ing] a set of field notes, transcribed or synthesized, and dissect[ing] them meaningfully, while keeping the relations between the parts intact."

One of the commonly accepted qualitative analysis techniques is thematic analysis (Boyatzis, 1998; Fereday & Muir-Cochrane, 2006; Miles & Huberman, 1994). Boyatzis (1998, p. vi) defines thematic analysis as:

> *a process for encoding qualitative information. The encoding requires an explicit code. This may be a list of themes, a complex model with themes, indicators, and qualifications that are causally related; or something in between these two forms.... Themes may initially be generated inductively from the raw information or generated deductively from theory and prior research.*

Thematic analysis guides the identification and analysis of themes in data. It helps organise and describe the data, which explicates many facets of a study while ensuring consistency and reliability (Boyatzis, 1998). According to Ryan and Bernard (2000, p. 780), themes are "abstract (and often fuzzy) constructs that investigators identify before, during, and after data collection". Thematic analysis involves becoming familiar with the data, and coding and identifying themes (Braun & Clarke, 2006) (see Chapter 5 for applying thematic analysis guides in Chapter 5).

Coding is a central categorising strategy for fracturing data and rearranging it into categories that assist in comparison and for facilitating the development of theoretical concepts (Maxwell, 2005). Boyatzis (1998, p. 63) describes codes as the most essential element of the raw data that can be used in a meaningful way to describe the phenomenon. Beekhuyzen, Nielsen, and Von Hellens (2010) argue that qualitative research is more than just coding. Other strategies, mostly informal ones, are used as well, such as reading and thinking about the transcripts and developing and evolving coding categorisations.

In this thesis, thematic analysis was employed as a qualitative data analysis technique. In particular, a hybrid approach of inductive and deductive thematic analysis was adopted (1) to improve the rigour of the study (Fereday & Muir-Cochrane, 2006), and (2) to generate richer insights (Chiasson, Germonprez, & Mathiassen, 2009; Fereday & Muir-Cochrane, 2006). Such an approach is particularly useful in novel settings where extant work is limited (Fereday & Muir-Cochrane, 2006), and it is also an accessible and theoretically flexible approach for qualitative data analysis (Braun & Clarke, 2006). Deductive analysis, in which theory is used to guide the analysis, was conducted using the a-priori model. Inductive analysis was further used to identify codes and themes from the data (Chiasson, et al., 2009) to extend the identified generative mechanisms and the evolution outcomes.

All the interviews were transcribed, read, and imported to qualitative analysis software (Nvivo) to prepare the data for analysis. Then, the analysis was conducted following thematic analysis. Interview data was analysed in an iterative process of inductive and deductive analysis to create, identify, refine, and update codes and themes (see Chapter 5 for the detailed process).

The data was further analysed using the retroduction approach (which Danermark, et al. (2002) and Wynn and Williams (2012) suggest) using the adopted theory. The retroduction approach involves identifying and elaborating on the generative mechanisms that may interact to generate evolution outcomes. It is achieved by using the morphogenetic theory three analytical phases (conditioning, interaction, and elaboration) to organise the findings.

As a result of analysing the data in the interview phase, the a-priori model was extended. In other words, the identified generative mechanisms and evolution outcomes were refined and extended through identifying more generative mechanisms and evolution outcomes.

3.5.5 Multiple Case Study Phase

This multiple case study design phase followed the interview phase to contextualise the developed model (the contextualisation stage of the adopted CR methodological framework). In this thesis, the identified structures and their generative mechanisms were examined using case studies to further understand SOA's integration into EA and the relationship between the proposed generative mechanisms and evolution outcomes. This stage helps the researcher understand the manifestation of the proposed generative mechanisms in a given context. It also examines their explanatory effectiveness in that context (Wynn Jr & Williams, 2008).

3.5.5.1 Case Study Design

The case study design may comprise a single-case design or multiple-case design (Yin, 2009), which Figure 3.5 shows. Yin (2009) and Benbasat et al. (1987) suggest the use of a single case study when the case is a revelatory one and where the researcher has access to a case that was previously inaccessible to scientific investigation. It is also appropriate when the case is critical to confirm or challenge a theory, or when it is a unique or an extreme case. On the other hand, multiple-case designs are advantageous when the researcher's purpose is to describe a phenomenon, or to build or test a theory (Benbasat, et al., 1987; Yin, 2003; Yin, 2009).

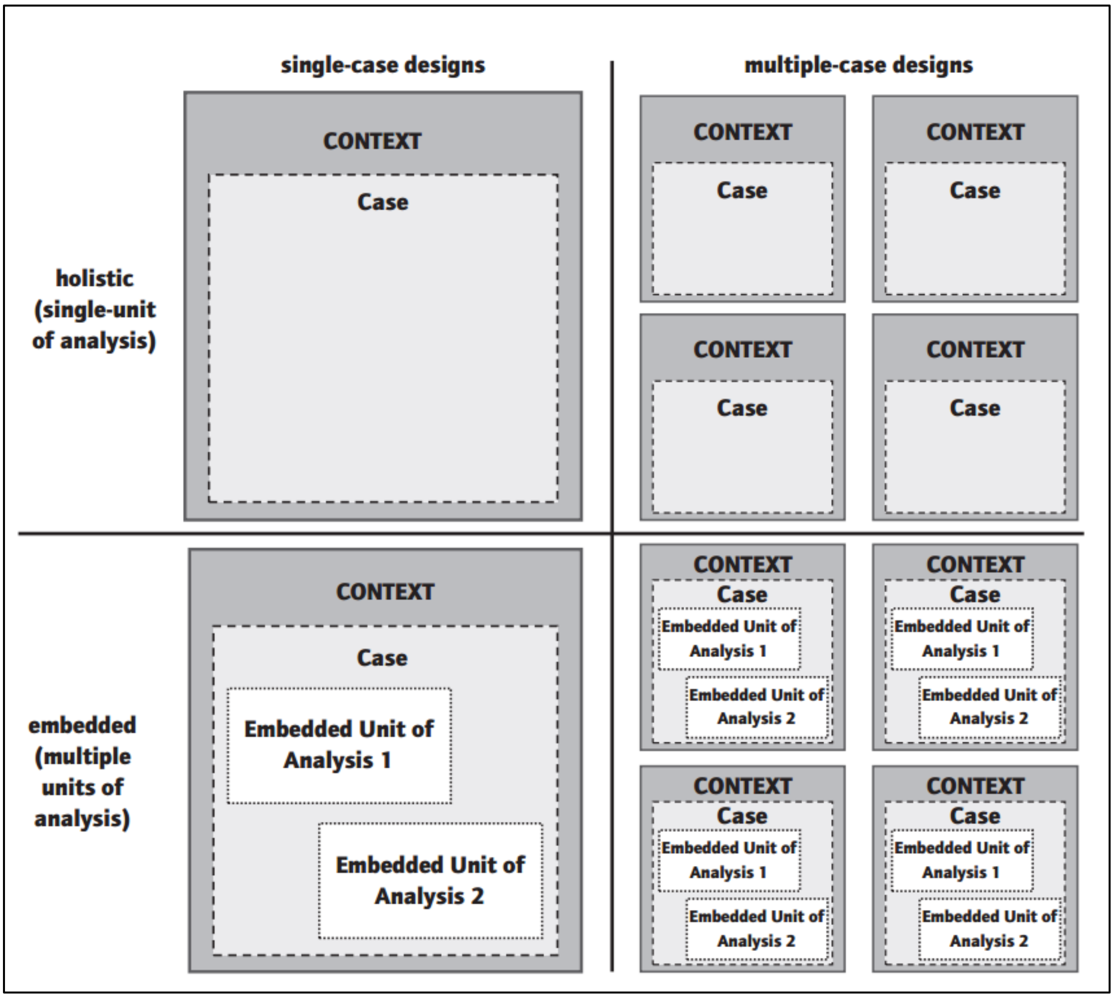


Figure 3.5 Case design and units of analysis (Yin, 2009, p. 46)

Furthermore, Jensen and Rodgers (2001) provide a typology of case study design; for example, snapshot (cross-sectional) case studies where one entity is studied at one point in time. Case studies also can be longitudinal, which provide a different type of information than the snapshot study. It is a time-ordered analysis of events that happen during a period of an entity's history (Jensen & Rodgers, 2001). Street and Ward (2010) further classify longitudinal case studies into three types: concurrent, retrospective, and historical. Street and Ward (2010, p. 825) argue that:

> *these three types of longitudinal case studies differ along two dimensions: (1) whether the events being studied have already occurred and (2) whether researchers have access to informants who were involved in the events or phenomena being studied.*

Both concurrent and retrospective case study designs usually include informant interviews in the data set, while historical designs do not. In both historical and retrospective designs, the events have already happened and the outcomes are known. However, in concurrent designs, the outcomes are not yet known during the first data-collection round. Retrospective designs are often more efficient than concurrent designs because the latter requires investigators to wait until the passage of time creates the proposed changes in processes or variables under investigation. Retrospective designs are advantageous due to the fact that data are collected from multiple prior periods all at once.

A retrospective case study has three common aspects: (1) data are collected after the significant event has already happened, (2) investigators have access to both first-person accounts and archival data, and (3) the outcomes of that event are known at the time of data collection. These outcomes are included in observations and help the investigator to construct the timeline that connects the event and outcomes (Street & Ward, 2010). MacQuarrie (2010) argues that concurrent longitudinal approaches allow researchers to see changes as they take place, and offer rich empirical evidence for the work. However, they are time and resource intensive. The retrospective case study has the advantage of being at the end of a process and looking back. It is also less expensive in terms of time and resources (MacQuarrie, 2010; Street & Ward, 2010).

This thesis adopted a retrospective multiple-case design. First, a retrospective case design was chosen over the snapshot design because this thesis examines EA evolution (time-based) and because its adopted theory considers a temporal dimension of EA evolution. The thesis seeks to understand EA evolution process and the event (SOA's introduction) that generated EA evolution outcomes. Therefore, it was necessary to choose a longitudinal design. A concurrent longitudinal design takes time and would require significant amounts of resources to investigate EA evolution before, during, and after SOA's introduction, which could take more than two years, as seen in the two conducted case studies (Chapters 6 and 7). Therefore, due to the limited time and resources allowed in a PhD study, a retrospective design was chosen. Retrospectively analysing historical data provides rich phenomena for improved understanding of the context (Becker & Burke, 2012). Retrospection enables researchers to consider the past where the creation of meaning of what is happening at that moment is a retrospective process that arises from the awareness of what has previously occurred (Tansley & Watson, 2000). The retrospective analysis was needed in order to explain how the evolution outcomes have been generated, which many CR authors (e.g., see Mingers, 2004; Ryan, Tähtinen, Vanharanta, & Mainela, 2012) suggest.

Second, a multiple-case design was chosen because of its inherent advantages, such as rigour and theory generalisation (Benbasat, et al., 1987; Eisenhardt, 1989; Miles & Huberman, 1994; Yin, 2003; Yin, 2009). This design was selected in this thesis to explore and understand EA evolution in different contexts. It also enables the search for cross-case patterns and themes, and supports the comparison of the differing observations to advance propositions in various settings.

3.5.5.2 Unit of Analysis

Another aspect related to the case study design is identifying the unit of analysis. The unit of analysis is a fundamental aspect in qualitative research studies: it defines what the case is (Yin, 2009). The unit of analysis is important because it determines the sources of evidence and the boundaries of the evidence gathering. The unit of analysis could be single or embedded (multiple units of analysis). Yin (2009) suggests five possible units of analysis: individuals, decisions, implementation processes, programs, and organisational change. Easton (1998) proposes a form of embedded unit of analysis based on time; that is, the case must have a longitudinal component and should be seen as a series of cases (embedded) depending on the period of time being investigated and described. As such, in this thesis, the overall unit of analysis is EA evolution due to SOA's introduction, which includes embedded units of analysis. These embedded units of analysis are the investigation of EA prior to SOA's introduction, a detailed examination of SOA's introduction, and the outcomes of SOA's integration into EA. The aggregation of analyses of the sub-units composed the analysis of EA evolution (here: SOA's integration into EA).

3.5.5.3 Data Collection Methods

Case study research employs multiple methods of data collection (e.g., interviews, documents, and archival analysis) to collect data from one or more entities (Benbasat, et al., 1987; Yin, 2003; Yin, 2009). To ensure data quality, the evidence gathering was triangulated using a combination of archival analysis, interviews, and internal documents. The gathered evidence covers one morphogenetic cycle of SOA integration within EA, which is this thesis's scope. The scoping of such morphogenetic "breaks", which indicate the start and the end of a cycle, "is the business of any particular investigator and the problem in hand" (Archer, 2011, p. 66). Thus, SOA's integration into EA morphogenetic cycle and its three analytical phases are determined according to a stability-change-stability analysis (Njihia, 2008; Trosper, 2005).

A researcher needs to tentatively adopt a set of periods, distinguished by times of stability and times of change, in order to identify the analytical morphogenetic cycle boundaries (Njihia, 2008; Trosper, 2005), where each morphogenetic cycle (see Figure 3.6) signifies a substantial change in the structure (Njihia 2008). Time (T1) represents the beginning of the analysis of one morphogenetic cycle and Time (T4) is the end of it.

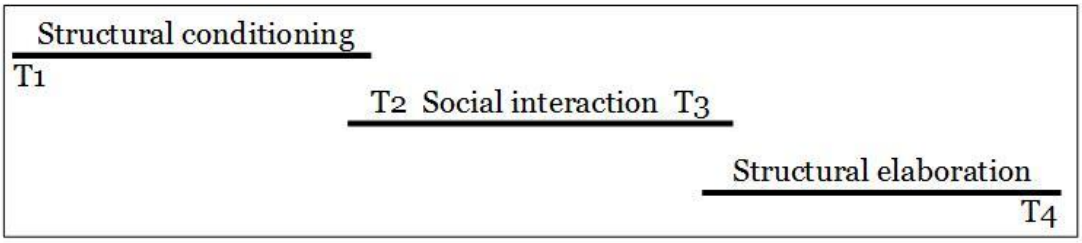


Figure 3.6 A morphogenetic cycle boundaries

- The time point (T2) represents the beginning of the interaction, and (T3) is the end of it
- It is valuable to broaden the temporal frame to the state of structure at time (T1).
- Examining the interaction that took place between T2 and T3 is focused to search explicitly for what led to those features at time (T4) (Archer, 2011), and

- T4 is the end of a cycle and the beginning of a new morphogenetic cycle. In other words, it becomes the architectural conditioning phase for the coming change cycle.

Chapters 6 and 7 detail more fully this thesis's data sources.

3.5.5.4 Sampling Strategy

Sampling signifies that researchers make strategic choices about with whom, where, and how their research is conducted. Cases may form a purposeful but non-probability sample. Patton (1990) suggests that such a purposeful sample has a logic and power and that it offers richer information. The rationale of selecting purposeful sampling is to choose cases that provide rich information (in other words, cases that offer a great opportunity to learn about issues of high and whose examination will help answer a study's questions of interest) (Patton, 1990).

The researcher selects individuals and sites for study because they can purposefully inform the understanding of research problems and research questions (Creswell, 2003; Maxwell, 2005). Patton (1990) provides different types of purposeful sampling such as criterion sampling, snowball (chain) sampling, maximum variation sampling, and theory-based sampling. Yin (2003) argues that case sampling has to follow replication logic. A literal replication produces similar findings for predictable reasons, and a theoretical replication produces contradictory findings for predictable reasons (Yin, 2003). Benbasat et al. (1987) suggest that site selection should be based on organisational or technologies characteristics when research is about organisation-level phenomena.

In this thesis, the case sites were selected using purposeful sampling and, in particular, criterion sampling. The rationale of using criterion sampling is to choose cases that meet some predetermined criterion of significance (Patton, 1990). From a critical realist perspective, choosing cases that share a structure of interest but are different with respect to other aspects is best. In order to understand the interaction between structures and mechanisms and their outcomes, Danermark, et al. (2002, p. 105) suggest that researchers need to select "a number of cases which are all assumed to manifest the structure she wishes to describe, but which are different in other aspects”. Moreover, in accordance with both critical realism and qualitative principles, the type of purposeful sampling utilised is that of varied cases. These are cases that indicate but are not representative of a variety of different EA evolution outcomes. Danermark et al. (2002, p. 170) describes the critical realism basis for choosing the varied cases form of purposeful sampling by saying:

> *The purpose of selecting this type of case is to attain information about the importance of various conditions for producing the particular phenomenon under investigation. We analyse how mechanisms operate under different conditions.*

Selecting a case generally reflects the existence of outcomes of the phenomena a researcher is attempting to understand (Wynn & Williams, 2012). In this thesis, cases were not studied in order to prove a pattern of events (e.g., to prove that, if many cases performed something in certain ways, that those ways would be the best EA evolution options). Thus, a large number of varied cases were not necessary nor were cases that needed a control group against which to measure their outcomes because this study occurred in an open system (Danermark, et al., 2002; Wynn & Williams, 2012; Yin, 2009). In critical realism, an open system’s dynamic shifts the focus onto pinpointing the tendency of mechanisms to act in a specific contextual settings at a specified time (Wynn & Williams, 2012). Nonetheless, to ensure a variety of EA evolution cases, particular sampling criteria were employed. This thesis endeavoured to find organisations that integrated SOA into EA in different ways in order to gain a wider and deeper understanding of what happened under the surface in CR terms to generate the observed EA evolution outcomes.

Two cases (Dubai customs and Businesslink) were selected using a purposeful (criterion-based) sampling strategy based on the criteria that Table 3.2 summarises. Two cases are considered appropriate to provide enough information for generating explanatory propositions (Sayer, 1992).

Table 3.2 Case selection criteria

Criteria
The evolution outcomes were different
The organisation manifested the structure of interest (EA)
The organisation implemented SOA
The people who were/are involved with EA and SOA were accessible
The internal documents and archival analysis about EA and SOA were accessible
The significant event already happened (SOA implementation)
The outcomes of SOA's integration into EA were known at the time of the data collection

Building on Danermark, et al. (2002) previous arguments, the selected cases had an enterprise architecture and needed to have implemented SOA. Theses cases were different in terms of their EA's maturity and their SOA's characteristics.

Moreover, three criteria related to the longitudinal (retrospective) case design were adopted. First, the significant event (SOA's introduction) already needed to have happened and so the data collection started after the event. Secondly, in both cases, the investigator needed access to both first-person accounts and archival data relevant to SOA and EA. Thirdly, the outcomes of the SOA's introduction needed to be known at the time of the data collection. The outcomes (from presentations available on the Internet) were known, and they were used as the main rationale to contact these organisations. Both cases were identified through online documentation and presentations at practitioner-oriented conferences that had high-level results of SOA's integration into EA.

Participants in each case site were selected based on their role in their organisation's SOA integration into EA. SOA and EA team members and managers were selected as appropriate participants because they are involved and knowledgeable about the phenomenon under investigation (see Chapter 6 for more information).

A case protocol was developed to guide the data collection (see Appendix B). The case protocol includes the interview protocol, which defines the structure of the overall interview effort and guides the researcher during interviews. In this thesis, the semi-structured interview protocol, which was employed in the previous phase, was extended and used.

3.5.5.5 Case Study Data Analysis

The analysis of the case study data followed the same strategy used in the previous explorative interview phase; namely, the thematic analysis technique. The analysis was largely similar to the process that was employed to analyse the explorative interview data. The case analysis was mostly deductive and used the theoretical model that was revised following the explorative interview phase. The data was further analysed using the retroduction approach using the morphogenetic theory's three analytical phases (conditioning, interaction, and elaboration) as suggested by Danermark, et al. (2002) and Wynn and Williams (2012) suggest. Both within-case analysis (Chapters 6 and 7) and cross-case analysis (Chapter 8) were conducted. Sections 3.5.5.6 and 3.5.5.7 address such analyses.

3.5.5.6 Within-case Analysis

Within-case analysis helps researchers to cope with data's complexity and richness (Eisenhardt, 1989). As Eisenhardt (1989) and Yin (2003) note, researchers use within-case analysis to become strongly familiar with each case as a stand-alone unit prior to conducting cross-case analyses. This involves going through all the data, removing whatever is unrelated, and bringing together what is considered important. The idea is to let the most important observations emerge from gathered data and reduce the data's size.

In this thesis, within-case analysis involved detailed case study write-ups for each site. Results from within-case analysis, in which the thematic analysis and the retroduction (three analytical levels) were used to assist in data analysis, are presented in Chapters 6 and 7.

3.5.5.7 Cross-case Analysis

Applying cross-case analysis to cases usually conveys more robust outcomes and help to reinforce the findings (Yin, 2003). This thesis uses cross-case analysis to look for similarities and differences between cases by comparing several categories at once. According to Yin (2009), one way of doing cross-case syntheses is to build tables that exhibit data from individual cases according to a consistent framework. Thus, to facilitate the cross-case analysis, both of the cases in this thesis were written in the same format. The two cases were compared along the three analytical levels of Archer (1995) morphogenetic theory (see Chapter 8 for more details).

3.6 **Research Quality**

Scholars perceive qualitative studies differently from quantitative ones (Creswell, 2009; Yin, 2003). Some qualitative researchers argue that the measures used for evaluating quantitative research are not usually appropriate for qualitative research (Lincoln & Guba, 1985). As a result, many qualitative researchers have developed their own measures. For example, Lincoln and Guba (1985) use validity measures such as applicability, consistency, and neutrality, while Yin (2003) uses construct validity, internal validity, external validity, and reliability. Thus, this thesis used several the tactics recommended by well-known qualitative scholars to enhance its quality (Creswell, 2009; Yin, 2003) because general qualitative quality aspects are still relevant in CR studies (Easton, 2010; Henfridsson & Bygstad, 2013). Because this research is grounded on the work of Yin (2003) and Creswell (2009), the quality measures used in this thesis are presented under the dimensions of validity and reliability.

3.6.1 Reliability

Reliability refers to the consistency of the researcher approach across different projects and different researchers (Creswell, 2009; Yin, 2003), and demonstrates that the conduct of a study can be repeated with the same outcomes if the same procedures and instruments are used (Yin, 2003).

Therefore, to improve the consistency of the findings and to ensure the rigour and thoroughness of the research process and findings, this thesis employs reliability recommendations that Yin (2003) and Creswell (2009) suggest. Table 3.3 summarises the adopted reliability measures.

Yin (2003) suggests the use of a case study protocol to enhance reliability. In this thesis, a comprehensive case study/interview protocol was developed to provide clear guidance for the data collection process and to ensure the consistency of the collected data. Another measure undertaken to ensure better rigour and thoroughness was to maintain a case study database, which Yin (2003) suggests. In this thesis, collected data was organised and stored to allow for later retrieval. All transcriptions and supporting documents were kept in folders to facilitate access to the raw data at any time. NVivo was used as a repository for all the raw and analysed data. Creswell (2009) also suggests that researchers need to document their procedures in detail, improve their transcription to avoid obvious mistakes, and use a consistent definition of codes (Creswell, 2009). In this thesis, a detailed interview protocol and database (using NVivo) were employed during the two empirical phases, the interviews and case studies, to improve reliability. In addition, interviews were transcribed, read, and obvious mistakes corrected. Further, the codebook (codes and their definitions) was developed before the beginning of the analysis and updated as the analysis progressed. However, the main researcher was the only coder involved in the coding.

Table 3.3 Employed reliability measures

Technique	Application in this thesis
Use of case study protocol (Yin, 2003) (see Appendix A and B)	A comprehensive case study protocol and an interview protocol were developed to provide clear guidance for the data collection process and to ensure the consistency of the collected data.
Maintain a case study database (Yin, 2003) (see Chapters 5, 6, and 7).	In this thesis, the data was organised and stored to facilitate later access. All transcriptions were kept in folders to facilitate raw data access at any time. NVivo was also used as a repository of all the raw and analysed data.
Document the procedure in details (Yin, 2003)	In this thesis, the data collection and analysis for both the interviews and case studies were described in detail.
Accuracy of the transcripts of obvious mistakes (Creswell, 2009)	In this thesis, the transcripts were read and corrected if there were obvious mistakes such as spelling or transcribing errors by listening to the audio recording.
Consistent definition of codebook codes (Creswell, 2009) (see Chapter 5)	This thesis used a consistent codebook that has the name of the code and its definition during the analysis. Any new codes or changes of the existing code definition were updated in the codebook.

3.6.2 Validity

According to Creswell (2009), validity is based on determining whether findings are accurate from the researchers', participants', and readers' points of view. There are many strategies to increase the validity of a qualitative study. Some of them are frequently used and easy to apply, and some are infrequently used and difficult to implement. Creswell (2009) suggests the use of validity strategies such as triangulation, the obtaining of information from different sources, the use of rich descriptions to convey the findings, the use of peer debriefing by involving people who review and ask questions about the study, and member checking. Yin (2003) also suggests the use of multiple sources of evidence in order to realise an inclusive perspective on what happens in reality and achieve triangulation (Yin, 2003). This thesis applied two forms of triangulation: data and method triangulation (Yin, 2003). Data triangulation was achieved by collecting information from multiple sources in order to corroborate a fact or phenomenon. Method triangulation was achieved by applying a multi-method approach, including explorative interviews and a multiple-case study. The data collection methods included interviews, online reports and obtained internal documents to examine SOA's integration into EA. Triangulation during the case study phase was achieved by interviewing many participants from the same organisation and obtaining other evidence such as documents, presentations, and meta-models. A huge amount of data was collected during the explorative interview phase and the case study phase. The transcription of the collected data during the explorative interview phase was over 300 pages. Further, the transcription of the interviews (200 pages each case) and the obtained documents during the case study phase was very large. Thus, and due to confidentiality agreement signed with the participants, the obtained data was not made available in the thesis for other researchers.

Creswell (2009) also suggests using rich description to express findings. This may transport readers to the setting in question and give the discussion an aspect of a shared experience. This thesis presented rich descriptions. Creswell (2009) argues that using peer debriefing enhances the accuracy of the research. This involves locating a person (peer debriefing) who interrogates and reviews the qualitative study so that the description will resonate with people other than the main investigator. This thesis employed peer debriefing by involving the supervision team (three people) of the researcher to review and ask questions about the procedures and the findings. This thesis also used a member-checking validity measure to check findings for validity. Each case study's final findings were sent to participants so that they could comment on the findings and their accuracy. Table 3.4 shows a list of the techniques employed to increase the validity of this thesis.

Table 3.4 Employed validity measures

Technique	**Description**	**Application in this thesis**
Triangulation (Creswell, 2009; Yin, 2003)	Triangulate different data sources for information by examining evidence from the sources and using it to build a coherent justification for themes.	Triangulation was achieved through the collection of data using multiple methods such as interviews, archival analysis, and document analysis.
Rich and thick description(Creswell, 2009)	Use rich description to convey the findings. This may transport readers to the setting and give the discussion an element of a shared experience.	This thesis used rich descriptions to enrich the description of the settings of both the interview phase and the case study phase.
Spend prolonged time in the field (Creswell, 2009)	Spend prolonged time in the field. In this way, the researcher develops an in-depth understanding of the phenomenon under study and can convey detail about the site and the people that lends credibility to the narrative account.	In this thesis, the researcher spent one week at each case study site. The researcher also spent two to three weeks prior to the conduct of the two case studies gathering information available online about the two organisations. The researcher also emailed after the visit for further clarification or information.
Peer debriefing (Creswell, 2009)	Use peer debriefing to enhance the accuracy of the account. This process involves locating a person (peer debrief) who reviews and asks questions about the qualitative study so that the account will resonate	In this thesis, three supervisors of the researcher were involved in reviewing and asking questions about the process of the analysis and the reported data. The researcher also had three consultation

	with people other than the researcher.	sessions with the qualitative writing support consultant at QUT to review the process and the reported data.
Member checking (Creswell, 2009)	Use member checking to determine the accuracy of the qualitative findings through taking the final report back to participants.	In this thesis, the findings of the case studies (chapters) were sent back to case study partners and they provided their comments on the final drafts of the findings.

3.7 **Chapter Summary**

This chapter discusses the philosophical and methodological aspects of this thesis. It introduces critical realism, which was adopted due to the fact that the chosen theory—Archer's (1995) morphogenetic theory—was developed based on CR foundations. Critical realism was appropriate for this thesis, because it seeks to comprehend the underlying reasons for how and why EA evolves differently after SOA has been introduced. Moreover, the chapter provides an overview of the critical realist methodological framework, which was used to direct this thesis's overall conduct.

Furthermore, the chapter covers and justifies the selection of a qualitative research approach to answer the research questions. Such an approach is selected due to (1) the need to examine a contemporary socio-technical phenomenon—EA evolution, (2) the explorative nature of the thesis, and (3) the need to reveal the relevant generative mechanisms. Two specific research methods were chosen and justified: explorative interviews and case studies. The explorative interviews were used to refine and extend the a-priori model. The case study phase was employed to contextualise the developed model in two contexts.

In addition, the chapter details the research plan, which comprises four main phases: the literature review, the a-priori model development, the explorative interview phase, and the multiple case study phase. The research plan covers issues such as data analysis, sampling, case study design, and the unit of analysis. Finally, the chapter highlights the chosen reliability and validity measures to maintain the thesis's quality.

Chapter 4: Research Model

4.1 Introduction

Following the theoretical redescription (abduction) stage of the methodological framework that Danermark, et al. (2002) expound (see Chapter 3), this chapter proposes that Archer's (1995) morphogenetic theory can provide a useful foundation to study EA evolution. This chapter derives an a-priori model based on Archer's morphogenetic theory to theorise about EA evolution in a field that often lacks solid theoretical groundwork. Archer's morphogenetic theory is used as an analytical approach to distinguish the architectural conditions under which SOA is introduced, to study the relationships between these conditions and SOA introduction, and to reflect on EA evolution outcomes (elaborations) that then take place.

The chapter progresses as follows. Section 4.2 presents this thesis's theoretical foundations. Particularly, it discusses Archer's (1995) morphogenetic theory and its value for this thesis. Section 4.3 introduces the a-priori model of this thesis using Archer's morphogenetic theory and the literature review's findings. Section 4.4 summarises the chapter.

4.2 Archer's Morphogenetic Theory

As Chapter 3 describes, one of the critical realist methodological framework's main stages is theoretical redescription (abduction), which involves selecting a theory about structures and relationships to redescribe the literature review findings (Danermark, et al., 2002). Thus, candidate theories are those that discuss structure and agency and their interactions (Danermark, et al., 2002).

This thesis views EA evolution as an interaction between existing structural settings (existing EA) and the action of introducing SOA, which results in EA evolution outcomes.

Archer's morphogenetic theory (Archer, 1979; 1995) is a critical realist explanatory theory for change. It is an analytical approach for examining the structuring of systems over time. Similar to other critical realism (CR) theories, it emphasises structures, actions, generative mechanisms, and outcomes (Archer, 1982; Bhaskar, 1975). Archer (1995) notes that her model provides explanatory foundations that acknowledge and incorporate: (1) pre-existing structures that have generative mechanisms, (2) their interplay with other objects possessing generative mechanisms, and (3) outcomes arising from the interaction between the above, which occur in an open system.

The theory conceptualises the relationship between agency (action) and structure to understand the nature of change. Archer's theory rejects all forms of structure and agency conflation because they ignore the autonomy and independent effects of structure and agency (Archer, 1995; Gimenez, 2007). Archer's theory reflects her argument about "analytical dualism" when dealing with structure and agency. Structure and agency are held apart analytically in a dualism rather than a conflationary duality. A detailed investigation of the interaction between structure and agency in social situations is a complex undertaking. Therefore, this thesis uses analytical dualism, whereby structure and action are analytically separated, to investigate their interaction and the outcomes (Archer, 1995; Cuellar, 2010; Dobson, Myles, & Jackson, 2007).

There is a continuous cycle of interaction between structure and agency. The methodological significance of this separability of structure and action is the examination of the continuing interaction between structure and action (Elder-Vass, 2007). Archer states:

> *Fundamentally the morphogenetic argument that structure and agency operate over different time periods is based on two simple propositions: that structure necessarily pre-dates the action(s) which transform it; and that structural elaboration necessarily postdates those actions* (1995, p. 76)

The morphogenetic perspective is both dualistic and sequential. It includes time in the analysis in the form of the morphogenetic cycle. The morphogenetic perspective:

> *is not only dualistic but sequential, dealing in endless cycles of – structural conditioning/social interaction/structural elaboration –*

> *thus unravelling the dialectical interplay between structure and action* (Archer, 2010, p. 228)

The link between structures and agency occurs in a morphogenetic cycle that defines how the structural changes occur (Archer, 1995; Cuellar, 2010). Cuellar (2010, p. 41) states that "previous cycles have created a particular set of existing structures and distributions of resources as the result of prior cycles which condition the actions of existing agencies".

Neither action nor structure solely determine outcomes (Elder-Vass, 2007). The structural elaborations (changes) are co-determined by the conditional influence of existing structures together with the causal powers (generative mechanisms) of the action. The outcomes are due to the interplay between the two sets of generative mechanisms related respectively to structure and action (Archer, 1995).

The morphogenetic cycle has three analytical phases: structural conditioning, social interaction, and structural elaboration. It denotes the interactions between structure and action and their operations over analytically different time periods (see Figure 4.1). According to Archer (1995), every morphogenetic cycle differentiates between three broad analytical phases which comprise: "(a) a given structure (a complex set of relations between parts), which conditions but does not determine (b), social interaction". The social interaction also influenced by agents' orientations and "in turn leads to (c), structural elaboration" (Archer, 1995, p. 91). Conditioning, interaction, and elaboration are continuous through time but, as an analytical tool, dualism is employed to let the researcher cut into reality and project cycles forwards and backwards. Archer (1995, p. 168) notes that:

> *Although all three levels are in fact continuous, the analytical element only consists in breaking up the flows into intervals determined by the problem in hand: given any problem and accompanying periodization, the projection of the three levels backwards and forwards would connect up with the anterior and posterior morphogenetic cycles.*

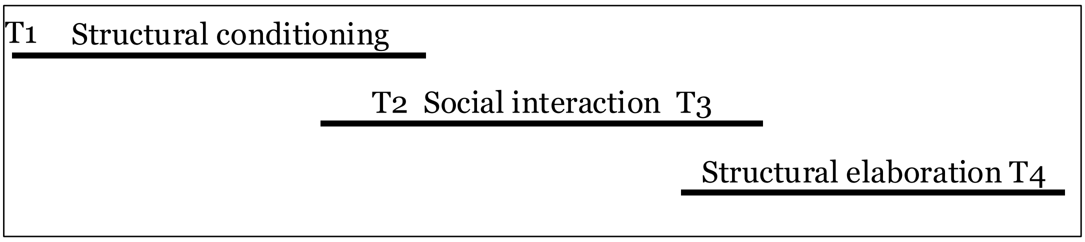


Figure 4.1 Archer's morphogenetic theory (Archer, 1995)

Archer argues that the analysis must start at time (T1) that is to include the structural conditioning that is formed by previous actions, not at the time of (T2) when the social interaction takes place. It is important to distinguish between the structural conditions under which social action takes place, to investigate the interplay between those conditions and the action that occurs, and to understand the elaborations that then happen (Mutch, 2010). Sections 4.2.1 to 4.2.5 further discuss these three analytical levels in general. Section 4.3 discusses the levels as they relate to this thesis.

4.2.1 Structural Conditioning

Phase one of the morphogenetic cycle is structural conditioning. Previous morphogenetic cycles have formed a particular set of existing structures and distributions of resources that condition the actions of existing agents (Archer, 1995; Cuellar, 2010). It is the result of previous agents' actions, and represents the conditions in which the existing agents find themselves (Archer, 1995; Cuellar, 2010; Volkoff, et al., 2007). These conditions affect actions. They enable certain actions and make others difficult. They also may form an opportunity cost for pursuing certain actions (Cuellar, 2010). This structural conditioning could take the place of more or less material structures, such as organisations, markets, or ideas (Mutch, 2002).

Structures are pre-existent, independent, and causally efficacious (Archer, 1995; Gimenez, 2007). Structure is defined as the "set of internally related objects or practices" (Sayer, 1992, p. 92) that comprise the real entities we seek to examine in a particular contextual situation (Wynn & Williams, 2012).

Archer (1995) argues that structure rationally precedes the action(s) that may transform it, and structural elaboration rationally follows those actions. Structures have properties that allow them to influence the world around them (Archer, 1995; Cuellar, 2010). The structure could be part of another larger structure, and it even may include smaller substructures (e.g. a national market system, a single organisation, or even smaller non-social structures) (Sayer, 1992; Wynn & Williams, 2012).

4.2.2 Social Interaction

The second analytical level is social interaction between T2 and T3. During social interaction, actors engage with the pre-existing conditions (Archer, 1995). Social interaction starts when one or more actors decide to make an effort to cause change or maintain the current situation (Cuellar, 2010). At this level, interaction is enabled or constrained by pre-existing conditions. It is also influenced by agents' orientations, interests, and interpretations (Archer, 1995; Danermark, et al., 2002) and resources (Cuellar, 2010).

Archer (1995) defines agency as a concept that encompasses individual actors, collectivities of people, and organised groups such as corporate agents. Agency "involves real actions by real people" (Archer, 1995, p. 258). She differentiates between primary and corporate agency. At any given time, primary agency includes those who neither express interest nor organise for their strategic pursuit, while the corporate agency shapes the context in which all actors operate.

Agent's actions are continual and necessary to both the continuance and further elaboration of a system. In relation to agency, the investigator needs to recognise what actions are undertaken, by whom, and what the outcomes of these actions are (Morton, 2006). There are no static points and agents actions are continuously occurring. Actors might be unaware of the conditions under which they are taking action, but this does not eliminate the impact of those conditions (Cuellar, 2010).

4.2.3 Structural Elaboration

The third analytical level is the structural elaboration phase, either reproduction or transformation of pre-exiting structure. Structures may stay as they are with no changes (morphostasis) or may change (morphogenesis)

(Archer, 1995; Archer, 2010; Gimenez, 2007). Archer (1982) uses morphostasis (reproduction) and morphogenesis (transformation) to describe the process of change. She notes that:

> *Morphostasis refers to those processes in complex system-environment exchanges that tend to preserve or maintain a system's given form, organization or state. Morphogenesis will refer to those processes which tend to elaborate or change a system's given form, structure or state* (Archer, 1982, p. 480)

This phase represents the outcomes of the interplay between pre-existing structures and other objects (Archer, 1995). The time of structural elaboration (T4) denotes that pre-existing structures are transformed or reproduced. Eventually, the outcomes at T4 become structural conditions (T1) in a new cycle of interplay between structure and agency (Archer, 1995; Gimenez, 2007).

4.2.4 Generative Mechanisms

Generative mechanisms are one of the main components of critical realist studies. A central aspect of mechanisms in the critical realism tradition is that they present a source of explanatory power (Archer, 1995; Bhaskar, 1998). Archer (1995) states that structures have properties that enable them to influence the world around them (Archer, 1995; Cuellar, 2010). Johnston and Smith (2008, p. 28) define a generative mechanism as the "causal power that gives rise to something or the reason that something is". Mechanisms are renowned by contingent causality. The actualisation (or deficiency of actualisation) of a mechanism may result in different outcomes in different contexts because the existence of other mechanisms in the same context mostly causes this multi-finality (Henfridsson & Bygstad, 2013).

The aim of a CR-based study is not to uncover general laws, but to understand and explain an outcome through the interaction between structures and agency. The objective is to investigate an observation and hypothesise mechanisms that might explain the observation. These mechanisms are related to the nature of the object of study, not to the regularity of events (Bygstad & Munkvold, 2011; Sayer, 2000).

The emphasis on generative mechanisms is not cause and effect in the positivist sense (Mingers, 2004; Mutch, 2002). It does not look for statistical relationships among variables, but seeks to explain a given event, structure, or development through identifying the processes through which it is generated. It aims to uncover generalisations based on processes not correlations (Mayntz, 2004). The result of enacting these mechanisms typically results in changes or the reproduction of a certain structure (Mingers, 2004; Wynn & Williams, 2012; Wynn Jr & Williams, 2008). Mechanisms fill in the black box of a particular phenomenon; that is, they show how and/or why one thing leads to another (Hedström & Swedberg, 1998; Hedström & Ylikoski, 2010; Wynn & Williams, 2012; Wynn Jr & Williams, 2008).

Generative mechanisms are capacities, and potentials, and tendencies that may or may not be activated depending on the particular contextual settings and the influence of other generative mechanisms that may be active in the same context.

> *Although the outcomes are the result of the enactment of mechanisms, it is possible that no change occurs because of the counteracting effects of one or more other mechanisms. It is also possible that the outcome of one mechanism may exacerbate the effects of another mechanism, further varying the direction, magnitude, or perceptibility of actual events* (Wynn & Williams, 2012, p. 792).

Most established work on mechanisms addresses social aspects (Hedström & Swedberg, 1998; Hedström & Ylikoski, 2010). They provide a typology of mechanisms. Mechanisms are composites and, in order to understand macro-level relationships, we need to understand the situational (macro-micro), action-formation (micro-micro), and transformational (micro-macro) (Hedström & Ylikoski, 2010) chain (see Figure 4.2).

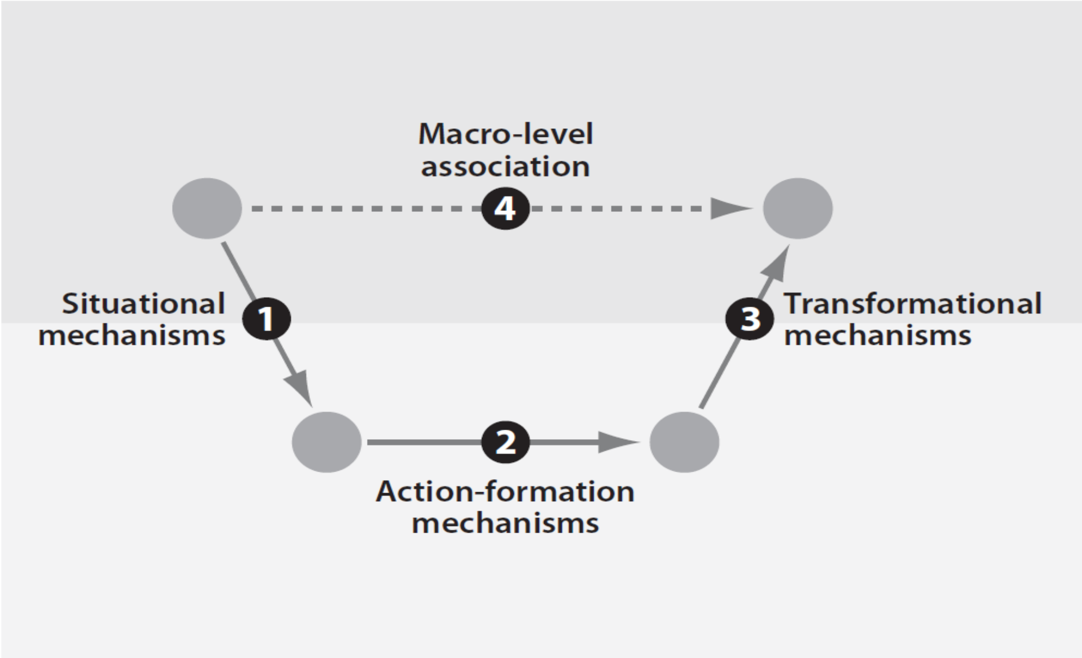


Figure 4.2 A typology of social mechanisms (Hedström & Ylikoski, 2010)

A situational mechanism links macro conditions with individual (micro) behaviour: "the individual actor is exposed to a specific social situation, and this situation will affect him or her in a particular way" (Hedstrom & Swedberg, 1998, p. 23). An action-formation (micro–micro) mechanism explains "how a specific combination of individual desires, beliefs, and action opportunities generate a specific action" (Hedstrom & Swedberg, 1998, p. 23). A transformational mechanism (micro–macro) explains the emergent outcomes (Hedström & Ylikoski, 2010; Henfridsson & Bygstad, 2013). Mechanisms form a hierarchy: "While a mechanism at one level presupposes or takes for granted the existence of certain entities with characteristic properties and activities, it is expected that there are lower-level mechanisms that explain them"(Hedström & Ylikoski, 2010, p. 52).

Hedstrom and Swedberg's (2010) work and classification of generative mechanisms are mapped to the morphogenetic theory's analytical phases (Volkoff & Strong, 2013). The structural conditioning phase includes mechanisms that are situational (conditional), the social interaction phase includes action-formation mechanisms, and the structural elaboration includes transformational (outcomes) mechanisms. Figure 4.3 shows the mapping between the types of generative mechanisms and the SOA integration into EA morphogenetic cycle.

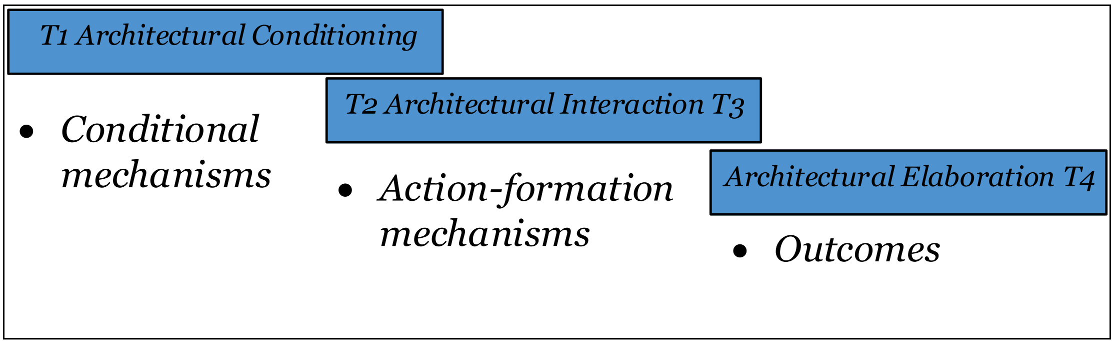


Figure 4.3 Types of generative mechanisms mapped to Archer's morphogenetic cycle

Some studies argue that previous work on generative mechanisms has not paid much attention to technology (Henfridsson & Bygstad, 2013; Volkoff, et al., 2007). Recent studies, however, argue that technology plays an important role at both structural and action levels (Volkoff et al., 2007), and the interaction between social and technical elements constitutes aspects of the process of the mechanism (Henfridsson & Bygstad, 2013). Bygstad and Munkvold (2011, p. 4) describe a well-known mechanism by saying: "user participation in IS development may lead to a higher degree of user acceptance of an IS solution". The basic context for such a mechanism could be that the technical settings are adequately flexible to accommodate changes, and that developers and users are willing to share knowledge (Bygstad & Munkvold, 2011).

4.2.5 Justification for Archer's Morphogenetic Theory

This thesis uses Archer's (1995) morphogenetic theory as a lens to theorise about EA evolution. It is used to describe EA evolution and analyse the interaction between structure (pre-existing EA), action (SOA introduction), and their resulting outcomes. There are many other theories that investigate the interaction between structure and action, such as structuration theory and actor-network theory (Giddens, 1984; Mingers, 2004; Volkoff, et al., 2007). However, this thesis adopts the morphogenetic theory for the following four reasons.

First, as Chapter 3 presents, this thesis chosen a longitudinal retrospective case study design and thus the thesis needed an analytical lens that facilitates such analysis. The morphogenetic theory pays attention to the temporal dimension of change. Social action is conditioned by structures that emerged and endured over a period of time before a specific action occurs (Mutch, 2010). The theory therefore serves as an analytical tool to represent conditions, causes, and consequences during organisational change as a result of IS introduction (Volkoff, et al., 2007). The morphogenetic theory could be seen as a fitting approach to describe the EA evolution process (architectural conditioning (current EA of an organisation), architectural interaction (e.g., SOA introduction), and architectural elaboration (evolution outcomes)) (see Figure 4.4).

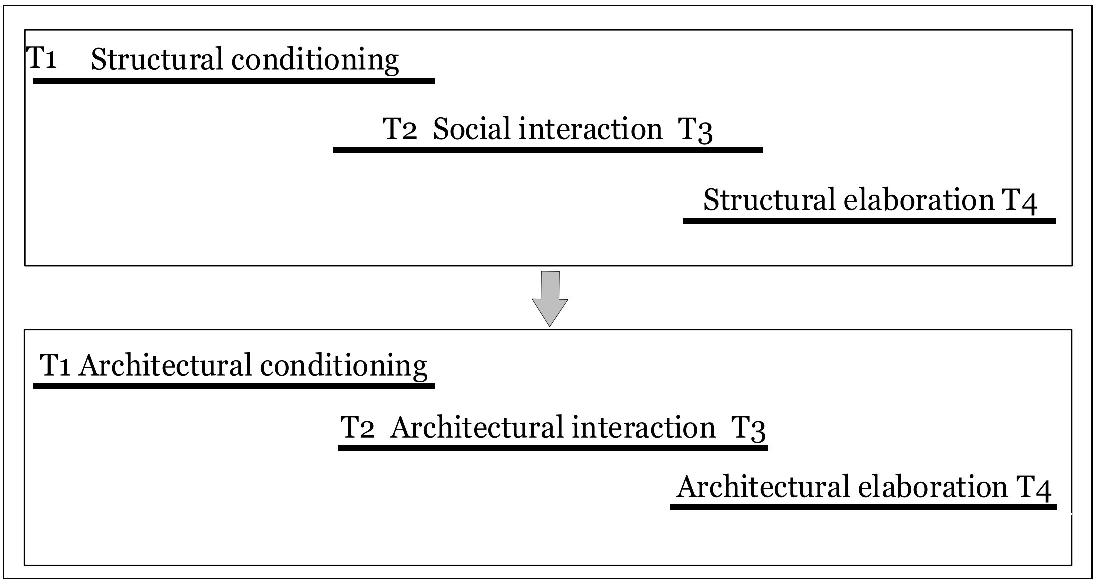


Figure 4.4 Mapping between the morphogenetic cycle and EA evolution

Second, the morphogenetic theory argues for analytical dualism, which rejects the conflation approach used in many other similar theories such as structuration theory and actor-network theory. Analytical dualism denotes that "the emergent properties of structures and agents are irreducible to one another, meaning that in principle they are analytically separable… and given structures and agents are also temporally distinguishable" (Archer, 1995, p. 66).

Some prior studies have compared the morphogenetic theory with rivals (Archer, 1995; Mutch, 2002; Volkoff, et al., 2007). For example, Volkoff et al. (2007) point out that most proposed models of IS-mediated organisational change are based on structuration theory, institutional theory, and actor-network theory. They argue that all the proposed models offer different views that present conflicting perspectives, and these models are problematic. Models based on structuration theory or actor-network theory lean toward agency and pay less attention to technology. On the other hand, models based on institutional theory pay less attention to agency. In addition, technology is typically considered a unitary entity and, therefore, as disregarding the idiosyncratic properties of each technology (Volkoff, et al., 2007).

In structuration theory, the comparative overlook of technology's role in the change process is inherited from Giddens's (1984) argument that structure only exists in the moment of instantiation. In other words, without an actor, there is no structure. While this viewpoint may be suitable for depicting social structures that have no tangible form, it neglects the inherent materiality of technology. The actor-network theory (ANT) downplays the difference between structure and agency: it places technology on the same level as individual actors, together referred to as "actants", and presents them all as actors in a heterogeneous network of diverse components. ANT acknowledges the material aspects of a technology more than structuration theory does. However, it conflates agents and structures, which minimises its ability to investigate how technology mediates change in organisations (Mutch, 2002). Furthermore, ANT neglects the context where the social action takes place because it investigates social action at a micro-level (Volkoff, et al., 2007). Based on this argument, Volkoff, et al. (2007) adopted Archer's morphogenetic model to study organisational changes as a result of IS introduction.

Third, Archer's theory is a widely employed explanatory framework in CR research (Dobson, Jackson, & Gengatharen, 2011). However, very little information systems research has endeavoured to relate or adapt Archer's theory to organisational settings and practices (Morton, 2006). Smith (2006) argues that CR-based theories have the potential to advance IS research. The main benefits:

> *flow from the reinterpretation of the activity of science … that then can better explain previous research … Indeed, such a reinterpretation of the current practice of information systems research arguably resolves some long-standing theory-practice*

> *inconsistencies. In resolving these inconsistencies, CR provides a notion of causality that allows for the capturing of the underlying "why" questions posed in IS research* (Smith, 2006, p. 207)

The morphogenetic theory has been recently used in several IS studies (Dobson, et al., 2011; Dobson, et al., 2013; Dobson, et al., 2007; Morton, 2006; Volkoff, et al., 2007). Wong (2005) used the morphogenetic theory as an explanatory framework in order to holistically explain a complex dynamic and multi-level phenomenon, organisational innovation. Volkoff, et al. (2007) employed Archer's model to study organisational changes associated with implementing ERP. They identified embeddedness of organisational routines, data, and roles into ERP systems as the main mechanisms that describe observed organisational changes over time (Wynn Jr & Williams, 2008). Dobson, Jackson, and Gengatharen (2011) used Archer's theory to study broadband adoption decision processes and the outcomes of such decisions.

Fourth, the morphogenetic theory supports analysis of the interplay between the features of technology and aspects of organisations over time (Mutch, 2010). The morphogenetic theory focuses analysis not only on the properties of technology, but also on their contextual conditions. It facilitates a way to ensure that the richer descriptions of phenomena are not isolated from their context, and thus provides helpful resources for ongoing efforts to understand the interrelationship of technology and organisation (Mutch, 2010). Archer's theory is valuable to investigate the introduction of IS, such as SOA, to organisations and its introduction outcomes (Morton, 2006; Volkoff, et al., 2007). Such analysis suggests considering the existing practices prior to IS introduction (Dobson, et al., 2007). It considers the pre-existing structural conditions arising from previous interaction between structure and agency.

4.3 Building the A-Priori Model[3]

This section details the a-priori research model building phase, which corresponds to the abduction (theoretical redescription) stage of the

[3] Published in Alwadain, Fielt, Korthaus, and Rosemann (2013c)

framework that Danermark, et al. (2002) propound. This phase redescribes the literature review findings using Archer's (1995) morphogenetic theory. It is also a starting point for the subsequent phases; namely, the interview phase and the case study phase.

The findings of the literature review are presented in the following sections using Archer's (1995) morphogenetic theory (see Figure 4.5). The structural conditioning at T1 is called "architectural conditioning" to reflect this thesis's scope, which comprises EA and its changes as a result of introducing SOA. The social interaction is called "T2 architectural interaction T3". "Architectural elaboration T4" represents the result of the interplay between the architectural conditioning and the architectural interaction phases.

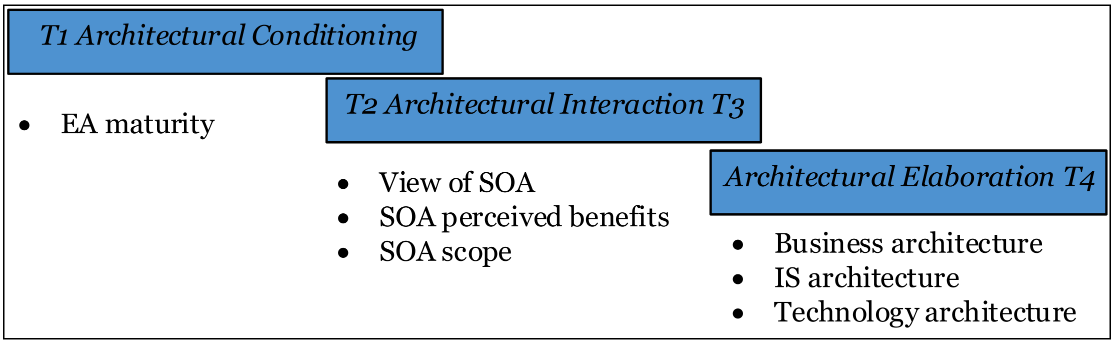


Figure 4.5 The a-priori research model

The three analytical phases of the morphogenetic theory are determined according to stability-change-stability analysis (Njihia, 2008; Trosper, 2005). Njihia (2008); Trosper (2005) suggest that researchers need to tentatively adopt a set of periods, distinguished by times of stability and times of change, in order to identify the analytical morphogenetic cycle (as discussed in Chapter 3)

4.3.1 Architectural Conditioning

As Sections 4.2.1 describes, structural conditioning is the first analytical phase of the morphogenetic cycle. In this thesis, it is called "architectural conditioning" to reflect the thesis's purpose.

IS-related initiatives occur in organisations that have hierarchical structures of previously related activities, resources, and rules. They engage and impact organisational, managerial, and technological aspects of the organisation in which they are implemented (Morton, 2006). Volkoff et al. (2007) identified pre-existing routines, data, and roles that are part of the structural conditioning that impacts their embeddedness into ERP systems (Wynn Jr & Williams, 2008). Some other possible structures are IT artefacts or design specifications. For example, if the object of analysis was an implementation, the structure would be a business process or processes that would be changed as a result of the implementation (Cuellar, 2010).

In this thesis, EA is the main structure that is investigated with respect to its evolution as a result of SOA introduction. In Archer's terms, the implemented EA is a structure that has been shaped by previous morphogenetic cycles, and SOA's introduction may result in EA's elaboration. EA as a structure has its own conditional generative mechanisms (causal powers) that influence its evolution. When EA is implemented in organisations, it must evolve due to organisational changes. As Chapter 2 reports, the maturity of EA could be an enabler or constraint for EA evolution. Thus, the concept of maturity was considered a conditional generative mechanism of EA.

Maturity assigns different levels of achievement by means of a maturity assessment to multiple EA dimensions. These levels point to how mature these dimensions are (Meyer, Helfert, & O'Brien, 2011). In Archer's terminology, EA as a structure has a generative mechanism; namely, its maturity. A generative mechanism has an influence on the world around it and so does EA maturity. It constrains or enables actors' actions. A higher level of maturity enables actors to better identify SOA's relationship to EA and thus achieve better SOA-into-EA integration. On the other hand, a low level of EA maturity would constrain actors' ability to transform EA, leading to a reproduction of existing EA settings that would not reflect the required changes of SOA's introduction.

4.3.1.1 EA maturity

Van der Raadt et al. (2005) state that architecture maturity is measured in terms of an organisation's capability to manage the development, implementation, and maintenance of enterprise-wide architecture on different levels.

EA's value and scope increase when EA maturity increases. Yet, to reach a high level of maturity, EA requires strong planning and management. Research suggests that mature organisations make synergies between EA components and processes to achieve business value (Espinosa, et al., 2011). Schmidt and Buxmann (2011) conclude that EA is still a young practice. EA immaturity is noticed by the degree of variety in regard to objectives, methodology, and organisational implementation of EA (Schmidt & Buxmann, 2011). Organisations that do not evolve their EA over time may find it challenging to adapt rapidly and appropriately to shifts in the marketplace or to strategic restructurings (Bradley, Pratt, Byrd, Outlay, & Wynn, 2012). Although EA is seen a vital management instrument, it still suffers from relative immaturity. Such a low maturity leads to difficulties in establishing an EA function that is effectively integrated into existing organisational practices, and in motivating effective collaboration between architects and other stakeholders. A fragmented and badly integrated EA function typically fails to achieve the expectations of EA (Raadt & Vliet, 2008).

Similar to any other projects, organisations need mature EA to enable SOA's implementation and integration. Mature EA is important to successfully implement SOA and realise its expected benefits (Perko, 2008). EA supports SOA's introduction. It facilitates services identification and classification, and aligns SOA with organisational missions (Brooks, 2009). EA helps business people better understand SOA (Antikainen & Pekkola, 2009), and it is valuable to have business architecture artefacts during SOA implementation to deliver the required set of services (O'Brien, 2009). EA should act as a blueprint for SOA and should be used as a starting point for SOA projects. The availability of detailed business architecture models during SOA implementation and of architects' skills affects SOA implementation (Kokko, et al., 2009). Kokko, et al. (2009) discovered that low maturity of business process models, a subset of EA, is a main obstacle for implementing SOA. Mature EA practices improve business and IT operations (Lagerstrom, et al., 2011) and facilitate SOA's integration into EA (Postina, et al., 2010).

Several EA maturity models have been proposed (see Chapter 2). Most of these EA maturity frameworks have similar dimensions for assessing EA maturity. This thesis adapts the NASCIO maturity model guided by the most commonly used EA dimensions in these studies (Lagerstrom, et al., 2011; Schmidt & Buxmann, 2011). The Open Group consider the NASCIO maturity framework is considered a good example of EA maturity models that could be used to assess government and private EA maturity (The Open Group, 2009e). It conforms to the well-known maturity model SEI SMM (NASCIO, 2003). It is also a widely used EA maturity model (Gosselt, 2012). In this thesis, EA maturity is used as a conditional generative mechanism that conditions SOA's integration into EA and EA evolution in general.

4.3.2 Architectural Interaction

The architectural interaction is the second analytical level of the morphogenetic cycle. Introducing information systems artefacts into organisations results in either the transformation or reproduction of pre-existing structures (Volkoff, et al., 2007). Smith (2005), as cited in Dobson et al. (2007, p. 144), suggests that:

> *Technology introduces resources and ideas (causal mechanisms) that may enable workers to change their practices, but these practices are also constrained and enabled by the structures in which they are embedded.*

In this thesis, the architectural interaction is SOA's introduction. The interaction between SOA and agents when implementing SOA triggers generative mechanisms that impact SOA's introduction. These types of generative mechanism are action-formation mechanisms (Hedström & Swedberg, 1998; Hedström & Ylikoski, 2010). Bygstad and Munkvold (2011) provide examples of typical events in IS studies that trigger mechanisms, such as the decision to buy an ERP system and the integration of the ERP system with other systems. They note that it is often a group of objects that trigger a mechanism and generate an outcome that is dependent on the objects, but not reducible to them. The interaction of agents and technology may activate a group of mechanisms pertinent for the IS field. Triggering the mechanism and the result it might produce is not predetermined, but will depend on other active mechanisms in the context. Nevertheless, the process tends to result in certain outcomes. For instance, user participation in IS development regularly enhances the likelihood of user acceptance, but not always (Bygstad & Munkvold, 2011).

Three action-formation generative mechanisms that may have an impact on SOA introduction were identified from the literature in Chapter 2. As Archer (1995) suggests, the action (SOA's introduction) is shaped by the orientation of the agents and their interests. Agents introduce SOA while entertaining a certain perspective of service-orientation, anticipating certain benefits, and determining a certain scope. These generative mechanisms are "view of SOA", "perceived SOA benefits", and "SOA scope". They are hypothesised to affect SOA's introduction and thus its integration into EA. Chapter 2 shows that the literature views SOA differently (five different perspectives are identified). Chapter 2 also finds that SOA offers wide benefits and could be implemented in different scopes. These three generative mechanisms are presented below.

This thesis focuses on SOA and thus it emphasises SOA-related action-formation mechanisms. *Yet, these action-formation mechanisms are assumed to be relevant for any other architectural interaction triggers such as cloud computing***.** For example view of cloud computing, its perceived benefits and its implementation scope are thought to be applicable action-formation mechanisms for cloud computing.

4.3.2.1 View of SOA

Understanding and perception are widely discussed in IS literature. For example, studies have found that IT and EA perception can impact planning and implementation. Salmans, Kappelman, and Pavur (2009) investigated how IT professionals perceived EA and its implementation. Karimi, Gupta, and Somers (1996) used an organisational understanding of the role of IT among other factors to measure IT maturity. In this thesis, it is hypothesised from the literature findings, presented in Chapter 2, that SOA has different views associated with it, which impacts its introduction.

There are different perspectives for SOA. Even with the view of SOA as an architectural style, there are diverse and different opinions that impact

SOA's implementation (Viering, et al., 2009). Luthria and Rabhi (2009) discovered that most organisations in their study adopted SOA for technical implementation purposes and ignored the wider business perspectives. Although most SOA definitions are predominantly technical, recent publications have taken a broader, business-based viewpoint (Joachim, et al., 2009; Lee, et al., 2010). Organisations in different industries are eagerly pursuing SOA not just as an architectural style but also as a business strategy (Chen, et al., 2010; Shan & Hua, 2006). SOA covers not only EA's technical domains of EA, but also its other domains such as business architecture (Kistasamy, et al., 2010).

Hirschheim, et al. (2010) identified different views of SOA that determine its adoption maturity. They organised these SOA views into five maturity stages: fine-grained service components, emerged software architecture, business process support, enterprise service architecture, and adaptive architecture. Each view is associated with different implementations. The first stage represents a very technical view of SOA. While the last one, adaptive architecture, represents the highest level of SOA adoption maturity, which includes business and IT aspects. Hirschheim, et al. (2010) conclude that how SOA is viewed by an organisation impacts its implementation. This thesis uses Hirschheim et al.'s classification of SOA views (Hirschheim, et al., 2010; Welke, et al., 2011) to investigate view of SOA as an action-formation mechanism that may influence SOA introduction.

4.3.2.2 SOA Perceived Benefits

This thesis hypothesises that different perceived benefits influence the way SOA is adopted. Kohlmann, et al. (2010) conclude a SOA's design varies depending on an organisation's goals for implementing SOA. Further, findings from multiple case studies suggest that SOA has multiple drivers. Kohlmann, Börner, and Alt (2010) conclude that SOA is adopted using different approaches, and that each approach is distinguished by a set of related perceived benefits (e.g., to standardise integration infrastructure, to decouple application domains, and/or to achieve flexible business process integration). These perceived benefits have significant implications for SOA implementation strategies (Legner & Heutschi, 2007).

As Chapter 2 discusses, this thesis adopts Becker et al.'s (2009) and Mueller, Viering, Ahlemann and Riempp's (2007) classification of SOA perceived benefits (i.e., IT, operational/process, and strategic benefits). This thesis hypothesises that different perceived benefits influence the way SOA is introduced.

4.3.2.3 SOA Scope

As Chapter 2 discusses, there are different SOA scopes. Each scope has different characteristics and requires different strategies. Each scope also seems to influence SOA's implementation and thus its association with EA. For example, Campbell and Mohun (2007) discuss three different approaches for SOA adoption: project, portfolio, or enterprise level. Each scope affects different levels of the organisation (see Figure 4.6). Afshar (2007) also distinguishes between three SOA adoption strategies: project driven, infrastructure driven, and enterprise driven.

SOA projects have different scopes. Each project type may require different resources, methods, and tools to determine its implementation activities, and have wide technical and organisational aspects that need to be determined O'Brien (O'Brien, 2009). This thesis adopts the scope of SOA (e.g., project, portfolio, and enterprise), and hypothesises that different scopes impact the way SOA is introduced.

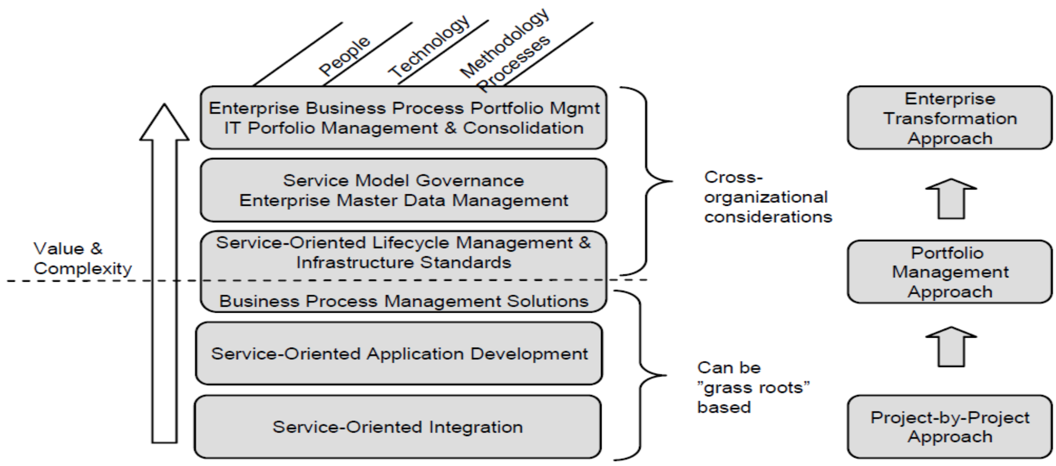


Figure 4.6 Value and scope of SOA adoption (Campbell and Mohun, 2007)

4.3.3 Architectural Elaboration

Structural elaboration is the third analytical phase of the morphogenetic cycle. It is called "architectural elaboration T4" in this thesis. It represents the architectural reproduction or transformation of pre-existing architectural conditions.

The EA literature discusses wide EA frameworks that use different architectural layers. Thus, this thesis aimed to find common layers in the literature to use them to structure the findings. These layers of EA—business (information and applications) and technology—are widely accepted and used in the EA discipline (Lankhorst, 2004; Pulkkinen, 2006). Additionally, The Open Group Architecture Framework (TOGAF), a widely used EA (Infosys, 2009), uses a similar structure: business, information systems, and technology (The Open Group, 2009d). Therefore, this thesis adopts the three layers of business, information systems, and technology to structure EA evolution outcomes.

The interaction between pre-existing architectural settings (conditional generative mechanisms) and SOA's introduction (action-formation mechanisms) leads to an architectural elaboration. It results in either architectural transformation (SOA's integration into EA) or the reproduction of EA on one or more of three levels. They are business architecture, IS architecture, and/or technology architecture.

4.3.3.1 Business Architecture

The first level of architectural elaboration resulting from SOA's introduction is SOA's integration or lack of integration into business architecture. Integration (transformation) means that SOA and relevant SOA elements such as business services, service description, service channels, SOA vision, drivers, service actors, SLAs, and QoS are integrated into business architecture. Such integration often builds on SOA integration with lower layers of EA architecture (e.g., SOA is integrated into IS and infrastructure architectures or is going to be integrated). Building on the examples already presented in Chapter 2, the following paragraphs provide a summary of such examples.

First, Correia and Silva (2007) introduce a new column to the Zachman Framework to represent the service concept. They incorporate services' business, IS, and technology aspects. Second, Khoshnevis, et al. (2009) integrate high-level SOA-relevant elements into Zachman's Framework by adding a new column. Third, Schmelzer (2006) integrates SOA into the Zachman Framework by using the existing nine squares without elaborating on what elements need to be integrated. Fourth, Iacob et al. (2007) show how ArchiMate has been developed as a modelling language that uses services on all the three architectural layers. Fifth, TOGAF 9 integrates SOA elements into all three of its architectural layers (The Open Group, 2009b, 2009d). Sixth, some organisations have adopted the view of becoming a service-oriented enterprise (SOE). For example, Intel has an architectural vision of integrating SOA into their business architecture, IS architecture, and technology architecture. The organisation is structured in terms of services on all the architectural levels (Hirschheim, et al., 2010). Seventh, Aier and Gleichauf (2009) propose a three-layer enterprise architecture representing three sub-architectures: service-oriented process architecture, service-oriented integration architecture, and service-oriented software architecture. However, the abovementioned examples vary in their coverage of SOA elements (see Chapter 2).

4.3.3.2 Information Systems Architecture

The second level of the architectural elaboration resulting from introducing SOA is SOA's integration or lack of its integration into information systems architecture. The integration is determined based on SOA elements' presence such as application services, service descriptions and SLAs in information systems architecture.

This section summarises the detailed examples in Chapter 2. First, Laplante, Zhang, and Voas (2008) argue that SOA belongs to the network (where) column in the Zachman Framework. The SOA network model is a list of possible services to be used in a software system under development. Second, Schelp and Aier (2009) report the findings from a bank in Switzerland that introduced SOA. Several architectural levels are identified: business, application (integration), software component, and technical

Architecture. SOA is integrated into the application architecture. Another company that introduced SOA is a telecommunication service provider in Germany (Schelp & Aier, 2009), which introduced it to reduce the complexity of its distributed application landscape. Enterprise services are integrated into the integration architecture, while basic services (software components) are integrated into the software architecture. Fourth, in Erl's (2005) enterprise model, the service layer is located between the business process layer and the application layer, where most of the SOA characteristics are prevalent. Erl (2005) divides the service layer into three layers of abstractions. Fifth, Engels and Assmann (2008) and Assmann and Engels (2008) propose integrating the services layer between business and IT architectures. It has basic services (applications services) and composite services (process services). Sixth, Jung (2009) define SOA as an approach for application design and development, and integrated it into EA on the applications level. Seventh, Kistasamy, et al. (2010) propose integrating services and services components into application architecture. They also suggest integrating ESB, QoS, and services monitoring into technology architecture.

4.3.3.3 Technology Architecture

The third level of the architectural elaboration resulting from introducing SOA is SOA's integration/lack of integration into technology architecture. SOA is integrated into technology architecture using relevant elements such as technology services, service interfaces, messages, services monitoring elements, services security elements, and physical technology components (SOA infrastructure; e.g., repository, enterprise service bus (ESB), BPEL executors, and registry).

Chapter 2 examines broad examples of such integrations. The following paragraphs present them in brief. First, the N.S.W Department of Lands implemented SOA into its technology architecture. At the beginning of its journey, it began integrating SOA into its technology architecture when an enterprise service bus (ESB) was first introduced in 2005. Second, a logistics operator in Finland adopted SOA during 2005. It was an IT-driven project that used a technical bottom-up approach, and SOA was almost equal to web services. Later, it was expanded into multiple projects to integrate the legacy system landscape, employing SOA technology to mainly expose legacy system services via an integration platform (Kokko, et al., 2009). Third, a public sector organisation in Finland adopted SOA to build a service platform and J2EE-based infrastructure platform to support XML and web service interfaces. Later the platform was expanded in iterative SOA projects (Kokko, et al., 2009).

4.4 Chapter Summary

The chapter presented and justified Archer's morphogenetic theory (1995) to examine SOA's integration into EA. The theory is selected because it pays attention to the temporal dimension of change which is a fitting approach to describe the EA evolution. It supports the analytical dualism where EA and SOA are analytically held apart during the analysis. It is very useful explanatory framework for information systems research (Dobson, et al., 2011).

The morphogenetic theory is adopted to understand (1) the process of EA evolution when a new IT capability (here: SOA) is introduced and (2) explain the evolution outcomes. It is used as an analytical tool to exemplify the interaction between pre-existing architectural settings, SOA introduction and the elaboration (outcomes) occurring as a result of that SOA introduction.

This chapter discussed the three analytical phases of the theory in general (section 4.2) and in relation to this thesis (Section 4.3). It redescribed the literature review findings using the morphogenetic theory analytical phases as suggested in the theoretical redescription stage of the critical realist methodological framework by Danermark, et al. (2002). By using the theory as an analytical lens, this chapter builds the a-priori model of this thesis that describes the process of EA evolution and provides initial explanation (generative mechanisms) of EA evolution outcomes.

In Archer's (1995) terms, architectural conditions at T1, prior to SOA introduction, are the results of previous actions. These architectural conditions have an influence through their conditional generative mechanism (EA maturity) on the action (SOA introduction). The maturity of EA either enables or restricts the action of SOA introduction between T2 and T3 that may transform EA.

The architectural interaction phase (here: SOA introduction) is influenced by action-formation generative mechanisms. These generative mechanisms are the view of SOA, SOA perceived benefits and SOA scope. Although there SOA-specific, these action-formation mechanisms are assumed to be relevant when examining other EA evolution triggers such as cloud computing. For example, “view of SOA” would be “view of cloud” and so on.

The interaction between the architectural conditioning (T1) and SOA introduction (T2-T3) results in architectural elaboration (T4)—either transformation or reproduction of pre-exiting EA. EA is transformed or reproduced on some or all of these three levels: business architecture, information systems architecture and technology architecture.

Chapter 5: Interview Findings

5.1 Introduction

This chapter presents and examines the findings of twenty interviews conducted with enterprise architects and EA consultants to refine and extend the a-priori model presented in Chapter 4. It covers participants' demographics and the interview data analysis process. Following the analysis of the interviews, the updated a-priori model is presented.

Findings are structured according to the a-priori model's layout. As such, this chapter progresses as follows. Section 5.2 presents participants' information and selection criteria. Section 5.3 shows the interview analysis process. Section 5.4 presents findings related to the first phase of the model (architectural conditioning). It discusses the three conditional generative mechanisms: EA frameworks, EA objectives, and EA maturity. Section 5.5 focuses on the architectural interaction phase, SOA's introduction, and its generative mechanisms (e.g., the view of SOA, SOA perceived benefits, SOA scope, SOA governance, SOA design, and business and IT collaboration). Section 5.6 examines the architectural elaboration phase that results from SOA's introduction. Specifically, it presents EA elaboration on five levels in its sub-sections. Finally, Section 5.7 shows the refined model and concludes the chapter.

5.2 Interview Participants

Two types of informants were included in this thesis: EA practitioners (e.g., chief enterprise architects) and EA consultants. They were selected because they are involved in SOA's integration into EA and they have the knowledge and the expertise that this thesis requires.

The candidates were selected on the basis of (1) their role in the respective organisations (e.g. chief enterprise architect), and (2) their engagement in integrating SOA into EA. They were targeted and interviewed to elicit their EA and SOA practices and their approach to integrating SOA into EA. Each interview followed an interview protocol developed prior to their commencement.

Candidates from private and public sectors and consultancy groups were approached. The candidates were chosen using a mix of two strategies: (1) active solicitation based on media reports describing an organisation as having an EA and SOA focus, and (2) a snowballing technique leveraging a network of personal contacts in the industry and EA groups and forums.

To set up the scene, participants, when first approached, were informed about the study's nature and purpose. On agreeing to participate, they were given the study's details and asked to sign the consent form. Confidentiality issues were discussed and explained before the interviews were conducted. The protocol was used to guide the interviews. A broad set of open-ended questions loosely derived from the components of the a-priori model were used to guide the interviews and provide consistency across the entire process. These questions covered the following:

- A general understanding of the organisational profile, organisational structure, and the interviewee's role in the organisation
- The interviewee's understanding of EA and EA framework, methodology, and EA benefits
- The interviewee's perception of SOA, its perceived benefits, scope, and other relevant aspects of SOA introduction
- SOA's integration into EA approaches, issues, how SOA elements are integrated into EA, services in EA, service models, meta-models and examples, and
- Factors that might have influenced SOA's integration into EA: view of SOA, SOA scope, SOA benefits, EA maturity, and other potential factors.

Twenty interviews with enterprise architects and EA consultants (comprising 13 with enterprise architects and nine EA consultants) were conducted. Interviews I-5 and I-6 (i.e., interview number 5 and 6) were done separately with two participants from the same organisation. Two participants (differentiated as I-4 and I-11) were interviewed at the same time. Most of the interviews lasted between 45 and 60 minutes. Some interviews were conducted face-to-face while others were done via phone. Follow-up interviews were conducted later with three interviewees to obtain more information. Table 5.1 summarises the participants' information. It describes the industry sector of the interviewees' firms, the role or designation of the interviewees, the length of each interview, and the way each interview was conducted. Some interviewees provided richer information with extra documents, reports, presentations, or meta-models.

Table 5.1. Interview participants' information

Interview	Job title	Industry sector	EA framework	Interview length	Interview mode
I-1	Chief Enterprise Architect	Banking	Proprietary EA framework	90 minutes	Phone (via Skype)
I-2	Chief Enterprise Architect	Multi-business	Modified TOGAF	60 minutes	Face-to-face
I-3	Strategic Architect	Government	ArchiMate (Partial)	50 minutes	Face-to-face
I-4	(a) Chief Enterprise Architect and (b) Senior Enterprise Architect	Health	Modified TOGAF	60 minutes	Face-to-face
I-5	Senior Enterprise Architect	Banking	Modified TOGAF	90 minutes	Face-to-face
I-6	Senior Enterprise Architect	Banking	Modified TOGAF	60 minutes	Face-to-face
I-7	EA Consultant	Consultancy	TOGAF	60 minutes	Face-to-face
I-8	Architecture Manager	Health	DoDAF	40 minutes	Face-to-face
I-9	EA Consultant	Consultancy	TOGAF	60 minutes	Phone
I-10	EA Consultant	Consultancy	Modified TOGAF	60 minutes	Phone
I-11	(a)EA Manager and (b)Architecture Manager	Education	In-house-developed EA based on TOGAF	60 minutes	Face-to-face
I-12	EA Consultant	Consultancy	TOGAF, Zachman	60 minutes	Phone
I-13	Enterprise Architect	Government	Meta-Group methodology, now TOGAF	60 minutes	Face-to-face
I-14	Architecture Manager	Banking	Built-in Framework (Partial models)	50 minutes	Face-to-face
I-15	Enterprise Architect	Banking	Built-in Framework	60 minutes	Face-to-face
I-16	EA Consultant	Consultancy	Gartner	60 minutes	Phone
I-17	EA Consultant	Consultancy	TOGAF	60 minutes	Phone
I-18	EA Consultant	Consultancy	Modified TOGAF	50 minutes	Phone
I-19	EA Consultant	Consultancy	TOGAF	50 minutes	Face-to-face
I-20	EA Consultant	Consultancy	TOGAF, DoDAF	60 minutes	Face-to-face

5.3 Interview Analysis Process

As Chapter 3 discusses, a thematic analysis technique was used to analyse the interview data. The thematic analysis is a widely used qualitative analysis technique (Braun & Clarke, 2006) that searches data for themes that appear as being important to the description of a particular phenomenon (Fereday & Muir-Cochrane, 2006). It offers an accessible and theoretically flexible approach for qualitative data analysis (Braun & Clarke, 2006). A thematic analysis process that combines deductive and inductive coding improves the rigor of a qualitative study (Fereday & Muir-Cochrane, 2006). Thus, this thesis adopted thematic analysis processes similar to processes suggested in many qualitative analysis books and articles (e.g., see Braun & Clarke, 2006; Fereday & Muir-Cochrane, 2006; Miles & Huberman, 1994; Ryan & Bernard, 2000).

The employed data analysis process in this thesis is presented in the following paragraphs. Additionally, Figure 5.1 presents a high-level illustration of this process, and Figure 5.2 shows an example of the coding.

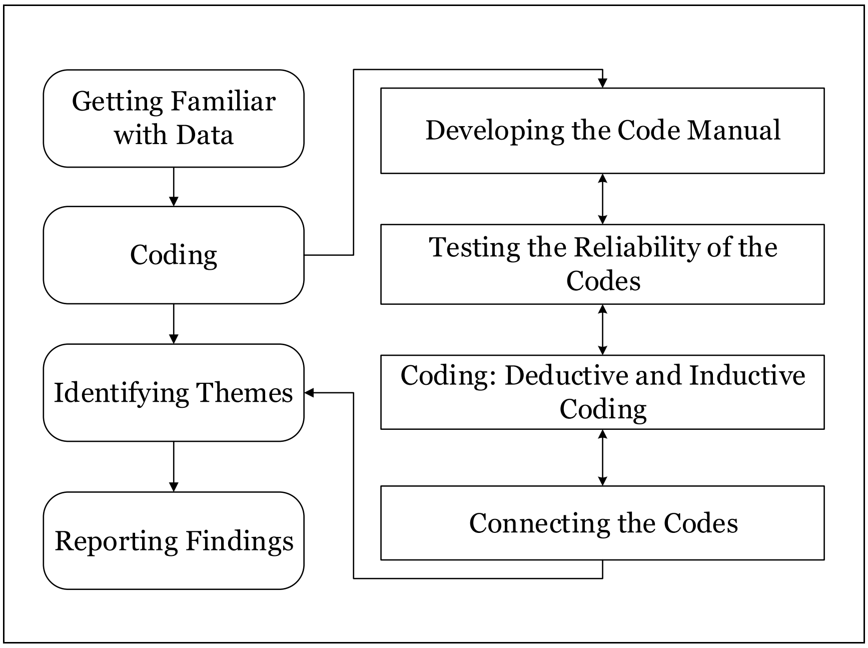


Figure 5.1 Interview analysis process

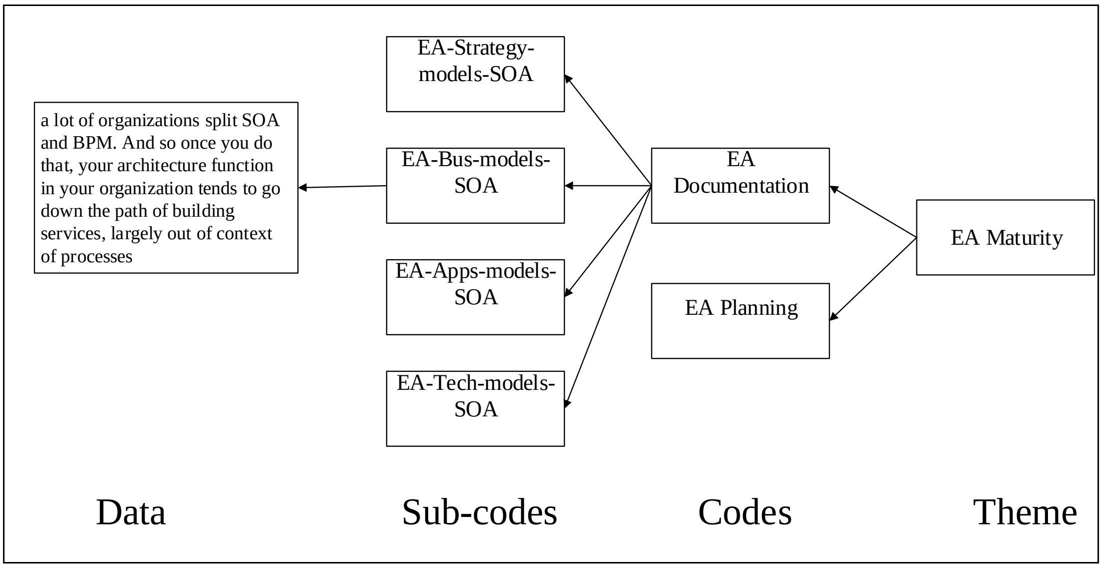


Figure 5.2 Coding example

5.3.1 Getting Familiar with Data

Braun and Clarke (2006) suggest that investigators should immerse themselves in their data to the degree that they become familiar with the depth and breadth of its content. Immersion typically entails reading the data, searching for meanings, themes, and so on (Braun & Clarke, 2006; Maxwell, 2005). Maxwell (2005) proposes that the initial step in qualitative analysis is reading the interview transcripts, notes, or documents to be analysed.

In this thesis, every interview was recorded, listened to, transcribed, and read. Then, the material was imported to a qualitative analysis tool (NVivo 9) to prepare it for coding.

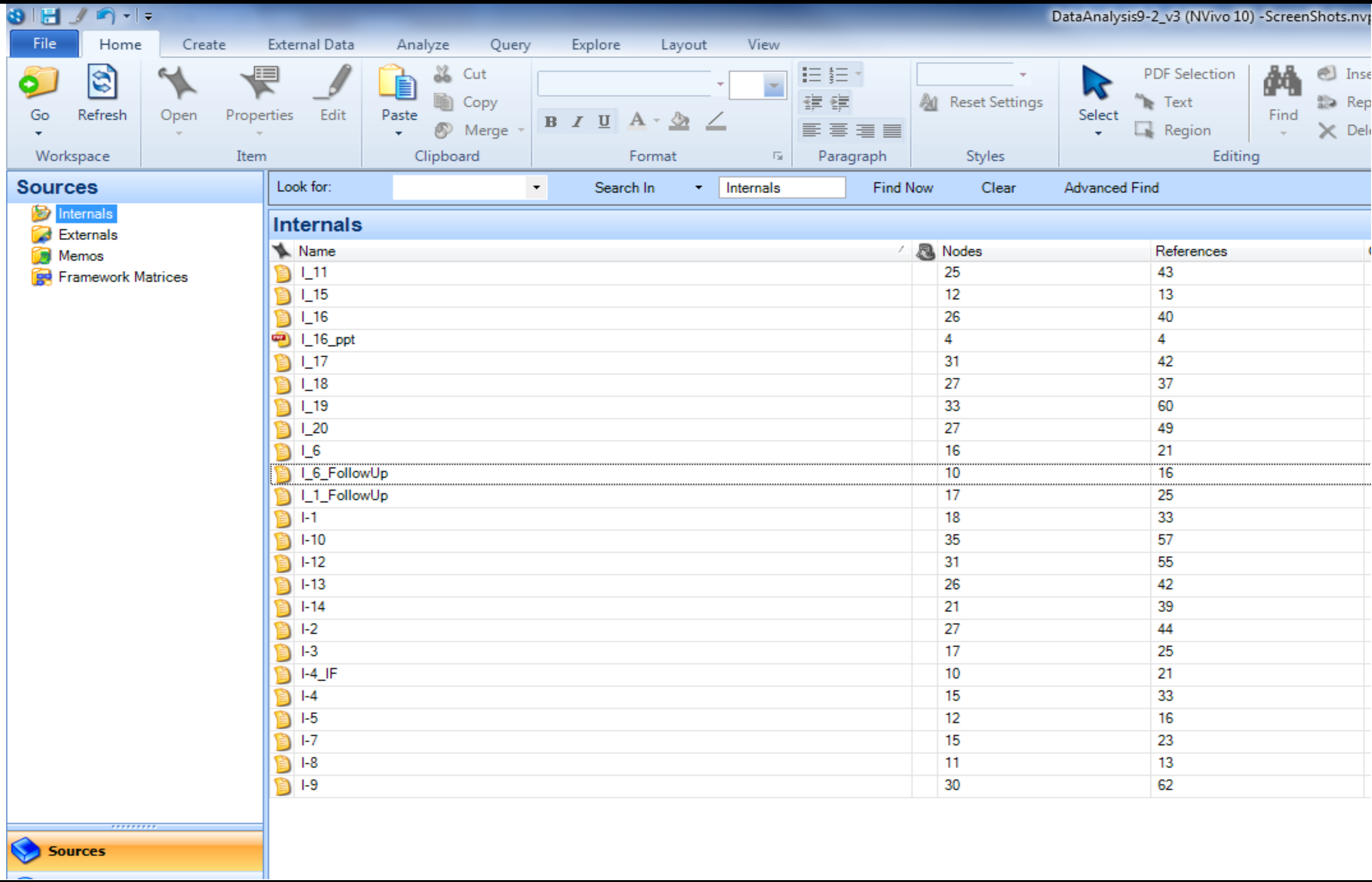
5.3.2 Coding

This phase begins after reading through the data and getting familiar with it. As a starting point for this thesis, and following the guidelines from Miles and Huberman (1994), a predefined set of codes were deductively developed from the a-priori model and the interview protocol. They were defined in a codebook to capture information related to EA, SOA initiative, SOA's integration into EA, and the factors that may influence SOA's integration into EA.

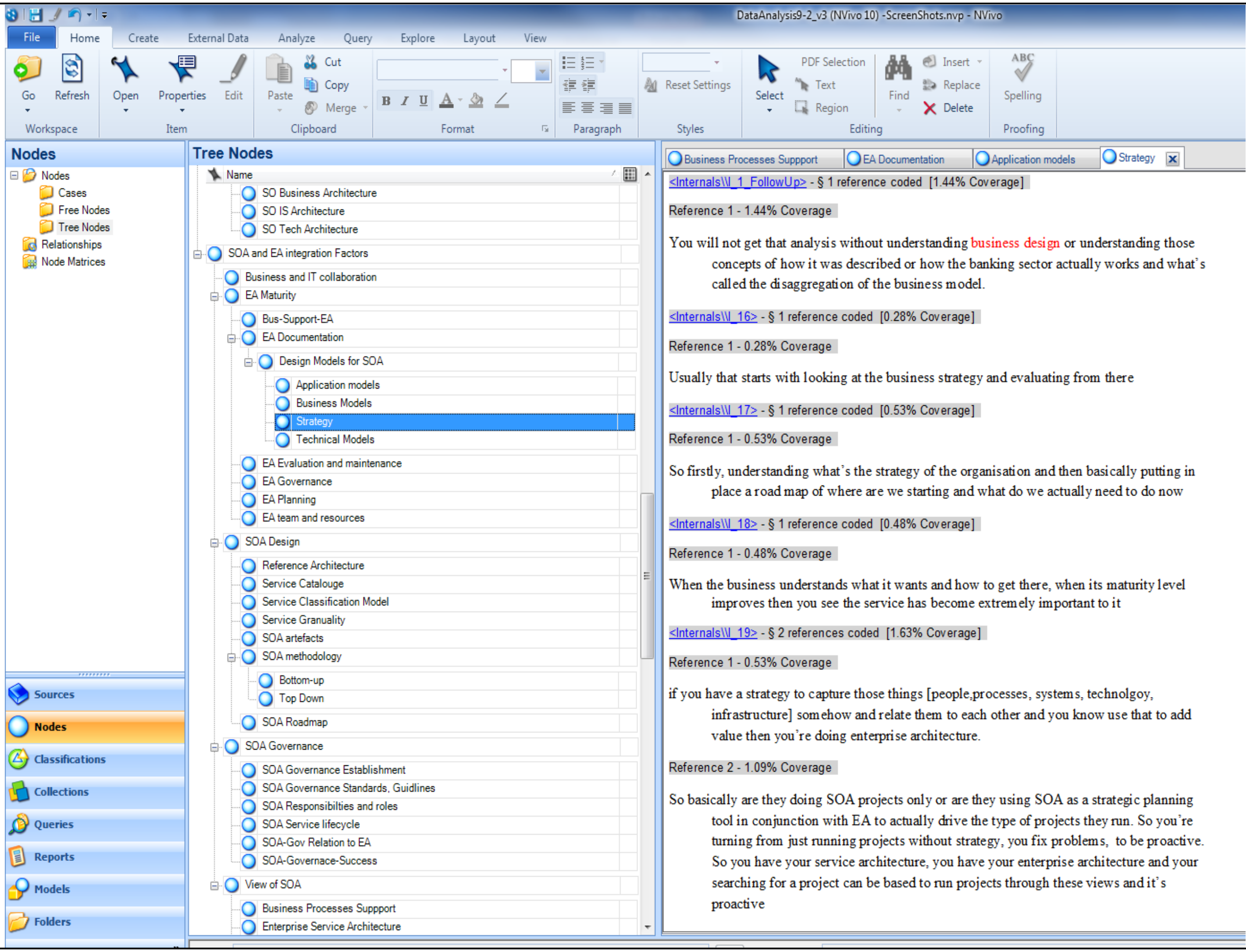
Similarly, inductive coding was performed during the analysis to identify key thoughts and concepts relevant to the study questions. When potential new relevant codes were identified, new nodes were created and data coded in NVivo 9. At the same time, the code and its definition were added to the codebook.

Some segments of the text were sometimes coded under more than one code in NVivo 9 when relevant. On completing each interview's coding, the codes were repeatedly evaluated to deal with redundant codes or codes that could be subsumed by other codes. During this process, the set of codes and their associated meanings were maintained in NVivo, representing the codebook for the analysis, as Ryan and Bernard (2000) suggest. Table 5.2 shows screenshots of NVivo's use.

Table 5.2 Using NVivo for the data analysis

Description	Screenshot
Including interviews in NVivo and preparing them for analysis	

Tree nodes section shows themes, codes and sub-codes while the right half of the figure shows the coded data.



The analysis process is iterative in nature; therefore, multiple passes were undertaken in order to code the data (Miles & Huberman, 1994). Some codes were refined and extended during the analysis, while others were merged with similar or redundant ones, or re-coded if necessary. This iterative process yielded the codes shown in the codebook in Table 5.3. The (*) sign denotes that a code emerged from the data.

Table 5.3 Interviews analysis codebook

Codes	Code meanings	Codes	Code meanings
EA-Accountability	EA is used for accountability (governance)	SOA business support	Level of business support of SOA
EA-Business-IT Alignment	EA is used for business -IT alignment (strategic)	SOA-EA governance *	The integration between SOA and EA in terms of governance practices
EA-Communication	EA is used for communication (operational)	SOA-EA methods, tools*	The integration between SOA and EA in terms of their methods and tools
EA-Decision Making	EA is used for decision making (operational)	EA vision	The view of EA and the purpose of it
EA-Enterprise Integration	EA is used for enterprise integration (operational)	EA planning	The process of building the target architecture based on the as-is model, roadmap and methodology.
EA-Governance	EA is used for org setting standards and policies (governance)	EA documentation	The documentation of as-is, the level of documentation, used framework, and methodology.
EA-Standardisation	EA is used for standardization (governance)	EA evaluation and maintenance	The degree of evaluation, maintenance, updated of EA artefacts and models
EA- Managing Change	EA is used for managing change (strategic)	EA governance	EA governance practices, standards and methods
EA-Reduce complexity	EA is used to reduce complexity (operational)	EA team and resources	EA team structure, position, resources, and skills
EA-strategy execution	EA is used for strategy execution (strategic)	SOA-reference Architecture*	The use of reference architecture, guidelines, defined target architecture, multiple viewpoints for SOA stakeholders
Used EA framework	Identification of the used EA framework	Service catalogue*	The use of service catalogue, its structure and content
SOA-Agility	SOA is used to achieve agility	Service classification model*	The use of service layering/classification model, with services descriptions and definitions

Codes	Code meanings	Codes	Code meanings
SOA-Business-IT alignment	SOA is used to improve B-IT-alignment	SOA artefacts	What are SOA artefacts that are captured in EA
SOA-Communication	SOA is used to improve communication	SOA methodology*	The chosen service identification methodology
SOA-Complexity	SOA is used to reduce IS landscape complexity	SOA roadmap*	The use of an incremental, iterative, long term plan for SOA
SOA-Increased availability of Information	SOA is used to increase the availability of information and information sharing	SOA-Governance relation to EA*	SOA governance and relation to EA governance
SOA- IT Integration	SOA is used to improve IT integrations	Importance of SOA-governance *	SOA governance and its relation to SOA success
SOA-BP improvements	SOA is used for business processes improvements	SOA-governance establishment*	SOA governance, guidelines and standards, and reference models are established and used
SOA-Facilitation of software development	SOA is used to facilitate software development	SOA-governance: roles and responsibilities*	SOA responsibilities and roles establishment
SOA-Reduce Time to Market	SOA is used to reduce products and services time to market	SOA-governance: service lifecycle*	Governance practices related to service lifecycle such as development, management, versioning policies, procedures, and so on
SOA-Reduce IT Costs	SOA is used to reduce costs	Enterprise service architecture	Define services in terms of business needs and in advance of their use in processes. Redesign of business processes to achieve organisational agility.
SOA-Reuse	SOA is used to achieve reuse	Business processes support	SOA becomes an effective means to support business process redesign. Services are driven by business requirements and defined in business functionality terms
Used EA framework	Used EA framework, its structure, description	Software architecture	SOA is viewed as software architecture, development of more fine-grained

Codes	Code meanings	Codes	Code meanings
			services. It is limited to the application landscape without business processes consideration.
Industry of Org.	Information about the organisation	Used EA framework description	Description of the used EA structure, its use and modelling
Job Title	Information about the interviewee	Software Components - Web Services	Development of fine grained services, mostly web services, unstructured development, "just a bunch of web services" (as an interviewee described them)
SOA-Org Level	SOA is an organisational level initiative / used on org level	SOA- IS architecture	SOA is integrated with information systems architecture and presence of SOA elements such as application services, service descriptions, and SLAs
SOA-Portfolio level	SOA is a portfolio level initiative / used on portfolio level / iterative manner	SOA-tech architecture	SOA is integrated with technology architecture, presence of SOA elements such as technology services, services monitoring, services security, ESB, XML standards, web services, and so on
SOA-Project level	SOA is adopted on small project level	EA-services examples	Examples of services on EA layers
SOA definition	SOA definition	EA-tech-models-SOA	Availability and use of technology and infrastructure models for SOA
SOA (Bus/IT) Skills	SOA team bus and IT skills and level of training	EA-strategy-models-SOA	SOA alignment with EA strategy, and use of strategy, goals, roadmaps for SOA
SOA (Bus/IT) team	SOA (business, IT or both?) team, its structure, position in org.	EA-apps-models-SOA	Availability and use of EA application models for SOA
Importance of EA for SOA integration	justifications for SOA's integration into EA	EA-bus-models-SOA	Availability and use of business models for SOA

Codes	Code meanings	Codes	Code meanings
SOA-Business Architecture	SOA is integrated with business architecture, presence of related SOA elements such as business services, service channels, SOA vision, drivers, SLA, and QoS	EA business support	Level of business support of EA

5.3.3 Identifying Themes

Developing and refining themes has long been central in classical qualitative analysis. Codes are analysed and organised into themes that might merge or be split into separate themes (Ryan & Bernard, 2000).

This phase focuses on analysing data at a higher level to identify themes rather than codes. The codes, identified previously, are grouped and organised into possible themes that describe them collectively. It is an iterative back-and-forth process.

Miles and Huberman (1994) suggest that theme analyses should begin with some general themes derived from the relevant literature; as the analysis progresses, more themes or sub-themes should be added.. Once a set of candidate themes are developed, it is necessary to refine these themes. Some candidate themes are not really themes (e.g., there are not enough data, the data are too varied, two themes might form one theme, or other themes might need to be broken down into separate themes) (Ryan & Bernard, 2000). Table 5.4 shows the final themes and their associated sub-themes after the interview data was analysed. The "**" sign denotes that the theme or sub-theme emerged from the analysis.

Table 5.4 Final themes and their associated codes

Themes	Codes
EA framework**	Used EA framework
	Used EA framework description
EA objectives**	EA-accountability
	EA-business-IT alignment
	EA-communication
	EA-decision making
	EA-enterprise integration
	EA-governance
	EA-standardization
	EA- managing change
	EA-reduce complexity
	EA-strategy execution
EA maturity	EA documentation
	EA planning
	EA governance
	EA evaluation and maintenance
	EA team and resources
	EA business support

Themes	Codes
View of SOA	Adaptive architecture
	Enterprise Service architecture
	Business processes support
	Software architecture
	Software components - Web services
SOA perceived benefits	SOA-agility
	SOA-business-IT alignment
	SOA-communication
	SOA-complexity
	SOA-BP improvements
	SOA-facilitation of software development
	SOA-increased availability of information
	SOA-IT integration
	SOA-reduce time to market
	SOA-reduce costs
	SOA-reuse
SOA Scope	SOA-org level
	SOA-portfolio level
	SOA-project level

Themes	Codes
SOA Governance **	SOA-governance relation to EA **
	Importance of SOA governance **
	SOA-governance establishment **
	SOA-governance: roles and responsibilities **
	SOA-governance: service lifecycle **
	SOA-reference architecture **
SOA Design **	SOA-reference architecture **
	SOA roadmap **
	Service catalogue **
	Service classification model **
	SOA methodology **
Business/IT collaboration **	SOA (bus/IT) skills
	SOA business support
	SOA (bus/IT) team

Themes	Codes
SOA's integration into EA	SOA-business architecture
	SOA-IS architecture
	SOA-tech architecture
	SOA-EA governance **
	SOA-EA methods and tools**

5.3.4 Reporting the findings

This phase started when the themes were identified to present the collected data in meaningful way (i.e., a way that would improve understanding of the research problem). It is important to provide a logical, coherent, and exciting story from the data (Braun & Clarke, 2006; Miles & Huberman, 1994). Figure 5.3 illustrates the a-priori model of this thesis, which is refined and extended using the interview findings at the end of the chapter[4].

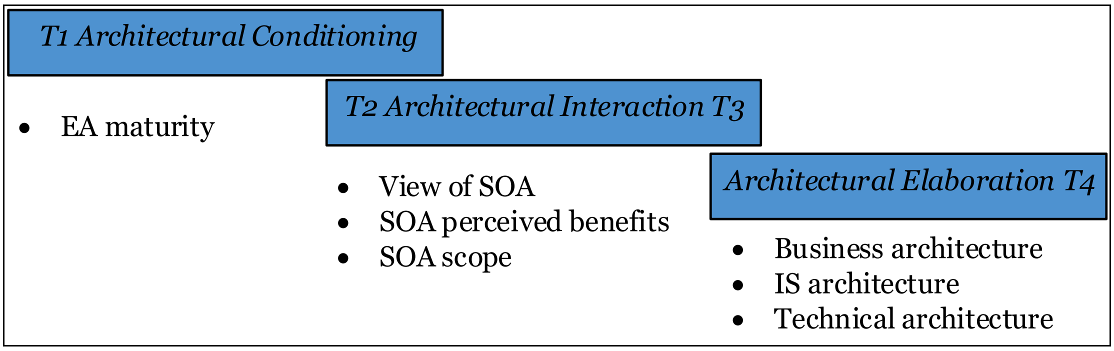


Figure 5.3 The a-priori research model

5.4 Architectural Conditioning

This section examines the diverse architectural settings reported in the interviews in order to enrich understanding of the impact of architectural conditioning on SOA's integration into EA. The findings are organised along the three conditional generative mechanisms: EA framework (inductively identified), EA objectives (inductively identified), and EA maturity (deductively derived). They are hypothesised to have a conditional influence on EA evolution.

5.4.1 EA Framework

EA framework defines recommended architectural artefacts, describes how those artefacts are related to each other, and provides generic definitions for these artefacts. EA frameworks are different in terms of their scope, artefacts, design, and focus. The participants reported different EA frameworks that had been modified, extended, or partially adopted, which adds to the already established variety. In other words, these adopted EA

[4] Part of this chapter was published in Alwadain, Fielt, Korthaus, and Rosemann (2013b)

frameworks have been shaped by previous morphogenetic cycles supporting Archer's (1995) argument that previous actions create the conditioning context (enabling or constraining) for the following action. Table 5.5 shows examples of the adopted EA frameworks. Some organisations adopted widely used EA frameworks, others modified these frameworks, and others internally developed their own EA frameworks. Additionally, these frameworks were adopted partially, lightly, or comprehensively across the organisations. According to EA consultant [I-18], SOA's integration into these frameworks is not straightforward:

> *Fitting SOA of that capability into a framework isn't as flexible as we all like to think … I haven't seen one yet that does the services component really well* [I-18].

Table 5.5 Examples of EA frameworks used

Interview	EA framework	Description
I-1	Proprietary EA framework	Proprietary EA framework has four layers (business intent, business design, people design and technology design)
I-2	Modified TOGAF	Integrated with other management practicians and security architecture is embedded within all four layers
I-3	ArchiMate (partial adoption)	Focus on business and application layers
I-4	Modified TOGAF	Adapted to develop an interoperability framework
I-5	Modified TOGAF	Focused on the technology aspects, later extended to included business aspects
I-6	Modified TOGAF	
I-10	Modified TOGAF	Use of some parts of TOGAF informally
I-11	In-house-developed EA based on TOGAF	Light-weight EA based on TOGAF.
I-12	TOGAF, Zachman	A consultant reported that they are used among their clients
I-13	Meta-group methodology, now TOGAF	Started with the meta-group methodology and now moving to TOGAF based EA.
I-14	Built-in framework (partial models)	Partial models no single unified framework
I-15	Built-in framework	Mostly focus on applications and infrastructure

The following paragraphs provide examples of the reported EA frameworks. Enterprise architect [I-13] reported that his organisation's architectural settings went through multiple iterations. At the beginning, the organisation adopted the meta-group EA framework that Gartner acquired in 2005. Recently, they started to adopt TOGAF-based architectural settings. He said:

> *We've been through a couple of iterations of this exercise … But fundamentally we used the meta-group methodology for a long time and we've worked through that and we're starting to look at the TOGAF framework* [I-13].

Chief enterprise architect [I-2] reported that TOGAF had been modified and extended (see Figure 5.4). It had four main layers: business, information, application and technology. The security architecture was embedded in all the layers. It was well integrated with strategic sourcing, project management, and customer services.

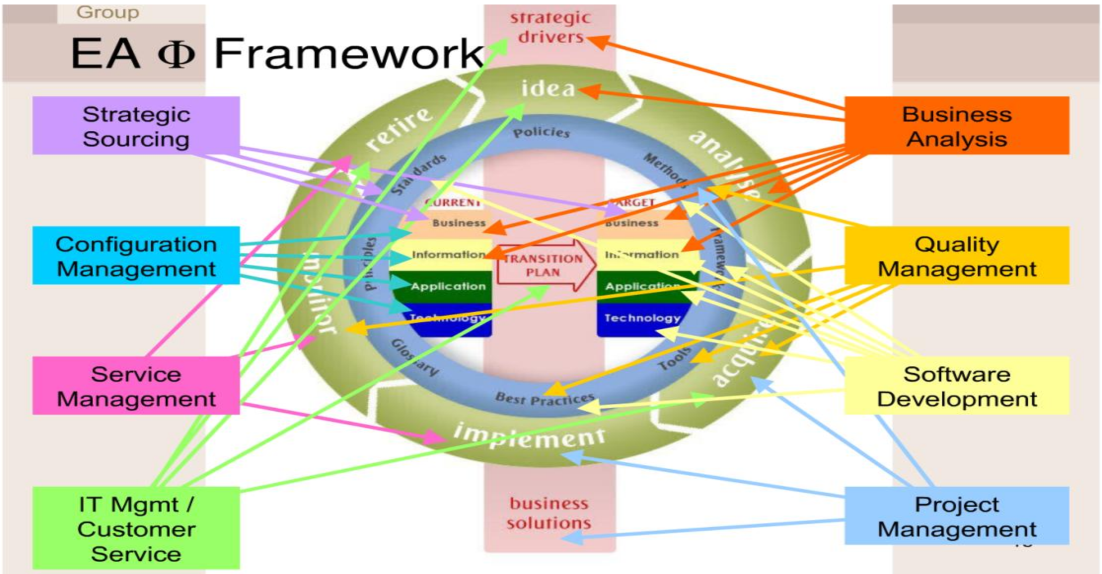


Figure 5.4 Modified TOGAF [I-2]

A health standardisation body extended TOGAF and used it to develop an interoperability framework [I-4]. It had three layers: organisational layer, informational layer, and technology layer (see Figure 5.5). It was developed for the Australian Health Sector to improve interoperability in the sector. The framework:

> *[makes] sure that the community with which we have a set of stakeholders is able to leverage a set of consistent artefacts in that there is a harmonisation towards a common practice* [I-4].

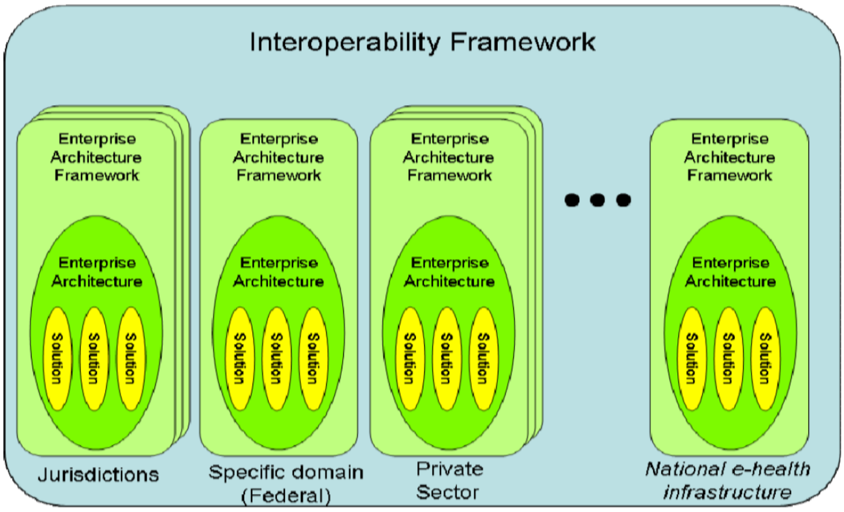


Figure 5.5 Interpretability framework and EA

TOGAF was adopted and customised at a large Australian bank to reduce complexity and improve agility [I-6]. It had three layers: business, information, and technology architectures. Their approach shows that their architectural settings matured over time. At the beginning, the adopted EA approach was technology focused. Later, it was improved to be business-aware EA, to capture business capabilities and business processes, to define the future for the technology, and to align business and IT. Enterprise Architect [I-6] declared:

> *But effectively what we're trying to do is move from a more of a technology-centric approach to EA, to more of a business-centric approach to enterprise architecture where we're using the business capabilities as our main guide and using that to then map processes, map functions and then start to map technology components to that.*

Some participants also suggested that following a certain EA framework improves EA management.

> *I think the process is just hamstrung by too much documentation and also not following a formal framework of handing over artefacts to different groups* [I-10].

And some participants suggested that some EA frameworks are still not supporting SOA design, which conditions (restricts) SOA's integration into EA. Chief enterprise architect [I-1] stated

> *he first major problem I have with Zachman and other EA framework is that they don't actually contain any artefacts, business design artefact that you can properly use to design an SOA architecture.*

EA consultant [I-20] stated:

> *Now it's the same thing with TOGAF as well, it is this all encompassing framework...it doesn't actually say well here's an example of a data architecture or here's an example of a physical architecture or conceptual architecture. It doesn't actually say how to develop it.*

Further, EA consultant [I-10] reported a case where TOGAF as a framework was used in some parts of the organisation, while other parts practiced EA without a defined EA framework:

> *They use TOGAF but it's not prevalent throughout organisations so probably some of their team use TOGAF but I'd say in general they don't use a framework.*

EA consultant [I-20] argued that EA could be implemented to address a segment of the business when needed.

> *I could be doing segment architectures where there's a business need over here and we've got to do what is called just in time enterprise architecture. So enterprise architecture but applied to a specific business problem.*

In conclusion, the findings indicate that adopted EAs were shaped by previous actions that preceded SOA's introduction. Some frameworks were suggested to have limited capabilities supporting SOA's integration. EA frameworks have different characteristics in terms of structure and scope (light, comprehensive, and partial models), which Table 5.5 shows. In some cases, an old EA framework was abandoned and a new one adopted due to lack of support of organisational activities ([I-13] and [I-15] was planning to switch to TOGAF). Some of the adopted EA frameworks were changed, modified, and adapted, which consequently become a conditional factor (enabler or constraint) for the next iteration of action related to EA.

5.4.2 EA Objectives

The interview analysis indicated that EA was adopted for different purposes that could be classified into: strategic, operational, IT, and governance-oriented approaches. These objectives act as a conditional generative mechanism for the next actions related to EA and, in particular, SOA's integration into EA. In other words, these objectives enable certain actions and obscure others by shaping the conditioning phase according to the objectives that initially drive EA adoption.

Strategic EA focuses on strategic alignment, and business and IT alignment. Operational EA addresses operational activities such as communication and decision-making improvements. It tends to reactively identify gaps due to its focus on operational issues. EA in some cases is adopted to manage the IT architecture. EA's other objective is to be adopted as a governance practice. It is used to establish architectural policies and to govern projects.

Figure 5.6 shows EA objectives and the number of participants' responses that fit under each category. Some participants reported some objectives that fit under more than one category. For example, enterprise architect [I-3] reported that EA was used for business/IT alignment (strategic) and to improve communication (operational), while, in another case [I-15], EA was used only for IT governance purposes.

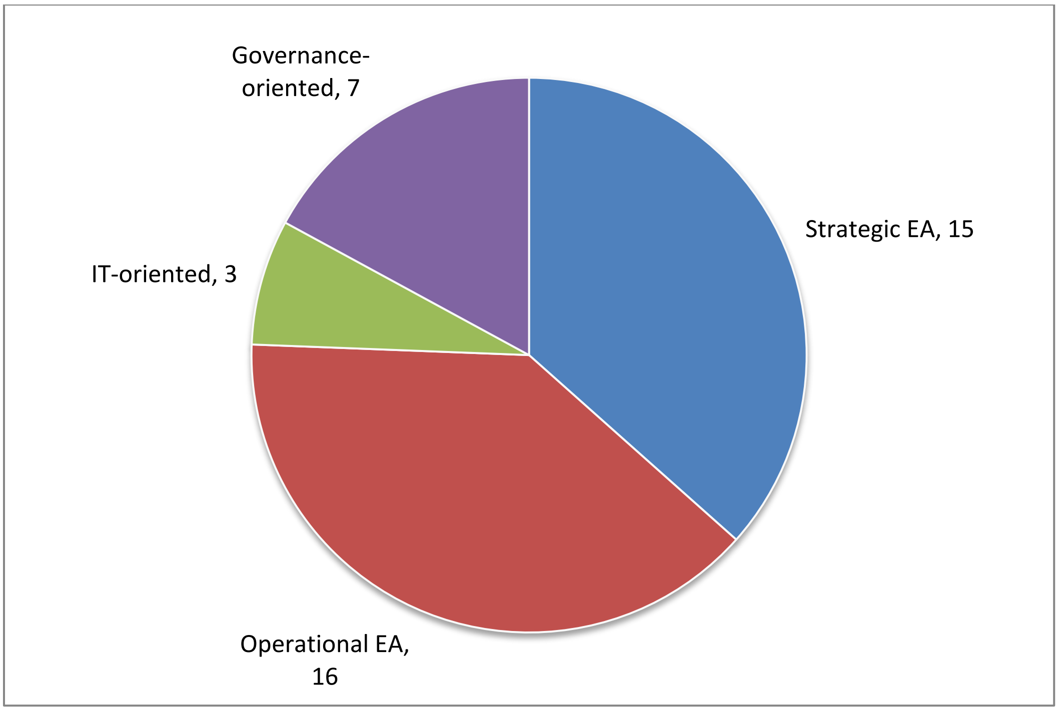


Figure 5.6 EA objectives

Some participants argue that the view of EA drives its purposes, which influences SOA's integration into EA. For example, participant [I-4] reported that TOGAF was customised to build an interoperability framework and, participant [I-2] reported that EA was integrated with other planning and management practices in the organisation, such as strategic sourcing, service management, software development, and customer services to achieve strategic, operational, and governance objectives. Further, EA was adopted to manage IT architecture [I-14].

In summary, the interview findings show that EAs were adopted differently to achieve varying objectives and that these objectives normally shaped EA-related activities and its use. Once these objectives are set up and embraced, they become a conditional factor for the next rounds of EA-related activities such as SOA's integration into EA. An EA developed to only provide *"high level governance of IT practices"* [I-15] will be a different conditional factor than an EA developed to *"[align] business design, people design which often gets forgotten [and] technology design to ensure that all three together deliver on business intent"* [I-1] or an EA adopted for *"not building a unified model [but] making sure that the business is growing"* [I-5].

5.4.3 EA Maturity

As Chapter 4 presents, EA maturity is hypothesised to be a conditional generative mechanism that influences SOA's integration into EA. The interview findings support the relevance of the conditional influence of EA maturity on SOA's integration into EA. Fourteen of the participants (nine EA practitioners and five EA consultants) expressed the importance of higher EA maturity for better architectural practices in general, and particularly for SOA's integration. For example, EA consultant [I-17] suggested there is a link between architectural practice and the maturity of EA. He reported that TOGAF architecture development methodology (ADM) was used differently in alignment with the level of EA maturity. He stated that

> *there's very few people who use the whole ADM but typically people will align their architecture practice at their level of maturity with the relevant aspects of the ADM [I-17].*

EA maturity was also reported to be a critical factor of developing sustainable EA practices and achieving EA benefits. Enterprise architect [I-6] stated that:

> *to make it* [EA] *sustainable and to derive the value from it the answer would be yes because enterprise architectures are all about planning and strategy effectively.... So you could implement a service's model without it but what you might implement may not actually be able to evolve over time as your organisation evolves.... It's critical to sustainability over time* [I-6].

EA consultant [I-16] supported the relationship between EA maturity and SOA's integration. He argued that the maturity of both EA and SOA has to be on similar levels to achieve better integration. He reported:

> *you'll find different levels of maturity on both sides. Sometimes the EA is more mature, sometimes the SOA is more mature and one thing we always say is when disciplines want to work together there can't be too much of a difference in maturity... let's say higher than two* [I-16].

Building on this view, EA consultant [I-12] also asserted that introducing SOA into an environment where EA practices are less mature constrains SOA introduction:

> *if you don't have a mature EA capability in organisation, SOA is very unlikely to even get a look in.... If you don't have a good EA function and have it engaged and accountable, in my opinion you cannot get SOA properly implemented* [I-12].

Most of the participants reported low levels (level one or two) of maturity. Among the EA consultants, three EA consultants [I-9, I-12, and I-16] reported that, based on their engagements with multiple EA projects, EA maturity was still at an early stage. EA consultant [I-12] stated

> *a lot of the clients we worked with, they have not matured their enterprise architecture function".*

This argument brought up by EA consultants is also supported by some EA practitioners. For example, enterprise architect [I-13] exemplifies the impact of low EA maturity on documenting SOA-related artefacts. He said:

> *We're not strong, we haven't been strong in publishing a huge range of artefacts in this space* [EA]*, we've got limited documentation out there and available to these groups.... So we've not created specific SOA documentation* [I-13].

EA maturity is measured using multiple dimensions, as presented in chapter two. This thesis examined these dimensions to identify their relevance to SOA's integration into EA. Table 5.6 shows examples of quotes supporting their relevance to SOA's introduction.

Table 5.6 EA maturity

Maturity dimensions	Quotes
EA documentation	"It [SOA] can't exist without a clear, to me, a strong process model and information model to support it. If you don't have either of those two things, you really can't do SOA. In my mind, you can't do SOA properly" [I-12].
EA planning	"The whole idea of the enterprise architecture is to improve the maturity and identify the strategy roadmaps and set the path of an organisation to better improvement. If that's not happening then there's

	something wrong with the EA team or the communication within the EA team" [I-18]. "JABOWS, it stands for just a bunch of web services and it's a derogatory term for SOA architecture, SOA that has not been designed using business capability modelling and enterprise architecture approaches" [I-1].
EA governance	"EA governance is the thing that is going to make this [SOA] work, so SOA is part of your overall enterprise architecture governance.... If you don't have a good EA function and have it engaged and accountable, in my opinion you cannot get SOA properly implemented" [I-12].
EA team and resources	"A lot of enterprise architects come from an IT background but a real enterprise architect is one who is able to step over into the business and work confidently with the business and has actually developed a whole range of skill sets which are non-technical" [I-20]. "Most of them [EA and SOA teams] they're very separate teams. They may have an integration architect, which SOA is one part of what they're doing and that person may live within the enterprise architecture team, they may not" [I-9].
EA evaluation and maintenance	"There's plenty of organisations that have a disappointing encounter with EA because they take a framework, start implementing the framework but forget about all the other things that need to be done" [I-16]. "Enterprise architecture is not a one-time activity, it's a continuous activity, we basically find that in our experience, that activity is generally done one time and it gets left in the corner" [I-7]. "The only other time you may do a refresh is if there's a significant project which has known to be impacting the enterprise architecture" [I-9].
EA business support	"You definitely need business support for your enterprise architecture and you need to understand it and be behind it. And your best practice for enterprise architecture is that it actually sits in the business and not in IT" [I-1]. "So to be successful embed enterprise architecture into the culture [so] that everyone, especially executive level management, use enterprise architecture to involve their decisions" [I-19].

In conclusion, this section examines aspects related to the architectural conditioning. The interview results showed different architectural conditions of EA (pre-existing EA) that have a conditional influence on SOA's integration into EA.

Besides the hypothesised generative mechanism (EA maturity), two other generative mechanisms emerged from the data: EA framework and EA objectives. The results show that the used EA frameworks vary. Some of the reported EA frameworks were based on well-known EA frameworks, and others were in-house developed. They had different structure and scope. The results also show that EA was implemented prior to SOA's introduction, and that EA framework and methodologies have gone through some cycles of change in these organisations prior to SOA's introduction. For example, TOGAF was appropriated (case [I-4]) to build a national health interoperability framework. In case [I-6], TOGAF was used as a foundation to internally develop a customised EA framework which that started in the IT department to manage the organisation's technology, and it was later extended to manage more business artefacts. The way these frameworks implemented, customised, and appropriated becomes a conditional factor for the next cycles of EA activities (here: EA evolution).

The findings also indicate that EA was adopted for various objectives based on how it was seen. This thesis classifies these objectives into strategic, operational, IT and governance-oriented EAs. These objectives became a conditional factor for coming cycles of EA change. The findings also confirm the importance of the conditional influence of EA maturity on EA sustainability and evolution.

While this section presents the findings relevant to the architectural conditioning analytical phase, Section 5.5 presents the findings pertinent to the architectural interaction (SOA's introduction) phase.

5.5 Architectural Interaction: SOA's Introduction

The second analytical level in Archer's (1995) morphogenetic theory is architectural interaction. In this thesis, this level is specifically about "SOA introduction" to emphasise the thesis's topic. It analyses the action taken (introducing SOA) that may cause EA to evolve.

Participants discussed different SOA implementations that were introduced to their organisations or to their clients' organisations (EA consultants). The purpose of the analysis is to examine SOA introduction related generative mechanisms, of the a-priori model, that have an influence on SOA introduction, and identify more generative mechanisms (if any) from the data. EA consultant [I-9] argued that SOA introduction is "usually driven according to the local organisation".

This section examines the generative mechanisms that influence SOA's introduction. It identifies three action-formation generative mechanisms from the literature: view of SOA, SOA perceived benefits, and SOA scope. Also, three new action-formation generative mechanisms emerged from the data: SOA governance, SOA design, and business/IT collaboration.

5.5.1 View of SOA

Analysis of the interview data confirmed that, as suggested earlier in the a-priori model, there are diverse views of SOA that may influence SOA's introduction. A classification identified from the literature was undertaken to classify views of SOA: software components, emerged software architecture, support of business processes, enterprise service architecture, and adaptive architecture. Supporting evidence emerged from the data to support the diversity of SOA views. However, none of the participants reported a case where SOA was seen or adopted as an adaptive architecture. These supported views of SOA are presented in the following paragraphs.

Table 5.7 Reported views of SOA

Views		Reported by	Criticized by
Technical-oriented views	Fine grained software components	6	7
	software architecture	4	3
Business-oriented views	Business processes support	4	None
	Enterprise service architecture	4	None
	Adaptive architecture	None	None

Prior to the examination of each individual view of SOA, the following paragraphs examine them from a broad business versus IT-oriented views. Four practitioners and two consultants reported cases that were adopting very technical SOA (web services and ESB). For example, EA consultant [I-12] declared that he had been involved in many SOA implementations where SOA was being implemented as an integration approach. He argued for wider perspectives of SOA. He declared:

> *We've been involved in a number of attempts, where organisations have tried to put SOA in but as I said what it comes down to, because the organisation sees it as purely an integration issue, it doesn't end up being SOA. It ends up being an integration approach, not an organisational* [or] *architectural pattern* [I-12].

However, eleven participants (five practitioners and six consultants) argued that IT-only perspectives of SOA restrict SOA's potential. For example, EA consultant [I-17] commented that such a limited view results in only partially achieving SOA's promises by focusing only on SOA's technical aspects. He described people's approach to SOA introduction:

> *I'm going to buy ESB then I'm going to build a bunch of services and I'm going to execute them. However that doesn't mean I'm a service oriented enterprise... so if I'm just using web services in the technical level that doesn't mean that I'm a service oriented enterprise* [I-17].

Other participants argued that SOA requires an understanding of business architecture and the use of business models in order to effectively design SOA and implement it. If SOA is limited to the technical level, the resultant architecture will be sparse software components that are not designed in alignment with business architecture. Therefore, SOA would not be delivering its promised value:

> *SOA requires you to determine things from a business design concepts. It cannot be designed inside technology. You will not get that analysis without understanding business design or understanding those concepts of how it was described. Without understanding that you will never be able to design your SOA properly you will just end up with a set of web services that are of no use, you won't get reuse for example* [I-1].

Chief enterprise architect [I-2] challenged such an IT-only perspective of SOA. She argued that the enterprise service bus implementation is just another tool. She argued for a mature perspective of SOA that is aligned with business and information side of EA:

> *… even if people say that they're using an ESB you know, whatever be it TIBCO or IBM Websphere whatever it is, again they're just using an ESB and nothing else. You know [when] you start getting up into the information and business layers you start achieving true service orientation* [I-2].

The reports above confirms the different perspectives of SOA that may influence SOA's introduction, and Table 5.8 shows some supporting quotes for each perspective.

Table 5.8 Views of SOA and supporting quotes

View	Cases	Quotes
Software components	6	"JABOWS stands for just a bunch of web services and it's a derogatory term for SOA architecture, SOA that has not been designed using business capability modelling and enterprise architecture approaches" [I-1]. "I actually don't believe how we're describing SOA as being integration technology and web services actually representing enterprise architecture" [I-6].
Software architecture	*4*	"Most organisations who use enterprise architecture don't include a service view because they believe that that is part of their solution architecture" [I-17]. "a lot of organisations split SOA and BPM. And so once you do that, your architecture function in your organisation tends to go down the path of building services, largely out of context of processes" [I-12]. "The gentleman who was responsible for IT on the board and the IT advisor on the board had advised the board that they didn't need to worry about SOA, it was just an IT thing about IT reuse and they needn't bother worrying about it and didn't bother to explain" [I-1].
Business process support	*5*	"What we strongly believe in SOA and EA is that it has to be process driven. I mean that's the strong belief we have in the company and we strongly believe it because process ties all the other architectures together" [I-19]. "Once these processes are documented and put into a tool, you can share a full workflow of processes so that people actually at the business side of it can actually buy in and start to understand where their processes are going, then you can create a service associated around those processes" [I-18].
Enterprise service architecture	*4*	"SOA is fundamentally about your business design and designing for your agility" [I-1]. "Unless you can look at your organisation as a collection of

		services that support the business then you're not service orientated" [I-17]. "A service-oriented approach ties business requirements to business services and processes" [I-4].

The discussed findings suggest that an adopted view of SOA has implications for SOA's integration into EA. For example, the very technical perspective of SOA is often not integrated into EA as enterprise architect, [I-6] stated:

> *I actually don't believe how we're describing SOA as being integration technology and web services actually representing enterprise architecture*

It was also supported by EA consultant [I-17] when stated that

> *most organisations who use enterprise architecture don't include a service view because they believe that that is part of their solution architecture*

On the other hand, business perspectives of SOA are mostly integrated into EA. For example, EA consultant [I-12] argued:

> *if you're talking EA, SOA, you can't really have that discussion completely without going EA, SOA and business process, because the whole three of those have to co-exist before you get a fully mature and understood SOA outcome.*

This view was supported by chief enterprise architect I-2] arguments that:

> *SOA is… a design paradigm… that takes into account all those four layers of the enterprise architecture.*

As hypothesised in Chapters 2 and 3, the interviews supported that there are diverse views of SOA that range from very technical to very business-oriented views that influence SOA's introduction. These perspectives are: fine-grained software components, software architecture, business processes support, and enterprise service architecture. According to some participants, the very technical views are considered undeveloped perspectives of SOA, and the adoption of SOA from such a perspective does not represent the wider nature of SOA nor achieve the full potential of its implementation.

5.5.2 SOA Perceived Benefits

The participants reported different benefits of SOA on IT, process, and strategy levels (see Table 5.9). These benefits are classified using the SOA benefits classification model that Chapter 2 presents.

Table 5.9 Reported SOA perceived benefits

SOA benefits	No. of sources	Benefit levels (IT, process, strategy)
Agility	13	Strategy
Reuse	9	IT
Business-IT alignment	6	Strategy
Reduce maintenance costs	5	Process/IT
Business process improvement	5	Process
Improved IT integration	5	IT
Increased availability of Information	4	Process
Reduce time to market	4	Strategy
Improve communication	3	Process/strategy
Reduce complexity	3	IT
Facilitation of software development	3	IT

In regard to the SOA introduction drivers, enterprise architect [I-4] claimed that SOA is always driven by what the business wants to achieve from SOA introduction. He reported that "*it [SOA] always has to be driven by what you're trying to achieve in the business*" [I-4].

However, justifying SOA's introduction based only on IT benefits is criticised by chief enterprise architect [I-1]. He said that organisations often justify SOA based on IT-related benefits such as reuse, and argued for a business-oriented justification of SOA. He said: "*if you try to justify SOA as many organisations do on the basis of IT reuse and minimising IT cost it will never pay for itself*" [I-1].

Some participants also suggested that the realisation of SOA benefits increases when SOA's introduction is planned. Enterprise architect [I-15] commented on the limited SOA benefits that his organisation (a bank) had achieved. He said the bank's adoption of SOA was partial and that SOA was viewed as an integration approach:

> *Well we are not getting much actually because our adoption of SOA is partial and this is the road map now to reach better maturity of SOA. We did not get all the benefits of SOA, but what we got from SOA up to now is very important in what the bank have reached.*

In summary, the findings support the argument that SOA's introduction is influenced by the perceived benefits (IT, process, and strategy-related benefits). IT-based justifications drive SOA to fix IT-related issues and to focus on IT benefits such as reuse. Some participants criticised IT-based justification and considered it to restrictive SOA's wider potential. On the other hand, strategic-driven SOA increases the potentials of SOA's introduction to achieve wider benefits.

5.5.3 SOA Scope

The interview findings confirm that there are different scopes of SOA introduction. In particular, four EA consultants and three EA practitioners explicitly mentioned that SOA introduction scopes are different. EA consultant [I-17] stated that SOA scope differs depending on the context of the organisation. Further, EA consultant [I-19] stated that the scope of SOA introduction is potential indicator of the maturity of SOA practices. He argued that a higher SOA maturity is achieved when SOA is introduced at a strategic level in alignment with EA.

Six participants report that their organisation introduced SOA at the project level. SOA can be adopted in small fragmented projects to fix integration issues or replace point-to-point integrations. EA consultant [I-12] pointed out that there are some issues associated with such a scope. For example, it leads to implementations of SOA projects that are not aligned with previous SOA work. He explained that:

> *what tends to happen is if the business is not going through a transformational project and they're just driving it from project to project, then the projects override any architectural type decisions that you might try to do bottom-up* [I-12].

Another EA consultant [I-7] added that such an approach (project-based scope) often doesn't lead to a successful implementation compared to an organisational-level scope because SOA requires a much larger scope of implementation in order to achieve its promises. He argued that:

> *We have not seen a lot of successful customers. The primary reason is because SOA cannot be done at a project level. It has to be done at the unit level, at an organisational level* [I-7].

A comment given by EA consultant [I-9] indicated one reason for adopting project-level SOA. He argued that the larger SOA scopes require the replacement of existing systems and large spending. He stated that:

> *one of the reasons they haven't done that* [wider SOA scope] *is because many of them have existing systems and replacing those existing systems is not cost effective*.

Five participants reported that their organisations adopted SOA portfolio scope. EA consultant [I-10] claimed that SOA needs to be adopted at the portfolio level where it gets enough support to deliver its objectives:

> *It* [SOA] *has to play at the portfolio level. Because at the project level, you don't have enough support to support whatever you do in SOA*.

Chief enterprise architect [I-2] further argued for a wider portfolio-based SOA scope. She explained how the chosen SOA scope affects SOA implementation and also its promised benefits. She suggested that:

> *You look at your whole portfolio and look at how service-oriented is my whole project portfolio because if you are moving your portfolio to more service oriented solutions you are then finally going to deliver the benefits of SOA to the organisation. But if you're just doing one or two projects you're not really delivering SOA benefits*.

Other participants (three consultants and one practitioner) also argued for a larger scope of SOA introduction; that is, for it to encompass the whole organisation by identifying services on all levels from business to technology and applying service thinking as a design philosophy. EA consultant [I-12] believed that SOA introduction scope at an organisational level is fruitful. He argued that SOA is

> *an architectural pattern that needs to be applied at a very high level in the organisation. So it needs to be applied at an enterprise*

> *architectural level and an organisational level before it can really be effective.*

Another EA consultant [I-17] also argued that SOA implementation has to be at an organisational level to successfully implement SOA:

> *we have not seen a lot of successful customers. The primary reason is because SOA cannot be done at a project level. It has to be done at the unit level, at an organisational level.*

EA consultant [I-10] asserted that SOA needs to be introduced on a large scale in order to achieve its potential benefits. A small project or multiple projects that do not follow the same strategy are not capable of delivering SOA in the right way. He noted that:

> *It's* [SOA] *a strategy, not a project. Because SOA doesn't, it absolutely doesn't work if it's a simple project because the only way SOA really works is if all of the projects follow the same philosophy. It's, you know, it's a philosophy; it's not just an activity or a project that can be ticked off. SOA strategies take years to implement.*

In summary, the interview findings confirm there are different scopes that impact how SOA is introduced. The findings indicate that a wider scope of SOA introduction is likely to result in a successful SOA implementation and wider realisation of SOA's benefits. Some also indicated that smaller scopes of SOA introduction often tackle IT issues and are isolated from business processes and EA.

5.5.4 SOA Governance

This section discusses an inductively emerged action-formation generative mechanism that influences SOA's introduction (namely, SOA governance). It is defined as the planning of SOA's direction, the management of services lifecycle, and the establishment of SOA standards, policies, roles and responsibilities.

Eight participants (5 practitioners and 3 consultants) reported that SOA governance influenced SOA's introduction. Some participants asserted that the lack of standards, policies, and control around SOA introduction was a major hindrance for implementing SOA. For example, EA consultant [I-17] reported that a survey among their clients found that more than fifty percent of SOA adopters had issues around SOA governance:

> *Among the things that were holding them back was, more than half of the respondents basically described a lack of SOA governance. So if we think about it, that lack of governance is not anything to do with technology, the lack of standards, policy, ownership and control around the way that technology is deployed.*

Enterprise architect [I-13] further emphasised that well-established SOA governance is significant for successfully introducing SOA into an organisation. It is a key foundation to realise SOA introduction objectives:

> *Governance is the single biggest factor that's crucial to SOA's success.... it* [SOA] *won't be successful unless you can govern it properly so governance is by far the most important thing that you'll ever do.*

SOA governance maturity is also highlighted by architecture manager [I-14] who challenged the success of large initiatives of SOA without mature SOA governance practices around it. He argued that *"You cannot just commit your organisation to be a total SOA where your governance layer is still not that mature yet"*.

Enterprise architect [I-6] also emphasised SOA governance's impact on SOA introduction. SOA introduction needs solid governance practices around it in order to keep SOA implementation on track and avoid inconsistency. He mentioned that:

> *You need to give it* [SOA] *a strong planning, strong direction, and strong oversight. And one of the things that we're facing into is that we did it a little bit piecemeal in certain parts of the organisation so that we now have different definitions of things or we're not a hundred percent aligned.... therefore, avoid inconsistencies in your model by being very clear about what your model is and by coordinating the roll out of it so that it is done consistently across the organisation.*

EA consultant [I-19] argued that SOA governance is about managing the lifecycle of services. The lack of governance leads to service duplication and versioning issues. He stated that:

> [SOA] *governance is all about how you control the life cycle of services, who can access it at what point in time and how collaborate across projects... the governance is key otherwise you would have a whole bunch of duplicates, you'd have versioning issues, you'd have two or three versions of the same service in operation.... You need to govern exactly how those services have been thought about, designed, created, and retired.*

SOA needs clearly defined roles and responsibilities as well as the establishment of a governance committee to monitor its introduction. Chief enterprise architect [I-1] argued for more than a simple establishment of SOA governance foundations. He stressed the importance of having an active governance committee to push SOA governance further and monitor its function. Once SOA governance is established, it has to be effectively used to govern, guide, and monitor SOA activities. EA consultant [I-20] also raised the importance of having an actively involved SOA governance committee. He highlighted the importance of clearly identifying SOA's roles and their boundaries and duties. He argued that:

> *When you get down into SOA you have got to work out who is going to be responsible for what. Is it just going to be the ICT people or is there going to be other people that are involved, some people from a business unit within the organisation that have to be involved.*

For instance, chief enterprise architect [I-1] noted that, when introducing SOA into an organisation, a service owner role needs to be created. He argued that services need to be owned and managed. At the bank where he was leading EA, services were owned and managed by the different business capabilities owners that services belong to. A business capability owner is responsible for developing, maintaining, and versioning services that are associated with their business capability.

In summary, the interview findings suggest that SOA governance is an important foundation for SOA introduction. It acts as an action-formation mechanism that affects SOA's introduction. The adoption or lack of SOA standards, guidelines, and governance activities impact SOA's introduction. Some participants argued for mature SOA governance practices in order to successfully introduce SOA.

5.5.5 SOA Design

This section presents the findings related to SOA design, which is an inductively emerged action-formation generative mechanism related to SOA introduction. SOA design refers to how SOA is designed in terms of how an organisation uses reference architecture, roadmaps, service identification methodologies, and services classifications.

Many participants (7 practitioners and 8 consultants) show that their organisations approached SOA design very differently. SOA design acts as an action-formation mechanism that influences the way SOA is introduced. It has multiple facets such as SOA reference architecture, SOA roadmaps, service catalogues, and services classification models. SOA is introduced by employing certain design aspects that affect its implementation. These design aspects are discussed in the following paragraphs.

Two practitioners and three EA consultants argued for the use of explicit SOA reference architecture. SOA reference architecture is a blueprint that describes SOA building blocks. For example, EA consultant [I-17] argued that the awareness and development of aspects such as reference architecture is essential when introducing SOA. He noted:

> *So the technology if you like is only going to be able to support it* [SOA].... *But there's a lot of work that needs to be done to actually set that up. So you would sort of start to look at things like building out the reference architecture.*

Enterprise architect [I-19] argued that SOA implementation should be based on solid architectural foundations. For example, some SOA initiatives shift its focus towards fixing implementation issues and overlook the underpinning foundations needed in advance. Ideally, SOA projects should be built on the same reference architecture that describes its SOA's environment and builds its foundation. He asserted that:

> *The biggest side effect I see people not using SOA right way is that they basically bought the SOA technologies and they built themselves black boxes. So they basically run projects and they build all this stuff and it is not architecture, it is implementation…. The only thing that I would caution is to put a lot more emphasis on laying down the meta-model, laying down the foundations, what we call the reference architecture of SOA in those projects.*

EA consultant [I-18] also described their decision to use in-house developed SOA reference architecture instead of TOGAF's one. He stated:

> *We suggested the TOGAF framework, particularly the TOGAF 9 which has the services architecture in there. It's not a very easy fit…. I haven't seen one* [EA framework] *yet that does the services component really well, most of it* [service architecture] *is developed out of our own reference architecture which we designed ourselves.*

Further, three consultants and two practitioners reported the value of developing a clear SOA roadmap in order to better introduce SOA. SOA introduction, like any other initiative, is better managed when it starts with a well-defined target or a goal to attain. For example, EA consultant [I-17] argued that SOA requires a transparent and properly designed roadmap. The roadmap guides SOA introduction based on organisational objectives: "So firstly, understanding what the strategy of the organisation is and then basically putting in place a road map of where we are doing and what we do actually need to do now".

Chief enterprise architect [I-2] also noted the importance of an SOA roadmap: *"In order to adopt SOA you also need to have your SOA blueprint that describes to the organisation how you're going to move"*.

EA consultant [I-17] also declared that building the right roadmap for SOA initiatives is among the issues that face their surveyed customers. He claimed that SOA is often introduced as a technology, but that the guidance of where or how to improve it is omitted. He noted that:

> *Out of the top three things that they replied was the difficulty in building an SOA roadmap. So what that meant was that they were in a position where they were utilising the technology but they had no way of knowing where they should improve, what they should do*

> *differently over time to actually build out the effectiveness of their technology solutions.*

Another participant, [I-19] revealed that understanding of the roadmap among SOA stakeholders is a key indicator of the maturity of SOA practices in an organisation. He commented that: *"[SOA] maturity is how much the key stakeholders that own the SOA understands the roadmap"*.

Moreover, participants reported that their organisations used diverse services classification models. EA consultant [I-12] argued that it is fundamental to have a clear classification and definition of services because of the different characteristics of each service type. He recommended that *"You need a very clear services layering model because they're, not all services are the same... there will be layering of services"*.

EA consultant [I-20] claimed that the service concept is confusing and could mean different things for different people. Thus, the use of a service classification model would reduce that ambiguity and help improve communication between stakeholders. Yet, the findings show that participants defined and classified services differently (e.g., business services, information services, and infrastructure services) (see Table 5.10).

Table 5.10 Example of service classifications

Sources	Service layering models
I1	Enterprise services – component services
I2	Business services – application services – technical services
I3	Business services – application services – infrastructure services
I4	Service channels -- business services – service – service components
I5/I6	Business services – business aligned technical services – technical services – asset services
I7	Business services – process services – integration services – infrastructure services
I12	Business function services -- process services -- data services -- technical services
I19	Business services – entity services – utility services

The findings indicate that services identification process can be approached differently using different (e.g., bottom-up, top-down, and meet-in-the-muddle) strategies. Also, the starting point for the service identification process varied (see Table 5.11). Some organisations used business capabilities, while others used business processes or applications.

Table 5.11 Service identification methodologies

Sources	Service identification
I-1	Top-down : business capabilities level
I-2	Top-down : business processes level
I-3	Top-down : business functions level
I-4	Top-down : business processes level
I-5	Top-down : business processes level
I-10	Bottom-up : applications level
I-11	Top-down : business processes level
I-14	Bottom-up : applications level
I-15	Bottom-up : applications level
I-17	Top-down : business capabilities level
I-20	Top-down : business processes level

EA consultant [I-17] argued that the bottom-up approach is the most used approach because it is easier. However, EA consultant [I-12] argued that a top-down services identification strategy is better for introducing SOA at an organisational level. SOA implementation using a bottom-up approach often lacks top management support, and, therefore, fails to change the organisation at higher levels. EA consultant [I-9] highlighted that every service identification approach has its own advantages and disadvantages. A bottom-up approach is quick to achieve some results and reuse, but it doesn't offer the strength to redesign a business because it is mostly IT-oriented. On the other hand, a top-down approach does support the redesign of business processes, but it is consequently hard to align them with existing IT applications and systems.

The classified services need to be managed in a service catalogue that keeps track of services, their descriptions, dependencies, rules, and URIs when they are automated. Seven participants highlighted the importance of setting up a service catalogue. It is a crucial management tool to make services available for discovery, use, and reuse. For example, EA consultant [I-10] stressed that managing services well is critical. EA consultant [I-16] showed how the catalogue provides some sort of traceability between high-level capabilities in EA and their implemented services. He said:

> *The service's catalogue would have a counterpart in the enterprise architecture so that there is a link to what we describe on the*

> *enterprise level, and what we end up on a technical level is actually implemented services.*

To conclude, the interview data show that SOA design is a potential action-formation mechanism that influences SOA introduction. As some participants emphasised, mature SOA design that considers the aspects discussed above guides SOA's introduction, improves its maturity, and helps organisations to realise SOA's benefits. Particularly, the data show the importance of using SOA reference architecture to build the foundations for SOA to ensure consistency and clarity of SOA concepts and foundations. The data also suggest that a SOA roadmap is a key step in SOA design to ensure that SOA reliably progresses through its stages of introduction. However, some EA consultants reported that some SOA adopters did not employ SOA reference architecture and SOA roadmaps. The findings also show the implications of using a service classification model to define services and their granularity. They also exemplify the variety of the employed service classification models that have disparate services and varied levels of granularity. The findings also establish that organisations use different service identification approaches. A top-down approach usually uses business capabilities, business functions, or business processes as a starting point to identify services. On the other hand, a bottom-up approach usually starts from applications and decomposes them into fine-grained services. The findings also demonstrate the value of a service catalogue to enable services discovery, maintenance, and reuse.

5.5.6 Business and IT Collaboration

This section details another inductively identified action-formation generative mechanism that impacts SOA introduction called business and IT collaboration. The data suggested that business and IT collaboration influences SOA introduction. In particular, the level of business support, and the SOA's team settings, and their skills are suggested to influence SOA introduction.

Many participants argued that organisations should provide the required business support for SOA (5 consultants and 3 practitioners). They also noted the need for highly skilled team (a mix of business and IT people) to drive SOA's introduction (4 consultants and 1 practitioner). Three practitioners reported that their organisation's SOA team included only one or two integration architects.

Chief enterprise architect [I-2] highlighted the importance of having business and IT stakeholders involved in the services planning process. In her organisation's case, business and IT worked together to plan and implement SOA. EA consultant [I-10] also asserted that almost half of SOA activities occur at the business side of organisations. Therefore, he argued that SOA requires strong business support to direct both its business and IT activities:

> *There's a lot of interaction with the business in SOA. Forty, fifty percent of it is business side. So in order for SOA to work, the business must back it one hundred percent and they must have enough of a stake in it so they must be able to influence whatever activity that is happening in IT. So SOA I think works if you have a very strong business sponsor.*

Some participants argued that organisations need business support specifically when undertaking an SOA transformation initiative. For example, enterprise architect [I-6] said: *"If you're going to turn yourself into a service-oriented organisation, the business needs to be willing to come on that journey too"*.

Another level of business/IT collaboration is SOA team members and their position in the organisation. Some participants stated that their SOA teams were not officially defined. Some viewed the SOA team as a couple of developers positioned under the integration or solution architecture teams. For example, enterprise architect [I-13] revealed that his organisation's SOA team was one, technology-oriented person: *"Right now, that particular person is a strongly technology focussed individual so they understand the services infrastructure and the components pieces of the services infrastructure"*.

Enterprise architect [I-6] described their SOA team as being integration driven and most of SOA work and skills in his organisation was centred on integration aspects.

Further, some participants noted that a mix of business and IT skills are needed for introducing SOA. For instance, EA consultant [I-20] suggested that a team of experts on the organisation's business, information, application, and technology aspects should drive SOA and its implementation: *"Bringing together a team of people which are subject matter experts in their specific fields of, say, information application, infrastructure and business, is how you coordinate SOA into an organisation"*.

Some participants also argued for mature thinking around services. For example, EA consultant [I-17] indicated that SOA created a new set of concepts that impact organisational approaches to design and development. Service thinking is essential, and what is even more significant is the articulation of the thinking to be used in SOA's implementation. He noted that *"The biggest capability gap in most organisations is a lack of maturity and capability around service thinking and, you know, and ability to actually articulate those concepts"*.

Thus, he argued for improved organisational awareness of SOA and the need for conduct training before SOA's introduction. He mentioned that "Too many organisations go into SOA without properly, let's say, first of all preparing themselves and second of all training their people into what it actually means to be SOA".

EA consultant [I-16] reported the consequences of having an immature team to manage and implement SOA: *"I think the immaturity of SOA teams contributes to the confusion on how to do it properly"*.

To conclude this section, it seems that business support, the SOA (business and IT) team, and their skills all influence SOA's introduction. As suggested, a skilled, diverse business and IT team needs support from management to contribute to an articulated SOA introduction. For example, some participants argued that SOA needs robust business support. SOA introduction requires development of its vision and strategic drivers. Also, it needs business architects' participation to drive service identification using business models and business requirements. Introducing SOA requires a very skilled team to establish it, support it, manage it, and implement it. The findings suggest that the level of business support and SOA team skills and structure in an organisation influence SOA's introduction. The next paragraph summarises the wider SOA introduction section.

5.5.7 Summary

In summary, this section examines the six action-formation generative mechanisms (view of SOA, SOA perceived benefits, SOA scope, SOA governance, SOA design, and business and IT collaboration) that the interviews data suggested to influence SOA's introduction. These generative mechanisms collectively impact SOA's introduction. First, organisations introduce SOA based on diverse perspectives ranging from very technical to very business-oriented, and each influences the process differently. Second, the process is driven by diverse IT, operational, and strategic benefits. Third, SOA has been introduced using different scoping options such as project, portfolio, and enterprise levels. Fourth, the SOA governance aspect (i.e., the use or lack of reference architecture, policies, roles, responsibilities, and the management of the service lifecycle) influence SOA's introduction. Fifth, organisations approach SOA design differently, such as by the use or lack of the SOA roadmap, the establishment of a service classification model and service catalogue, and the employed service identification methodology. Each of these designs can influence the way SOA is introduced into an organisation. Sixth, business and IT collaboration, such as the level of business support, and the skills and position of the SOA team, is an important action-formation mechanism that influences SOA's introduction.

5.6 Architectural Elaboration

The third analytical phase of Archer's (1995) morphogenetic theory is architectural elaboration. This phase represents EA elaboration outcomes (here referred to as SOA's integration into EA outcomes) that result due to SOA's introduction. SOA's introduction either transforms of reproduces the pre-existing EA.

Most of the participants expressed that SOA's introduction influences EA and requires its evolution. For example, EA consultant [I-10] argued that SOA is part of EA and influences EA artefacts and models. He noted that:

> *SOA is a part of how you interpret or how you create your framework for enterprise architecture... it* [SOA] *obviously influences what artefacts get created, influences what models, influences the patterns that get created, and all these things which cascade down to the operational activities within the architecture teams.*

However, the findings show that SOA's introduction impacted EA to different extents. The findings support that pre-existing EA is either transformed or reproduced at one or many of five levels (outcomes). Three of these levels (outcomes): business architecture, information systems (IS) architecture, and technology architecture were identified from the literature as part of the a-priori model. Two other levels (outcomes) emerged from the data: (1) EA governance, and (2) EA methods and tools. Table 5.12 summarises these levels of EA evolution and their definitions.

Table 5.12 Levels of SOA's integration into EA (EA elaboration)

EA elaboration levels	Transformation	Reproduction
Business architecture	SOA is integrated into business architecture. It accommodates related SOA elements business services, service description, service channels, SOA vision, drivers, service actors, SLAs, and SOA vision.	No changes to the business architecture
Information systems architecture	SOA is integrated into information systems architecture. It accommodates relevant SOA elements such as application services, service descriptions, SLAs.	No changes to the IS architecture
Technology architecture	SOA is integrated into technology architecture. It accommodates SOA elements such as technology services, service interfaces, messages, and services monitoring elements, services security elements and physical technology components.	No changes to the technology architecture
EA governance*	SOA governance is integrated into EA governance standards, committees and practices.	No changes to the EA governance

EA methods and tools *	SOA methods and tools are integrated into EA methods and tools.	No changes to the EA methods and tools

SOA's integration into each of the five levels are discussed in the Sections (5.6.1 to 5.6.5). The findings show that most of the cases that reported SOA's integration into the business architecture also reported SOA's integration into both the IS and technology architectures. Table 5.12 shows the cases that reported SOA's integration into EA at each one of the five levels.

Table 5.13 Cases reported EA evolution

SOA's integration into	Explicitly mentioned by
Business architecture	I-2, I-3, I-4, I-16, I-17, I-19, I-20 (three practitioners and four consultants)
Information systems architecture	All the cases above, plus I-1 , I-9 and I-11 (two practitioners and one consultant)
Technology architecture	All the cases above, plus I-13, I-14, two practitioners. Case I-3 has not integrated SOA with their technology architecture yet
EA governance	I-1, I-2, I-12, I-16, I-18, I-20
EA methods and tools	I-1, I-2, I-7, I-9, I-16 (two practitioners and three consultants)

5.6.1 Business Architecture

Seven participants reported that their organisations' business architectures were transformed (SOA integrated into their business architectures) (see Table 5.13). First, the business architecture has elements that SOA uses. SOA also changes some of the business architecture elements. For example, business capabilities and business processes are usually used for service identification. The interview findings show the use of business capabilities, business processes, and business functions to drive SOA and service identification processes. They are used to identify services to be decomposed and implemented on lower layers of EA (IS and technology). For example, according to EA consultant [I-16] SOA is an architectural style that is used to decompose the business capabilities of the organisation: *"Service-*

orientation, as a style, allows for different structures and different decomposition of your business capabilities".

Chief enterprise architect [I-2] presented another example of SOA's integration into business architecture. She argued that ultimate service-orientation value is hard to achieve if the information and business layers of EA are not involved in SOA. In her organisation's approach, SOA was dissolved in their EA and the differentiation was almost imperceptible. She stated:

> *We don't differentiate between SOA architecture and EA, you know for us, SOA is very much part of the enterprise architecture... You know until you start getting up into the information and business layers you start achieving true service orientation.*

SOA integration into business architecture often includes integrating SOA into other EA layers. EA consultant [I-19] argued that SOA is a strategically driven initiative that carries SOA across all EA layers:

> *It* [SOA] *is an intentional and strategic move towards turning the organisation into a service oriented view. So you encapsulate that into all sorts of things. You have processes involved, you have systems involved, you have data involved you have even technology infrastructure involved, they all can be turned into SOA at different layers.*

This view is also supported by another argument from participant [I-17]. He confirmed that EA manages different elements of an organisation, such as processes, applications, data, and technology. Services could be used to represent a combination of these EA elements on different levels. Thus, he suggested enterprise architects should understand how EA elements are structured into services and track these changes:

> *Your enterprise architecture is just managing a portfolio of different things. One of those things will be services as well as business processes, as well as rules as well as applications as well as a bunch of physical technology as well as a bunch of structured information. So a service can be a combination of any one of those components couldn't it? So all we really need to know in an enterprise*

> *architecture sense is which of those are structured in a service and what could they deliver, and how long do they live for.*

Participant [I-3] offered an example of SOA's integration into the business architecture. His organisation adopted ArchiMate. In this case, business services were mapped to lower level application services, and infrastructure services were not yet included in their use of EA. The reason for such decomposition is to align business services components to support application services in the tax domain. Enterprise architect [I-3] reported that services are identified in both business and IT: *"Decomposition on a business level follows decomposition on a technical level, aiming to have autonomous business services supported by autonomous application services".*

In the service catalogue, business services were documented. Each business service was described using a standardised format defining what is offered, what input is required, and what IT support is needed. Business services were divided into core and additional. Examples of core services were "receive", "deliver", and "archive". Additional services were further divided into: generic and specific. Generic services were the ones accessed by more than one tax domain, while specific services were used and owned by a specific tax domain. For example, "validate additionally" was a specific service for a specific domain.

Enterprise architect [I-4] reported another example of SOA's integration into business architecture. His organisation's approach to SOA is described below: *"The recent approaches towards a focus on services are more closely aligned to business functionality rather than technical functionality and provide a coarse grain of capability delivery".*

There was a clear distinction between business services and technical services in the modelling language. Enterprise architect [I-4] mentioned that:

> *we're probably among the first who started distinguishing between business service and technical service explicitly in the language...In the business view we've got things such as business service, policies, and the business collaboration which we call the community.*

The technology layer has artefacts such as a service, service description, service interface, and composition entities.

In summary, SOA's integration into business architecture is often associated with architectural transformation on other levels of EA, such as IS and technology architectures. Seven participants explicitly reported that SOA's introduction changed their organisation's business architectures. These changes included the addition of new architectural elements such as business services, their description, and their relationships with other architectural elements.

5.6.2 Information Systems Architecture

The second level of the architectural elaboration is the information systems (IS) architecture. SOA is integrated into IS architecture (the IS architecture is transformed) or not integrated (IS architecture is reproduced). All the cases mentioned above that integrated SOA into the business architecture also reported SOA integration at this level. Additionally, three participants specifically reported SOA's integration into IS architecture without changes to the business architecture. For example, EA manager [I-11] expressed the thought that SOA fits into the IS and technology architectures of EA. The application architecture includes services and services component as one of the main building blocks: *"Well it [SOA] fits within enterprise architecture; I guess the IBM view is the thing that guides us … The applications level has services and components".*

EA consultant [I-9] also argued that SOA is just part of IT architecture:

> *SOA again is just one part of the technology implementation… logical implementation would be service based thinking and then the physical implementation within the IT architecture was taken down to a particular model of service layers.*

Further, chief enterprise architect [I-1] presented another example of SOA's integration into IS architecture. In his organisation's approach, SOA was a concern for application and integration architectures. The IS architecture had SOA elements, such as enterprise service, component service, service descriptions, and SLAs. The technology architecture accommodates the enterprise service bus element, service monitoring, web services, and SOA-related security gateways.

Enterprise services and component services are presented in the integration architecture. The enterprise service is composed of lower-order services, which are component services. Other artefacts such as service descriptions and SLAs are represented too. In the integration architecture, core elements are identified and defined to answer questions such as: *"what is an enterprise service?" and "what constitutes such a service?"*, [I-1]. At the conceptual level, specific services are designed for specific projects. Detailed design is undertaken, such as data flow and data signature. In the logical design, detailed functional descriptions, SLAs, and transactional rates are defined. At the physical level, the actual service is built using relevant standards:

> *So royally speaking SOA is fundamentally an integration pattern and most of the artefacts that exist inside SOA are in the integration architecture but there are at different levels.... Integration architecture is where we have these artefacts and everything you described right down to the service descriptions, the SLAs and the detail, all exist within it, but they don't exist at the strategic level*. [I-1]

5.6.3 Technology Architecture

The third level of EA elaboration is technology architecture. SOA is or is not integrated into the technology architecture. All the cases mentioned in the previous two levels of architectural elaboration integrated SOA into their technology architecture except for one case. Additionally, two practitioners reported that their organisation only integrated SOA into their technology architecture. For example, enterprise architect [I-14] mentioned that, in his organisation, SOA was being used as an integration approach between backend and frontend applications:

> *We have multi-layered architecture and we have our factory applications in the back end layer. We have middleware layer and we have the front end layer and we have SOA concept in the middleware layer.*

SOA's integration into the technology architecture is further reported by another case. Enterprise architect [I-13] explained that his organisation's SOA implementation was based on point-to-point integration using web services, and that it later used an enterprise service bus:

> *We've been doing these projects now on and off over five years. We've been thinking about SOA and ways to make things happen. Those thoughts have started in the very early stages from a concept of point-to-point integration using web services, okay? Over time that thought changed…. We quickly realised that enterprise service bus was the way to go.*

He [I-13] elaborated that SOA in his organisation was technology driven when it should have been business driven. EA consultant [I-20] describes such a situation well. He suggested that driving SOA only from the technology side of the organisation often misses its business objectives. He argued that enterprise architects should be involved in driving SOA and align it with business activities:

> *People look at the service oriented architecture and then straight away they think of the technology adaptors, enterprise service buses and things like that and they forget about the business reason. So the architecture's role and the enterprise architect's role is to make sure that that business reason never gets lost otherwise that technology project will just be running off in all different directions and there's no alignment* [I-20].

This level of architectural elaboration could result from an early adoption of SOA or a low maturity of SOA adoption (considering also the architectural conditioning phase's possible influence). For example, both [I-13] and [I-20] reported that their organisation had low SOA maturity and SOA was driven by IT.

5.6.4 Enterprise Architecture Governance

The fourth level of architectural elaboration is EA governance. EA governance is transformed or reproduced due to SOA's introduction. Transformation occurs when SOA governance and EA governance practices are integrated.

The findings show that the participants had different perspectives on SOA governance in relation to EA governance. Six participants explicitly argued that SOA governance has to be integrated into EA governance. For example, enterprise consultant [I-18] affirmed that SOA is delivered as part of EA, and thus EA governance covers SOA governance and practices: "[EA] governance should stem into anything that is delivered as part of the enterprise architecture, which includes SOA".

Participant [I-12] further stated that SOA is essentially an architectural style of EA and, thus, that EA governance is capable of handling SOA-related governance issues: *"SOA is just an architectural pattern. EA governance is the thing that is going to make this work so SOA is part of your overall enterprise architecture governance".*

Chief enterprise architect [I-2] presented an example of SOA governance integration into EA governance. She noted that, in her organisation, SOA governance was established and integrated with the overall EA governance:

> *So one of the first things for us was to establish our SOA governance and to ensure that anything that we were doing in terms of SOA governance still complied with what EA governance required them to do. We have to comply with what the over arching enterprise architecture principles and governance policies were.*

However, EA consultant [I-20] argued that SOA is a large initiative and requires its own governance practices that are aligned with EA governance practices:

> *You need governance around SOA but the thing is, is that...so EA in and of itself has its own governance process but the thing is, is that SOA is a large enough chunk of the piece to have its own governance process but it needs to be linked in with the architecture governance.*

Some participants also argued that EA governance should not be integrated into SOA management or implementation decisions, particularly on the technical levels. For example, [1-16] argued that:

> *There's a lot of implantation decisions and management decisions that enterprise architecture does not need to be concerned with.... and especially when it gets to the lower layers of the SOA.*

In summary, there were mixed opinions on the integration or non-integration of SOA governance into EA governance. The level of the integration was also problematic in terms of defining the boundaries of both SOA and EA governance practices and their level of integration.

5.6.5 Enterprise Architecture Methods and Tools

This level of EA elaboration emerged from the data analysis. It represents an architectural transformation or reproduction of EA methods and tools resulting from SOA's introduction. This level of the architectural elaboration is the weakest among the five levels. The integration on this level was reported explicitly by five participants with different opinions. According to some participants, introducing SOA requires changes to EA development methods and tools. For example, chief enterprise architect [I-1] claimed that such changes need to be considered at a very early stage of SOA adoption in order to identify and build the right services: *"Not laying foundations first such as changes to SDLC, governance processes… [leads to] building the wrong services"*.

In a similar argument, chief enterprise architect [I-2] reported that, upon SOA adoption in her organisation, the guidelines, methods, and processes that are required to implement service-oriented projects were established:

> *we have methods, we have tools, we have guidelines, we have glossaries, we have each one of those artefacts which helps our development and project management to deliver service oriented projects.*

EA consultant [I-7] also believed that SOA changes the work of IT architecture and development:

> *We have a methodology where the traditional lifecycle model in IT is requirement design, development, testing and maintenance and support… So what we believe is that after the requirements, the business requirements, we need to have a service identification phase.*

On the other hand, enterprise architect [I-13] argued that SOA does not change the design and project delivery activities:

> *Fundamentally we're trying to say that SOA is not a new thing, we're not changing the way that we're doing business so we still do our standard business analysis up front, we still do our same solution architecture work.*

Chief enterprise architect [I-2] argued that her organisation's service lifecycle differed from other traditional software development. She reported some changes in existing development procedures and practices in order to support service-oriented development:

> *When we introduced SOA based software development approach we actually changed the developer's handbook and we changed the project manager's handbook because there're certain points in the project life cycle and the software development life cycle that they need to think about service orientation.*

Further, EA consultant [I-16] distinguished between two sides of services management that are shared by EA and SOA. EA should be focused on the business side of the organisation including activities such as services identification and services requirements identification. Then, the developers at the technology level manage the implementation of these services:

> *the architects focus on the business and identifying which services are required and what the requirements are for those services … The developers then takes those services as sculpted by the enterprise architecture and chooses their implementation for those services.*

Similarly, EA consultant [I-9] agreed that business architecture development should be guided by services thinking to drive the actual services implementations on the IT architecture.

> *that was done was that the enterprise architecture had a statement that the logical implementation of the business architecture would be done through services. So it was taken as a guiding principle in the enterprise architecture that effectively, logical implementation would be service based thinking and then the physical implementation was then within the IT architecture was taken down to a particular model of service layers.*

In summary, a small number of participants suggested that SOA's introduction brings changes to EA development methods and tools. It seems that an EA with a strong solutions development focus would require changes similar to those reported in cases [I-1 and I-2] and less-obvious changes when EA focuses on an organisation's strategic and business sides.

5.7 Chapter Summary

The a-priori model was extended according to the presented interview findings. Figure 5.7 depicts the updated a-priori model.

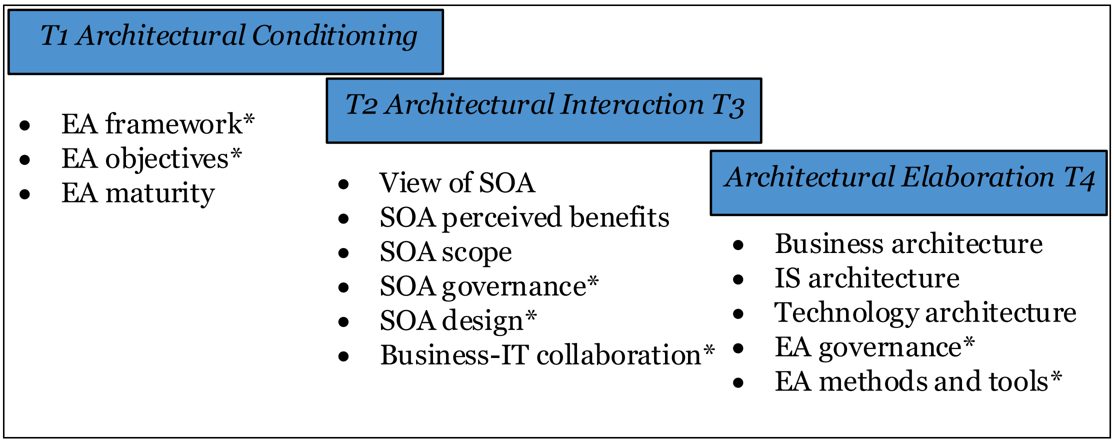


Figure 5.7 The extended a-priori model of this thesis

The interview analysis extended the a-priori model. First, the architectural conditioning phase was further enriched from the interview findings. The findings indicated the conditional influence of two other conditional generative mechanisms (EA framework and EA objectives) on EA evolution (here referred to as SOA's integration into EA). They also confirm the relevance of EA maturity as a conditional generative mechanism in the organisations studied.

The findings show the diversity that exists in organisations' adopted architectural frameworks. Some participants reported EA frameworks that were shaped by previous actions. Some also reported the adoption of partial or modified EA frameworks. These frameworks become conditional generative mechanisms for next EA-related activities by either enabling or constraining them. The findings indicate that EA was adopted for different objectives (classified into strategic, operational, IT and governance), which guided the architectural practices. The findings also confirm the relevance of

EA maturity for SOA's integration into EA. Some participants argued that mature EA practices increase the sustainability of EA and enable SOA's integration into EA. Mature EA practices keep EA documentation up-to-date, plan EA changes, enforce governance around EA practices, evaluate and maintain EA models, and involve business in their activities. Thus, these three conditional generative mechanisms are hypothesised to have a conditional influence on EA evolution. In other words, they enable or constrain EA evolution.

Second, six action-formation mechanisms (view of SOA, SOA perceived benefits, SOA scope, SOA governance, SOA design and business and IT collaboration) were found to influence SOA introduction. SOA is introduced by a certain perspective, entertaining certain benefits, through a determined scope, designed in many ways, governed differently, and with a different level of business/IT collaboration. Thus, it seems that SOA introduction will be different in different contexts due to the influence of the six action-formation generative mechanisms.

Third, SOA's introduction under a certain architectural conditioning influence results in different architectural elaboration outcomes. The architectural elaboration outcomes are classified into five levels, two of which emerged from the data (business architecture, information systems architecture, technology architecture, EA governance, and EA methods and tools).

The explorative interview phase explored SOA's integration into EA in depth in order to extend and enrich understanding of the three building blocks of the a-priori model: the architectural conditioning, SOA introduction, and architectural elaboration. The purpose of the next phase, the case studies phase, is to examine the theoretical model, developed in the previous phases, two different contexts to further understand EA evolution.

Chapter 6: Dubai Customs Case Study Findings

6.1 Introduction

This chapter introduces the findings of the first case study, which was conducted with a government agency (Dubai Customs) in the United Arab Emirates. Dubai Customs is one of the oldest Dubai Government agencies. As Dubai has grown, Dubai Customs has expanded its operations to manage such growth. In response to this growth, its customs role has expanded. It has become an organisation that facilitates trade and protects the borders of the country. Dubai Customs started an enterprise architecture (EA) program in 2006 to facilitate its strategy, business, and technology alignment. Its EA program was part of a large transformation initiative. EA was used to support fast decision-making and to support the dynamic business needs of the organisation. Moreover, Dubai Customs was one of the first leading public agencies in the United Arab Emirates to undergo an e-government transformation and to progress toward the wide electronic delivery of services. In particular, Dubai Customs introduced SOA (in the form of a service-oriented electronic clearance system) in 2008 to support the delivery of services.

This case study was conducted to satisfy the contextualisation stage of the critical realist methodological framework (see Chapter 3). In particular, the theoretical model, developed in the previous phases, was examined in this case study to explore EA evolution in this specific context. The proposed structures and generative mechanisms, in the developed theoretical model, were further explored to describe EA evolution and explain the observed EA evolution outcomes in Dubai Customs.

As Chapter 3 presents, the case study design is *a retrospective one*. The three morphogenetic phases (conditioning, interaction, and elaboration) are used to understand how the EA evolution outcomes were generated. The phase of architectural elaboration (evolution outcomes) was identified as the particular point in time that this study would illuminate, and then move backwards through the previous two phases of the model, seeking to uncover the generative mechanisms of the architectural conditioning and interaction phases that have interacted to generate the observed outcomes. In this case, SOA's integration into EA was completed prior to the researcher's engagement with the case, and the EA evolution outcomes were known (based on online empirical evidence) prior to the conduct of this case. The case was selected because Dubai Customs has a well-established EA program, has received EA-related awards from ICMG and IBM, implemented SOA, and SOA was integrated into EA (based on online evidence prior to the conduct of the study).

The chapter progresses as follows. Section 6.2 describes the data collection and analysis. Section 6.3 describes the background of the organisation, and Section 6.4 shows its organisational structure. Section 6.5 examines the architectural conditioning of the case: it looks at EA framework, EA objectives, and EA maturity prior to SOA's introduction. Section 6.6 examines SOA's introduction in 2008. Section 6.7 discusses the architectural elaboration on the five architectural levels of the model. Finally, section 6.8 summarises the findings of this case.

6.2 Data Collection and Analysis

During June 2012, eight interviews were conducted with eight senior executives at Dubai Customs in the United Arab Emirates. Table 6.1 presents the participants' information. Interviews were conducted following a case study protocol (Appendix B). Each interview lasted between 40 and 90 minutes, and was recorded, transcribed and analysed using NVivo 9, following the thematic analysis technique (see Chapter 3).

Table 6.1 Participants' information

Participant	Position	Years in org.	Years of experience	Background
P-1	Head of IT planning and enterprise architecture	5	7	Business and management
P-2	IT strategist	6	15	Strategy/planning
P-3	IT strategist	4	10	IT/strategy
P-4	Senior business	6	20	Business

	Architect			
P-5	Senior business architect	6	25	Business
P-6	Senior business architect	5	5	Business
P-7	Senior tech architect	5	10	Technology
P-8	Senior tech architect	3	10	Technology

In order to achieve triangulation, besides the conducted interviews, documents related to both EA and SOA were obtained from the department (internal) or online from the Dubai Customs website or other websites (see Table 6.2). These documents are cited as D-1, D-2, and so on throughout this chapter.

Table 6.2 Extra collected evidence (documents)

ID	Source	Description
D-1	Online	A presentation given at Telelogic User Conference to present the implementation of EA and its benefits (2007).
D-2	Online	A presentation given at Information Technology Governance Assurance Forum (2007). It presented the use of EA as a governance practice.
D-3	Internal	A suitability report by Dubai Customs represents its efforts and practices to achieve sustainability. It discusses Dubai customs strategy and achievements at social, economic, environmental and workplace levels as results of business practices during 2011.
D-4	Online	Report of COBIT implementation at Dubai Government and their efforts to assess governance levels at different agencies such as Dubai Customs.
D-5	Online	A white paper from IBM of the benefits of using EA to align business and IT. Dubai Customs is presented as one of the examples that achieved benefits from its EA implementation.
D-6	Internal	This document from Dubai Government presents an e-services delivery excellence model for the electronic provision and improvement of government services. It acts as guiding principles for services enablement evaluation.
D-7	Internal	A presentation presents the use of EA for knowledge management and knowledge sharing at Dubai Customs.
D-8	Internal	An internal document for services identification and classification.
D-9	Internal	Classification of Dubai Customs domains.
D-10	Internal	Dubai Customs EA's meta-model.

D-11	Internal	A document with information related to Dubai Customs (an overview of SOA program)
D-12	Internal	Another document with information related to the SOA (guide of SOA program).
D-13	Internal	A presentation by the head of EA about their EA implementation and benefits.
D-14	Internal	A report from IBM describes the implementations of EA at Dubai Customs.
D-15	Online	A report about the launch of Dubai Customs' electronic system.
D-16	Internal	Information about some technical aspects of SOA implementation.

In this case, the morphogenetic cycle of SOA's integration into EA was determined by using the stability-change-stability approach discussed in Chapter 3. The change is limited to SOA introduction as a trigger of EA evolution. There could be other aspects that cause changes to EA, but they are outside the scope of this study, which focuses only on SOA and EA integration. Dubai Customs' SOA integration into EA morphogenetic cycle is shown in Figure 6.1.

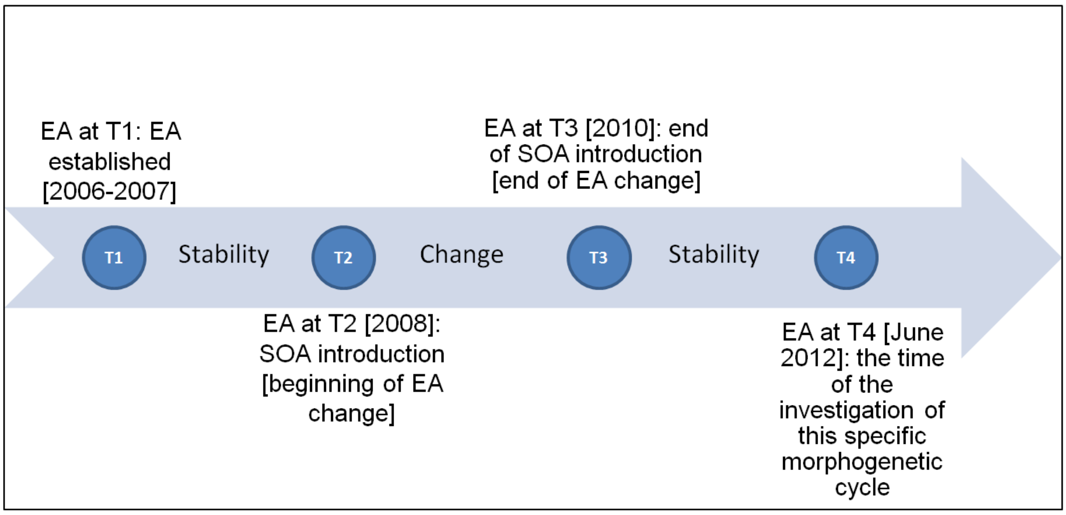


Figure 6.1 The morphogenetic cycle of SOA's integration into EA at Dubai Customs

EA was established in 2006 and completed by the end of 2007, which could be considered a morphogenetic cycle by itself. However, this study focuses only on SOA's integration into EA, where that period of change (EA establishment) was completed *prior* to SOA's introduction. The results of that period are considered the architectural conditioning (T1-T2) of the new morphogenetic cycle (SOA's integration into EA). The change period (architectural interaction) began when SOA was introduced in 2008 and finished in 2010. This study was conducted two years later, in June 2012, after SOA's introduction.

In order to understand SOA's integration into EA outcomes, the event (SOA's introduction) and the status of EA prior to the interaction were studied retrospectively. The retrospective analysis was achieved through intensive interviews with executives involved in EA and SOA, and was supported by the analysis of obtained relevant documentations (see Figure 6.2). In an effort to address the possible limitations of exploring time-consuming phenomena through retrospective interviews, multiple participants with different backgrounds and hierarchical levels were interviewed, and internal and online documents were examined to provide multiple triangularly perspectives.

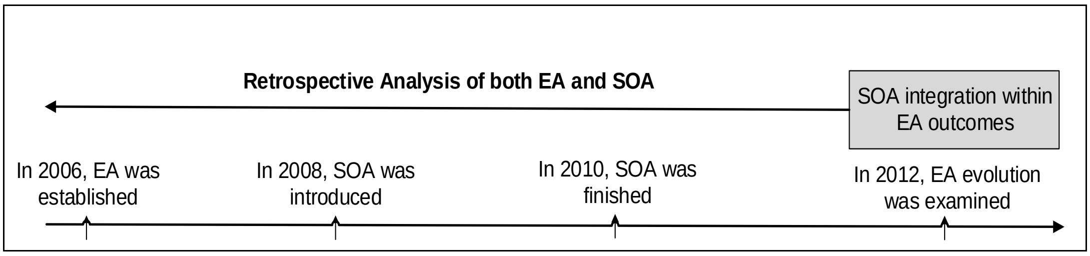


Figure 6.2 Employed retrospective analysis

Analysis of the collected data was informed by the analytical procedures used for the explorative interviews in Chapter 5. The interviews were transcribed, and all the interviews and obtained documents were imported to NVivo to prepare them for analysis using the final codebook of the explorative interview phase. The analysis used the thematic analysis technique (and, more specifically, a deductive approach using only this thesis's theoretical model as a lens). Figure 6.3 shows an extract of the codebook used and the analysed text. Most of the codes were repeated from the analysis of the previous phase. A few additional codes specific to the case study, such as its specific layers of EA and its operating model, were generated as part of this coding process.

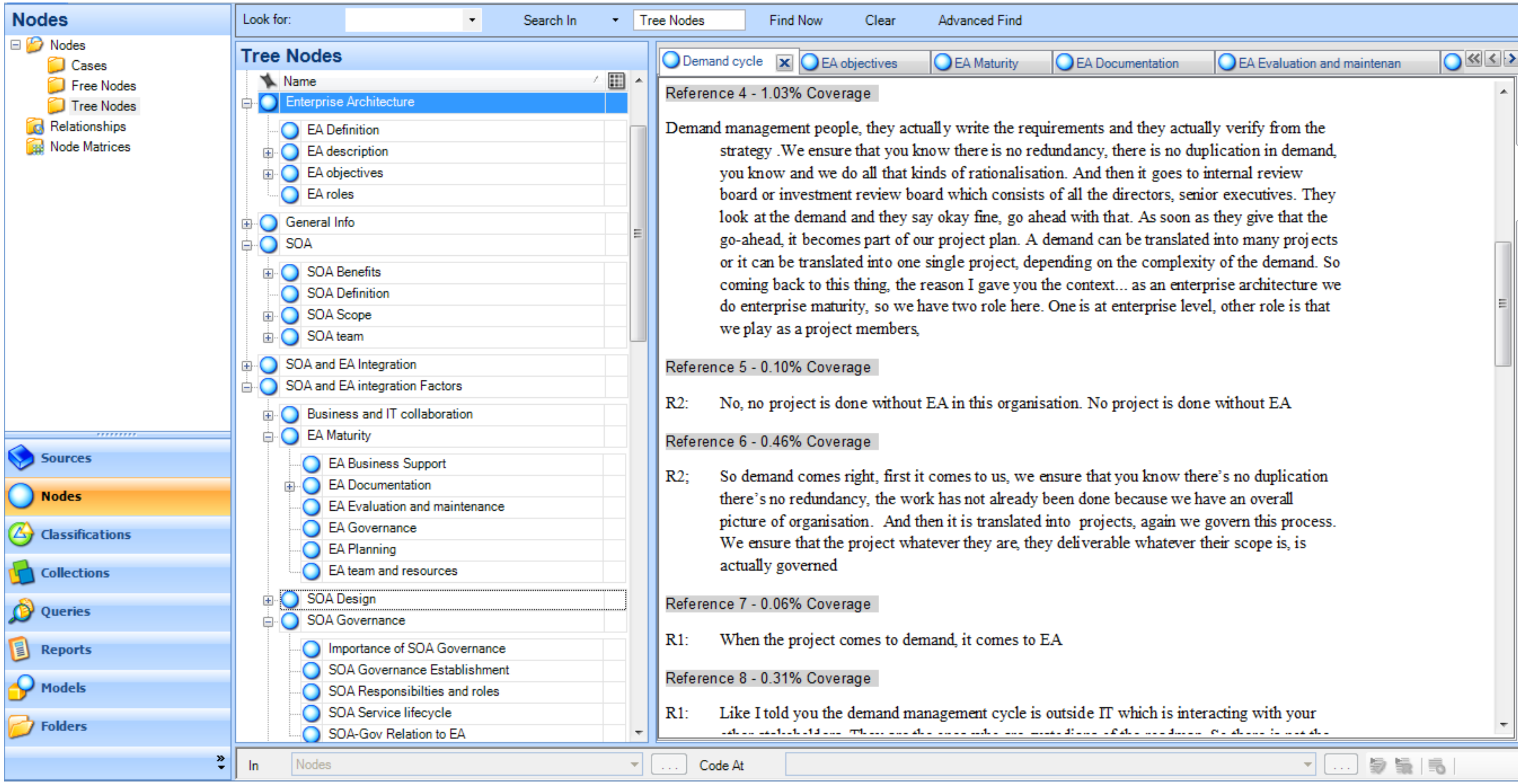


Figure 6.3 An extract of the case analysis using Nvivo

6.3 **Case Background**

Dubai Customs is one of Dubai's oldest government agencies. It was founded in the early 19th century to secure the integrity of Dubai's borders (Dubai Customs, 2011). As an integral agency of the Dubai Government, Dubai Customs continues to fulfil its objectives as an icon of Dubai's power and guardian of Dubai's trading interests [D-11]. Dubai Customs regulates the flow of trade, regulates the import and export industry, and generates trade statistics reports (Dubai Customs, 2011).

As part of their long term business strategy, Dubai Customs has a vision to be "the leading customs administration in the world supporting legitimate trade" and its mission is to "protect society and sustain economic development through compliance and facilitation". Dubai Customs has outlined the following strategic goals: (1) contribute to the economic and social development of Dubai, (2) adopt and share best practice in terms of business processes and systems, (3) provide the best human and technical resources, and (4) improve customer satisfaction and loyalty (Dubai Customs, 2011, 2012c).

Corporate governance is an important aspect of the customs business. Dubai Customs is committed to expressing and maintaining transparency and responsiveness to its stakeholders. Dubai Customs is managed by Executive Directors and led by a General Director to make sure strategies and goals are achieved and that values and principles are adhered to (Dubai Customs, 2011, 2012c). Dubai Customs' governance manual outline the frameworks, policies, laws, methodologies, practices, and procedures to control the way the customs is managed to achieve its strategic objectives. The manual aligns practices with international, regional, and local laws and standards. The manual is reviewed frequently. The roles and responsibilities are first and foremost governed by the laws of Dubai Government and the Executive Council to supervise Dubai Customs' overall strategic direction, set the general policies and plans, and oversee its implementation.

Since it was established, Dubai Customs has grown through many phases. During its infancy period, Dubai Customs adopted an institutional approach. At that time, the ruler of Dubai's personal office was situated in the old customs building, which emphasised Dubai Customs' significant role and its position in Dubai Government (Dubai Customs, 2011). During that phase, Dubai Customs had the traditional roles of collecting duties and inspecting freights and passengers. In the last decade, Dubai has experienced substantial growth in trade and in industrial and urban development. Dubai Customs has been challenged by such growth (Dubai Customs, 2011). Nevertheless, it is determined to ensure the preservation of its historical profile by relying on an extensive record of successful achievements since it was founded (Butti, n.d.). With a rich history and vast experience to guide the way, Dubai Customs is always looking ahead. Dubai Customs acknowledges the need to evaluate its current environment regularly and make the necessary improvements to increase its contribution to the future of Dubai, the Emirates, and the Gulf countries [D-11].

In response to this growth, Dubai Custom's role has been expanded. It has become an organisation that facilitates trade and protects Dubai's borders. The increase of Dubai Customs' workload required additional staff and activities and the modernisation of overall services and operations. Dubai Customs has adopted electronic, enhanced, and easy processes and procedures to facilitate smooth transition of cargos in and out of the country (Dubai Customs, 2011). The most outstanding of these programs was the introduction of a service-oriented electronic clearance system. It supports the delivery of business-to-government (B2G), services which are intended to encourage couriers with enormous volumes of transactions to lodge and process customs clearance and other services electronically (Butti, n.d.).

Dubai Customs has become a leader in delivering innovative services supporting national objectives and improving customer experiences. Contemporary systems, easy practices, speedy clearances, and cooperative and chivalrous service are and will be maintained as keystones of Dubai Customs' practices [D-11]. For a long time, the customs agency has contributed to Dubai's development, creating a far-reaching sphere of operations, and contributed to Dubai's economic power. It has reinforced

Dubai's role as an international trade hub and an influential trade point to the Gulf countries [D-11].

6.4 Organisation Structure

Dubai Customs has approximately 3,000 employees. They are functionally organised into five divisions and multiple departments (Dubai Customs, 2011). These divisions are (1) human resources (HR), finance, and administration, (2) customer management, (3) policy and legislation, (4) customs cargo operations, and (5) customs development (see Figure 6.4).

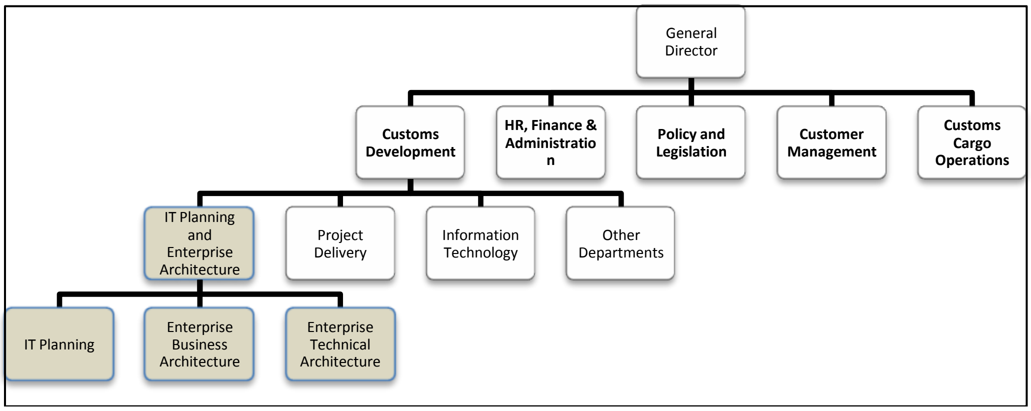


Figure 6.4 Organisation structure

6.4.1 HR, Finance, and Administration

This division has three departments: HR, finance and administration. The HR department is in charge of developing and implementing the best methods and practices necessary to build leadership and functional competencies. It is also responsible for developing the capacities and capabilities of staff across the department. It encouraged performance-based culture to offer top-quality services and operations.

The finance department is in charge of planning and managing the organisation's financial issues. It incorporates financial targets, planning, management, and control. It also formulates the organisation's annual budget. The administration department provides employees with an effective work environment. It manages and executes Dubai Customs' purchases.

6.4.2 Customer Management

The customer management division is in charge of service delivery improvements. It is dedicated to providing the best customer service delivery standards. Some customs facilitation programs have been adopted, such as the clients' accreditation program and the customs' warehousing system. Additionally, the customer service charter and customer guide were implemented to enable clients to find all the services they expect when engaging with the organisation.

Furthermore, the customer complaint system has been implemented to support customers' feedback and manage their complaints. Dubai Customs is the first government department of its kind in the Middle East and North Africa to be awarded the ISO (10002) certification on customer complaints handling. It was the first organisation globally to receive the ISO 10001:2007 conformity certificate from Lloyd's Register. A toll free call centre has been established to improve customer communication experiences.

6.4.3 Policy and Legislation

The policy and legislation division is in charge of the organisation's legal affairs and consultancy tasks. It sets up policies and procedures and manages international relations between customs, other countries, and international and regional organisations. It manages the implementation of customs valuation agreements, and tariff and payable duty rates on imported goods. It deals with prohibition and restriction decisions, economic agreements, and rules of origin. It is also responsible for protecting intellectual property rights and trademarks. It deals with suspended customs duty cases such as temporary admission, free zones, transit, and the re-exportation of goods (Dubai Customs, 2012a).

6.4.4 Customs Cargo Operations

The customs cargo division is responsible for implementing the common customs law of the Gulf countries and other relevant laws. The division supervises the electronic clearance transactions and conducts inspections using intelligence rules applied to goods. It performs post-audits on all customs transactions, employing a mechanism chosen by customs as a fundamental criterion for post-audit operations. Additionally, the division is in charge of investigating criteria development that relate to customs cases and provisioning related evidence. It carries out the initial detaining of goods and manages the storage of suspect goods, and provides devices and equipment necessary for examining and detecting suspect goods (Dubai Customs, 2012a).

6.4.5 Customs Development

Customs development is a dynamic division at Dubai Customs. IT planning and enterprise architecture is one of the customs development division's departments. The division ensures the constant modernisation and development of customs procedures, operations, projects, and services to keep pace with the most recent international practices and techniques. Recently, the division has driven a transformation initiative to transform Dubai Customs services into electronic ones. Currently, 100 percent of its services are accessible via the Internet, including electronic clearance services provided by SOA's implementation. Additionally, the division disseminates information related to its projects, services, and procedures to the public and customers through the customs website and different media techniques to ensure continuous communication with Dubai Customs customers and partners (Dubai Customs, 2012a, 2012b). Participant [P-2] described the division's role as being in charge of the organisation's development and changes:

> *Any changes, any development within the organisation is actually done within the customs development division (CDD). So, we deal with people, processes, technology and information. So the role of customs development is to transform Dubai Customs.*

The following sections use the developed theoretical model (see Figure 6.5) to examine SOA's integration into EA at Dubai Customs. The findings are organised along the theoretical model based on Archer's (1995) morphogenetic theory. Prior to describing the outcomes (the architectural elaboration), the architectural conditioning (T1-T2) and architectural interaction (T2-T3) are presented to comprehend SOA's integration into EA.

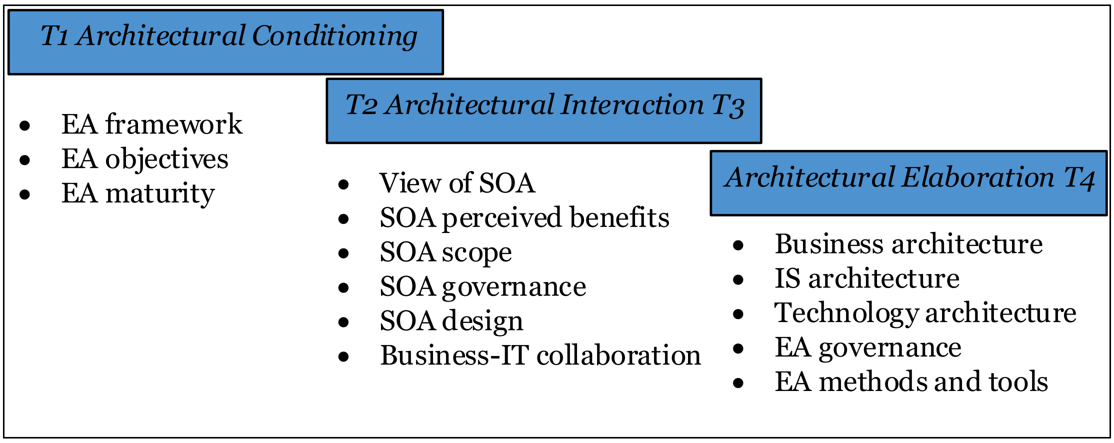


Figure 6.5 This thesis's theoretical model

6.5 **Architectural Conditioning**

This section addresses the conditions of EA prior to the architectural interaction (SOA's introduction). It briefly presents an overview of EA implementation and then organises the findings according to the following three conditional generative mechanisms: EA framework, EA objectives, and EA maturity.

Dubai Customs has adopted EA in order to align its strategy, business, and technology. The EA program was launched in 2006. Senior Business Architect [P-4] reported that the EA program was part of a large transformation initiative. It was adopted to make sure that the transformation objectives were realised. He stated: *"When the transformation initiatives started... the organisation wanted a mature practice to be followed to ensure that the realisation of the objectives happens"*.

The program was implemented in three phases (see Figure 6.6). The EA program was launched in late 2006 and the first two phases were completed by June 2007. By November 2007, most EA artefacts such as strategies, business processes, activities, technical artefacts, and their relationships were documented and stored into the system architect [D-1]. A year later, by the end of phase three, the EA program was successfully implemented across the organisation [D-1].

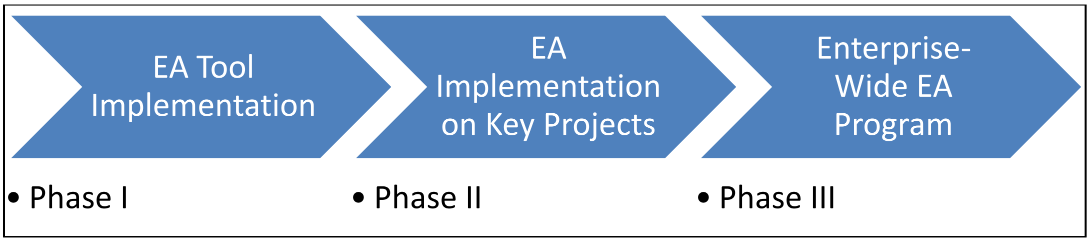


Figure 6.6 EA program implementation phases [D-1]

EA at Dubai Customs "articulates and connects organisational information to support fast decision making and to have a single point of truth containing information to support the dynamic business needs" [D-13]. The following quote from an EA Manager's presentation offers a clear definition of what EA is: *"Enterprise architecture is the broker between Business and IT. It provides the benefit of knowing why we need to build, what to build, when to build it, and how to build it"* [D-13].

When the EA program started in 2006, efforts focused on the processes and technical levels of the organisation. The organisation identified the value chain, business groups, business processes, and functions at the business level. Then, the business architecture was mapped to the technical architecture by identifying how business processes are realised at the technology level to answer questions such as what applications support what processes, and what applications run on what infrastructure. Participant [P-5] stated: *"When EA started, it started because of the transformation, business transformation initiatives; we had analysis on core areas. We did not define them as services at that time"*. Participant [P-4] stated: *"We did something called value chain and business groups and business functions. When you define it that way, you look at yourself only internally and not from the customer perspective at all*".

Based on the participants' statements, the organisation's EA approach, which was based on business processes, was an internal perspective of the organisation itself. This approach was called "inside-out" by an IT strategist [P-2]. He declared:

> *We were thinking inside-out. We were thinking whatever we were doing is something which our customers needed. We were thinking from the provider's perspective this is what I do, I can enable things and I can throw it on the web and I can ask the customers to use it.*

6.5.1 EA Framework

Dubai Customs built their own EA framework following customised Zachman and TOGAF principles. The organisation needed a way to help decision-makers take the right decisions by providing all the information needed about the organisation from different angles. The organisation divided EA into multiple layers: strategy, resources, process, information, and technology. The first layer, strategy, encompasses the organisation's business vision, objectives, enablers, and performance measures. This layer holds strategy-related elements such as directions, guidance, objectives, the means of delivering these objectives, and performance KPIs [P-3]. The second layer, resources, holds elements such as people, assets, organisation, and locations. The third layer, process, holds business processes, business process definitions, and metrics. The information layer includes information models and information flow. The technology layer includes applications, data models, technical reference models, hardware, and network.

According to a senior business architect [P-5], the whole organisation was decomposed to understand its current (as-is) state. It became a reference point for identifying the gaps and the changes that might be required to move to a future state. These as-is business architecture models were used to discuss project proposals and demands.

EA documentation outcomes were stored in a repository using IBM System Architect tool. A report from IBM [D-5] describes Dubai Customs' adoption of IBM Systems Architect. The report notes that Dubai Customs needed a platform that would enable effective business and technical planning to drive the organisation forward in the coming ten years. IBM System Architect was chosen because it met most of Dubai Customs' requirements such as flexibility and ease of use. The customs agency has used the extensibility of IBM System Architect to extend it in areas specific to its needs. IBM System Architect also supported Dubai Customs' EA methodology. It has enabled Customs to do business faster with notable agility [D-5].

To improve access to the information stored in IBM System Architect, the organisation internally developed the enterprise connected view (ECV), an interactive interface used for navigating and querying stored information (see Figure 6.7). The executive director of customs development division stated: “We use System Architect as the basis for our enterprise connected view (ECV), which enables us to manage enterprise information, run impact analysis, and make decisions more effectively” [D-5].

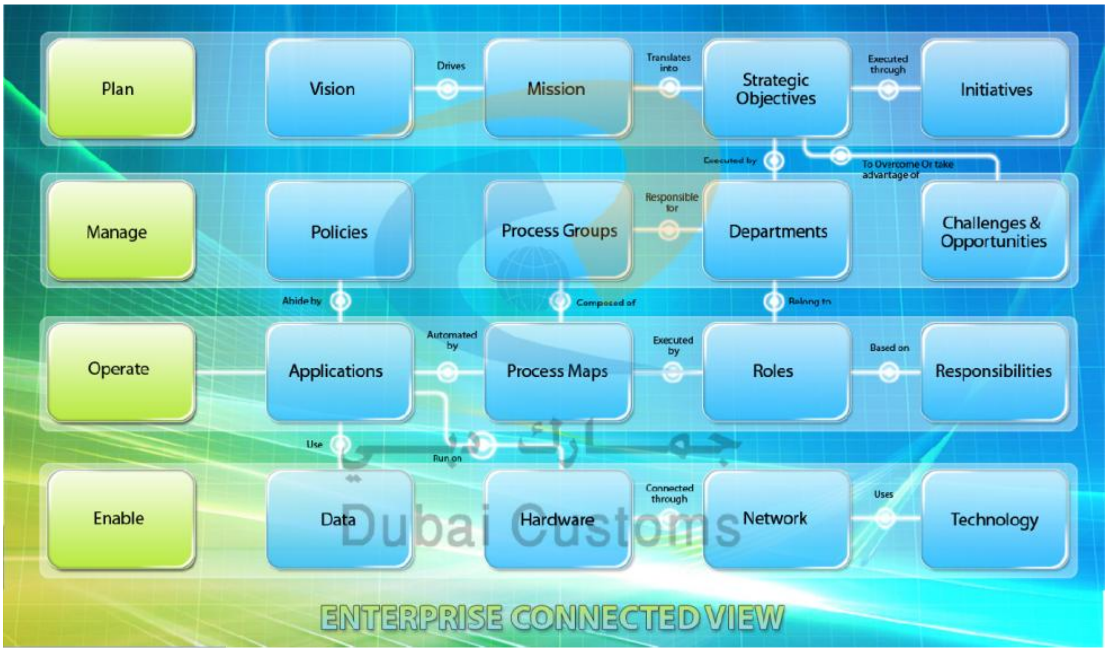


Figure 6.7 The enterprise connected view [D-1]

6.5.2 EA Objectives

Since EA’s early adoption at Dubai Customs, it has been based on a strategic long-term vision. EA has strategic, operational, IT, and governance oriented objectives (see Table 6.3). EA holds corporate strategies for the corporation, departments, and divisions. It holds business and technical information and stores them in one repository. Dubai Customs uses EA to align its strategy with that of the Dubai Government. It enables effective governance of both business and technical architecture and facilitates fast responses to changes in business and IT requirements [D-2].

According to the Head of IT planning and EA [P-1], EA and ECV focus fundamentally on documenting all informational assets in the organisation in order to manage and govern the organisation on multiple levels: strategy, business, information, and technology. EA and ECV were introduced to generate blueprints of the organisation and to be the single source of official information. They were used for planning, governance, decision-making, and impact-analysis purposes. Participant [P-4] commented on the reasons for establishing EA at Dubai Customs. He mentioned that EA was adopted to improve the decision-making process, prioritise work, develop new capabilities, improve technology implementation, and guide the future of the organisation. The organisation has used EA to achieve benefits in areas such as business and IT alignment, impact analysis, and strategic decision-making assistance [D-7].

Table 6.3 EA objectives at Dubai Customs

EA objectives	
Strategic	Business and IT alignment, strategic decision-making assistance, change management, knowledge management and identify gaps
Governance	Holds strategies, holds business and technical requirements, and effective governance of both business and IT architectures
Operational	Documentation of all enterprise components, reuse of components, impact analysis, discovery of duplications and standardisation
IT	Provide solutions requirements, monitor their development, reduce IT duplications and IT complexity

6.5.3 EA Maturity

As mentioned in Section 6.5, Dubai Customers adopted an organisation-wide EA program in late 2006. By the end of 2007, most of the architectural artefacts were captured and stored in the EA repository. The EA maturity assessment was measured based on: (1) the obtained documentations, which describe the early stages of EA prior to SOA's introduction, (2) participants' responses, and (3) the EA maturity assessment survey questionnaire (shown in Appendix B in the case study protocol), which was also handed to the participants. Four completed forms were received. The combined findings of the documentation, surveys, and interviews indicated that EA was mature (between level 3: well-defined program and level 4: managed program out of 5) before SOA's introduction (see Table 6.4). The details of each maturity dimension are presented in the following subsections.

Table 6.4 EA maturity prior to SOA's introduction

Level 3: well-defined program
• Templates are used to ensure the capturing of information is

consistent • Documentation of business and IT information is consistent • EA plans are well-defined, including a structured framework and timeline for developing the EA • EA activities are carried out according to the defined plan • Architecture Governance committees are defined, and have defined roles and responsibilities • Authority of the governance committees is aligned to work together smoothly • EA team includes business and IT staff • Training is provided for members of the EA team • Business and IT stakeholders have a good understanding of the architecture principals and participate in EA processes • There exist defined evaluation processes for EA framework and outcomes
Level 4: managed program
• Documentation has become a standard practice • The organisation captures metrics to identify the need for updates to blueprint information • EA plans are reviewed and changes are incorporated to improve the EA Program • The organisation captures metrics to measure the progress against the established EA plans • Goals are being set for the future of the EA Program Plan • Governance roles and responsibilities are reviewed and updated to incorporate changes to the EA Framework • Formal processes for managing variances feed back into architecture • EA awareness training is incorporated into new employee orientation • The organisation captures metrics to measure the effectiveness of the EA team • Senior management directly involved in the architecture processes • Senior Management participate in various EA committees • EA framework and outcomes are regularly evaluated • Meetings are held regularly to review modifications to the EA framework and outcomes

1.9.1.1 EA documentation

Dubai Customers was documented comprehensively at the strategy, process, information, resource, and technology layers. The documentation was stored in the IBM System Architect. Another tool, enterprise connected view (ECV), was built to represent the stored information in an easy, intuitive way for business and technical stakeholders. Participant [P-2] said:

> *"So we actually quite extensively document all the artefacts or all the information in the organisation in that tool… we have actually developed a dashboard… that is enterprise connective view (ECV)".*

Participant [P-4] further highlighted the value of using the as-is models for any project and particularly for SOA projects: "The first task in each phase is to use the existing architecture and that is applicable for SOA projects.

1.9.1.2 EA planning

Dubai Customers strategically used EA to help it move forward in different aspects. EA was adopted to align strategy, business, and IT, which enabled it to respond quickly to changes in business and IT requirements. The EA program was well defined with a structured framework and timeline for developing the EA. EA activities were carried out according to the defined plan. EA planning was actioned through demand management where all projects and initiatives are assessed against the captured strategies and architectural plans. Participant [P-6] stated:

> *We want EA to guide the organisation from the 'as-is' state to 'to-be' state in all dimensions: planning, business, and technology".*

Participant [P-4] noted that EA planning was well integrated with major strategic initiatives to help the organisation successfully implement these initiatives. EA was involved in building roadmaps for organisational improvements. EA was used to assess the current situation, identify gaps, and build roadmaps and action plans. The high-level EA planning methodology is represented in Figure 6.8. It starts with an assessment and documentation of current (as-is) architecture. Such assessment identifies gaps and builds a future architecture (planning). The planning phase was executed through projects and its execution is monitored and evaluated. The whole process was governed using well-established governance procedures [D-13].

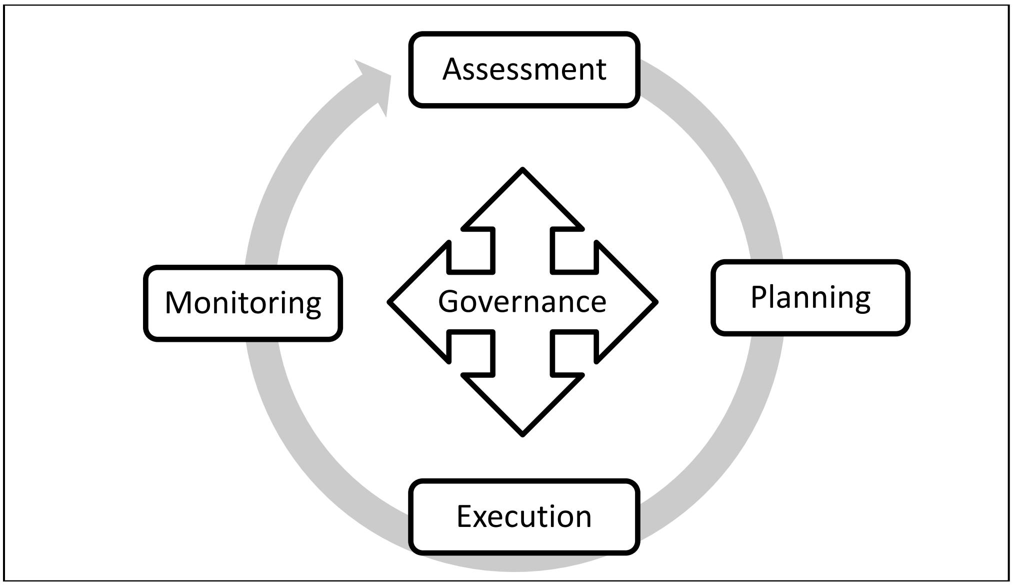


Figure 6.8 Adopted EA methodology

1.9.1.3 EA governance

Architecture governance was well established in Dubai Customers from the beginning. EA roles and responsibilities were defined. Formal processes for managing changes feedback into architecture were also established [D-1, D-2]. IT strategist [P-3] stated that *"the governance is ensured during the whole EA process"*. Architectural governance was also emphasised in the EA methodology (see Figure 6.8), where every step of the EA methodology is governed.

Architecture governance was also achieved through demand management (ensuring demands were aligned with the organisation's strategic objectives). It also ensured that demands were aligned with existing business and technology architectures [D-1]. Demands were raised by different business divisions or departments. If approved, the demand became a part of the project portfolio, where a demand was translated into one or many projects depending on the complexity of the demand. Participant [P-5] elaborated on the importance of EA's involvement in demands management. He reported that the EA team was the custodian of the whole picture of the organisation, and the team needed to identify the changes that these demands cause.

Changes to EA as a consequence of demands and projects are assessed on the initiation of these demands/projects. The results of the assessment determined whether the architectural changes are needed. If needed, EA is updated to reflect the changes that are introduced by the new demands (see Figure 6.9).

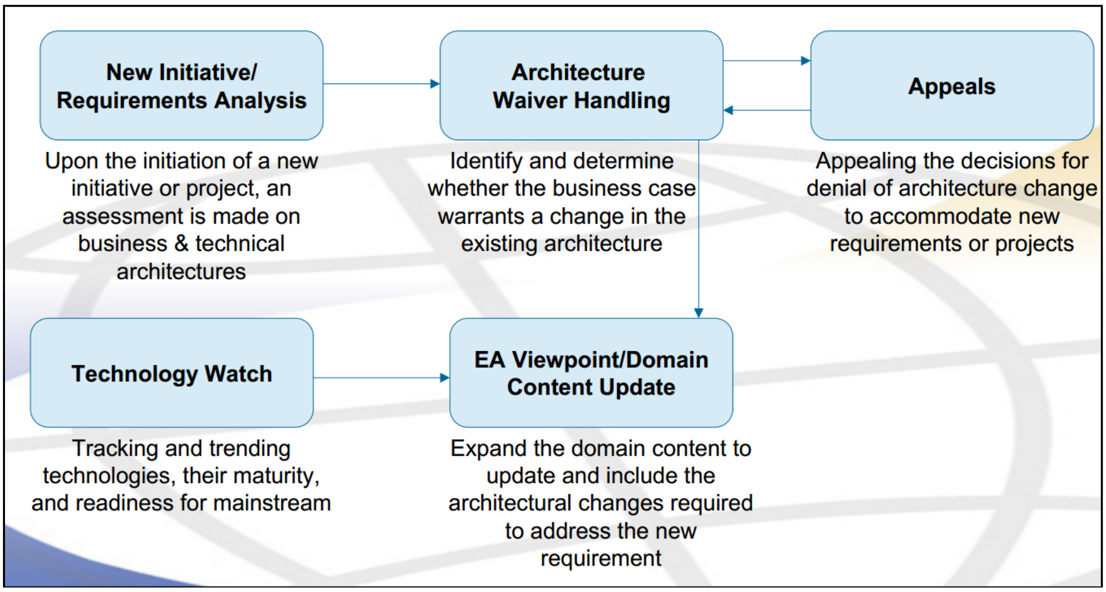


Figure 6.9 EA governance [D-1]

1.9.1.4 EA evaluation and maintenance

The governance aspects discussed above also add to this maturity dimension. The employed governance practices kept EA processes and outcomes engaged and up-to-date. At Dubai Customs, EA and its products were evaluated on two sides. First, EA, its methodology, and meta-model were reviewed and assessed every two years. Second, EA was reviewed and changed when needed; for example, if there were new trends or requirements that needed to be addressed from an EA point of view. Participant [P-5 described the practice as follows:

> *It's based on the requirements, directions and new issues coming... we decided internally that we will review our EA activities every two years.*

1.9.1.5 EA team and resources

The EA team at Dubai Customs was diverse and sufficiently resourced. The EA team included multiple members of strategy, business, and IT backgrounds. There were IT strategists, business architects, security architects, information architects, and technical architects. The team was led by an experienced EA leader. The EA team had very qualified people who had been working in business and IT for years. The team was supported by the head of the division and the director of Dubai Customs. The team's major task was to look at the organisation as a whole, and to document its strategic, business, and technical elements and their relationships. The team also contributed to demand projects, ensured its architectural fit and architectural compliance. It made sure that projects were aligned with the organisation's strategy and with its business and IT standards. Moreover, the EA team organised workshops to improve the understanding of EA in the organisation, and to involve business and IT. Participant [P-6] stated:

> *On a regular basis, we run workshops to re-explain what is EA, what are the EA offerings, what is the benefit of the EA.*

1.9.1.6 EA business support

EA at Dubai Customs was fully supported by the Managing Director of Customs [D-1]. Their EA had a strong leadership with a strong desire to achieve EA benefits. Their EA was well supported by people who had power and understood EA's value to align corporate strategy and IT strategy.

A report about early EA implementation [D-14] declared that the EA program has strong business support:

> *As part of this commitment, the senior management launched an initiative to transform the organisation and improve its responsiveness. At the heart of this initiative was the implementation of enterprise architecture designed to improve quality control and enhance the specification of core business processes.*

To conclude, based on the findings of the interviews and the answers to the accompanying survey regarding EA maturity, it is evident that Dubai Customs' EA practices are mature when considering the fact that EA practices internationally are still emerging and in the early stages of maturity (Gartner, 2012b).

1.9.1.7 Summary

This section summarises the architectural conditioning aspects. Dubai Customers adopted EA in 2006 in three phases. By the end of 2007, most of the EA implementation was completed and architectural artefacts were stored in the EA repository. The organisation's framework included the strategy, process, information, resource, and technology layers. It was internally developed using TOGAF and Zachman principles. EA was strategically driven to achieve strategic, operational, IT, and governance objectives. The maturity of EA practices was between levels three and four out of five on the adopted maturity model. EA was organisation wide, well managed, and business driven and supported. It was integrated with demand (projects) management.

The findings of the architectural conditioning phase build the foundation of the analysis to understand how EA evolved due to SOA's introduction. Following the architectural conditioning phase, the architectural interaction (SOA's introduction) began in 2008 and lasted for two years. The findings of the SOA introduction phase are presented in Section 6.6.

6.6 Architectural Interaction: SOA's Introduction

This section presents the findings relevant to the second analytical dimension of Archer's (1995) morphogenetic theory: the introduction of SOA into the organisation.

During this architectural interaction phase, other activities related to EA could be in process. However, the scope of the study covers only SOA-related activities and the outcomes of its integration into EA. Furthermore, in this case, participants did not mention another large-scale event other than SOA's introduction that could have significantly affected the observed evolution outcomes, nor were any identified in the obtained documents.

6.6.1 Overview of SOA's Introduction

Prior to the SOA's introduction, trade growth in Dubai was mounting at a rate that was not sustainable with Dubai Customs' delivery of services. Therefore, a new simplified and improved way of service delivery was required. In order to meet the challenges of the rapid growth of the Emirates (particularly at the trade level) and to implement the vision of becoming an international trading hub, Dubai Customs took initiatives to improve its service delivery. In particular, an initiative from the Dubai Government in 2008 to deliver eServices to citizens, called eServices Delivery Excellence Model [D-6], accompanied by internal thoughts to embrace what an IT strategist [P-2] called an "outside-in strategy", had led to changes in the organisation and its EA. The outside-in strategy is as an external view of an organisation (customers' needs) and its restructuring based on that view. For example, some participants described the move to service-orientation:

> *But later we noticed, we were lacking in this area and the global trend is purely from service-orientation perspective, even at the Dubai government, even at the emirates government level, they only talk about services* [P-4]
>
> *The challenges at the IT level implementation is until you define your services at the business level, customers' level, you can't do them right* [P-2]

Thus, Dubai Customs undertook a service-oriented initiative in 2008 that was officially launched in 2010 [D-15]. Their adoption of SOA to develop an electronic customs' declaration system was regarded as a world-leading customs suite that used Oracle's SOA suite and followed SOA design principles (Oracle., n.d.). The Customs suite was implemented using a combination of IT services, processes, and procedures to provide automated and paperless services. The system worked as an engine that managed all operations, including risk management and duty collection. It electronically connected other agencies in the Dubai trade supply chain, and provided electronic messages related to inspection, risk, and clearance of goods [D-12]. Dubai Customs implemented its SOA suite with large vendors such as IBM and Oracle. According to a report from Oracle (Oracle., n.d.):

> *Dubai Customs required a new operating model, based upon a service-oriented approach, to deliver the flexibility and scalability needed to accommodate existing and projected levels of trade. Additionally, to increase efficiency and reduce unneeded use of internal resources, Dubai Customs aimed to implement a Web-based, self-service system for its customers that would enable them to manage customs declarations seamlessly, rapidly, and accurately.*

SOA introduction was a web-based, scalable, and feature-rich business-to-government suite for Dubai Customs' customers and partners. Dubai Customs was among the pioneer enterprises worldwide for using Oracle's SOA Suite for this purpose. The implemented SOA is considered a world-leading customs suite designed and developed in compliance with world customs organisations' best practices. Moreover, a risk engine was developed for online risk scoring and assessment, which was capable of processing online assessments of tens of thousands of declaration transactions per day (Oracle., n.d.; Zawya., 2011) and more than 2000 companies used the system every week [D-12].

The project has been large: a great deal of funds, effort, and time has been put towards the success of its implementation [D-12]. At the beginning of the project, during the design phase, Dubai Customs contracted external vendors to plan and develop its requirements. It was developed by a competent internal work force with the help of external expertise when needed. In a nutshell, the SOA suite was developed using a combination of in-house development and products from vendors such as Oracle, IBM, and Microsoft [D-12]. Further, in order to secure SOA's successful implementation, employees were trained, the organisational restructuring was completed, and the assessment and training section was established. This initiative was managed and supervised by experts to facilitate the transition to the new system. A selection of clients representing all trade sectors were involved in the SOA's development, primarily on the design of the system. Multiple focus group meetings were held with clients to announce the new procedures.

The new electronic paperless declarations system facilitated the declaration process. It allowed goods to be cleared without the old manual paper-based handling system. If special documents are necessary or customs inspection is required, the system sends a "hold clearance" message to the cargo handler, and awaits the completion of the necessary steps to complete the declaration [D-11]. The implemented SOA suite was used by a wide range of divisions and people who participated in customs-related operations, such as importers, exporters, and declaration agents. The suite has offered a straightforward means to manage declarations and provided notifications of a particular consignment's movement [D-12]. It has provided several benefits to Dubai Customs and its clients. First, paperless declarations could be made remotely 24/7 with an electronic signature, which reduced hard copies of documents. This enabled the pre-clearance of goods and reduced service delivery times and supported clients to modify and cancel declarations online [D-11]. Secondly, its implementation helped reduce costs to importers, exporters, and the freight community through speed and streamlined import and export trade. Thirdly, it had an intelligent self-learning risk assessment engine with validation rules at the core of the system. The risk engine, along with the profile management and profile creation modules, applied selective and random profile matching on declarations. The adoption of predictive modelling profiling of the risk engine currently positions Dubai Customs at the forefront of all customs administrations globally in the way that it manages declarations [D-11].

The previous paragraphs have provided an overview of Dubai Customs' SOA activities, particularly the implementation of a service-oriented customs suite, between 2008 and 2010. The following subsections examine SOA introduction-related action-formation generative mechanisms.

6.6.2 View of SOA

There were two perspectives that influenced SOA's introduction at Dubai Customs. ***First***, SOA was considered from a business point of view. It was a business-driven initiative in response to the eServices delivery initiative from the Dubai Government. A senior business architect [P-5] described that view:

> We are a service oriented organisation on a broad basis, so we have customers who require services only. So when they require services we have to align and position our information, assets and processes in that way.

SOA's implementation offered an example of this viewpoint of SOA, where business and IT perspectives were combined to deliver SOA's value. The implemented SOA suite included IT services and business processes and offered business services to clients to manage declarations. It also connected the customs agency's value chain with other administrations involved in the declaration process [D-12]. SOA suite was:

> *A combination of IT services and Dubai Customs processes and procedures provided through an electronic environment in an automated and paperless way. The system works as an engine which manages all operations including collecting Customs duties* [D-12].

Further, IT strategist [P-2] stated that SOA influences the business side of the organisation as much as it does the technical side. He stressed that SOA should be adopted from both sides in order to achieve better alignments between business and IT:

> … the lack of adoption of SOA is [because] it has been practised by core technical people. So the people to whom it actually matters are the business people. [P-2]

He even argued that services identification and definition has to start from the business. Once business services are identified and defined, it becomes easier to map them to the technical services. He stated:

> It [SOA] has to actually come from the business and you have to define your services at the business level and break them down… to align business services to the technical services. [P-2]

Second, according to a senior technical architect [P-7], SOA is an architectural style that is adopted in the organisation among other styles: "Architecture governance includes SOA which is an architectural style that we use".

Another senior technical architect [P-8] declared that SOA has a business side, and that, in Dubai Customs, it was implemented based on both business processes and technical levels: "*There are business aspects of SOA; as I said, from EA perspective we have SOA suitably implemented…. SOA is done at business processes and a technical services layer*".

In conclusion, it appears that the view of SOA at Dubai Customs fits under the enterprise services architecture view (encompassing business and IT levels) as presented in Chapter 2, where SOA was introduced to redesign and align both business and IT architectures.

6.6.3 SOA Perceived Benefits

This section examines the SOA perceived benefits that were associated with SOA introduction at Dubai Customers. One of the main perceived benefits was to increase customers' satisfaction and delivery of end-to-end services. Customers' needs were considered and the structure of the business was aligned with what the customers wanted. IT strategist [P-2] described the challenge the organisation faced before it adopted the service-orientated approach to deliver services to their customers:

> *The challenges at the IT level of implementation is until you define your services at the business level, customers' level, you can't do them right… We were thinking inside-out. We were thinking whatever we were doing is something which our customers needed.*

SOA implementation offered multiple access channels for clients [D-12]. The processing of declaration became quicker and thus improved the delivery of services to clients. SOA implementation also improved the organisational agility. Another reported benefit from SOA was improved information capturing and availability capability. The captured information was useful for planning and improvements. SOA implementation also efficiently improved systems integration and thus improved the organisation's declaration, inspection, and payments practices [D-11]. At the technical level, reuse was one of the major drivers for SOA. Table 6.5 shows a sample of the supporting quotes.

Table 6.5 Quotes of reported SOA benefits at Dubai Customs

Reported benefits	Quotes

Increased customer satisfaction	"a quick online response means goods are able to move quicker without the need for Customs control" [D-11]
Enablement of new functionality	"offers the client the ability to process their declaration even before the goods arrive in Dubai" [D-11]
Agility	"Dubai Customs required a new operating model, based upon a service-oriented approach, to deliver the flexibility and scalability needed to accommodate existing and projected levels of trade" (Oracle., n.d.)
Reuse	"The advantage, when you split them [applications] into services, is reuse" [P-7]
Increased availability of information	"Goods are classified and validated during the declaration process... to capture the data leading to better defined trade data. Once published, these statistics can then be used for wider business uses (i.e., strategic planning)" [D-11]

In summary, SOA introduction was associated with perceived benefits at the strategy, process, and IT levels, which influenced its introduction. As the implementation was completed, the participants and the implementation-related documents showed that these anticipated benefits have been achieved. Table 6.6 shows the mapping of the identified perceived benefits to the adopted classification of SOA benefits in this study.

Table 6.6 Reported SOA perceived benefits at Dubai Customs

SOA benefits layers	SOA benefits	Reported
IT level	Reuse	x
	Facilitation of software development	
	Improved IT integration	x
	Reduce complexity	x
	Improved project management	
	Better assets utilisation	x
	Reduce maintenance costs	x
Process (operational) level	Increased availability of information	
	Reduce maintenance costs	
	Increased customer satisfaction	x

	Business process improvement	x
Strategy level	Agility	x
	Business-IT alignment	x
	Enablement of new functionality	x
	Improve communication	x
	Reduce time to market	

6.6.4 SOA Scope

SOA's introduction was driven by an organisation-wide scope. A report of the SOA implementation stated: *"It has been a significant project. The launching of this system required large numbers of funds, energy, time and working hours. There are more than 2000 companies which use the system weekly"* [D-12].

According to a senior technical architect [p-7], SOA-related projects were still open due to the maintenance of existing services and the addition of new services: *"Even today we are adding services. So this development is not closed today… maintenance is still happening, new services are added, so that is continuing"*.

6.6.5 SOA Governance

SOA's introduction was governed on multiple levels. First, COBIT was an adopted governance framework at Dubai Customs in general, which governed the organisation, including its SOA. Further, SOA's implementation was governed by established standards and guidelines for services development, enablement, and evaluation. One of the examples of such governance practices is the use of the eServices Delivery Excellence Model (EDEM). It was established by Dubai government for:

> *the electronic enablement (eEnablement) of government services. This model will not only serve as the guiding principle for Service eEnablement but also provide a foundation for the eServices Evaluation Project* [D-6].

EDEM governs and evaluates services on different issues such as security, privacy, usability, ease of service delivery process, performance, reliability and connectivity, service access, delivery, and execution in order to provide a seamless and user-centred service delivery. Also, Dubai Customs adopted a SOA reference architecture based on IBM reference architecture. Its purpose was to guide the architectural adoption of SOA and its principles: *"So those architecture principles that we use, it has to follow the SOA reference architecture. So we use SOA reference architecture, to put set of rules around how to build the architecture"* [P-7].

Moreover, SOA projects were similar to other projects governed from an EA perspective against business and IT architectural practices. According to a senior technology architect: *"What we do is we do governance on the whole architecture so architecture governance includes SOA which is an architectural style that we use"* [P-7].

6.6.6 SOA Design

This section examines SOA design aspects related to SOA introduction, which include services identification, services classification, services catalogue, and SOA roadmap.

During the initiative, SOA reference architecture was used to guide SOA's design. Dubai Customs also defined a clear roadmap for adopting SOA for internal and external improvements. According to a report from Oracle:

> *Dubai Customs has drawn a clear strategy to adopt SOA to support Dubai's trade growth and set up an enterprise solution for customs declaration processing. Based on this strategy, Dubai Customs selected Oracle SOA Suite to establish a centralized, Web-based suite of customs solutions that enabled the organization to achieve exponential scalability to support trade growth and improve internal collaboration* (Oracle., n.d.).

Services were identified at business and IT levels using a top-down approach. These services were classified into business and technical services. Business services are part of the business architecture. They were further classified into tier one, tier two, and tier three business services (see Figure 6.10). Tier one services were customer-facing services that were *"initiated by the customer and for which the customer receives an output"* [P-3]. Tier two services were services that other services, not customers, invoke. Tier three services were internally provided services such as payroll services, HR services, and "enterprise maturity assessment", which EA provided to the organisation as an internal service. Dubai Customs had more than 180 defined business services and only about 15 services were customer-facing services. The rest of the business services were tier two and tier three.

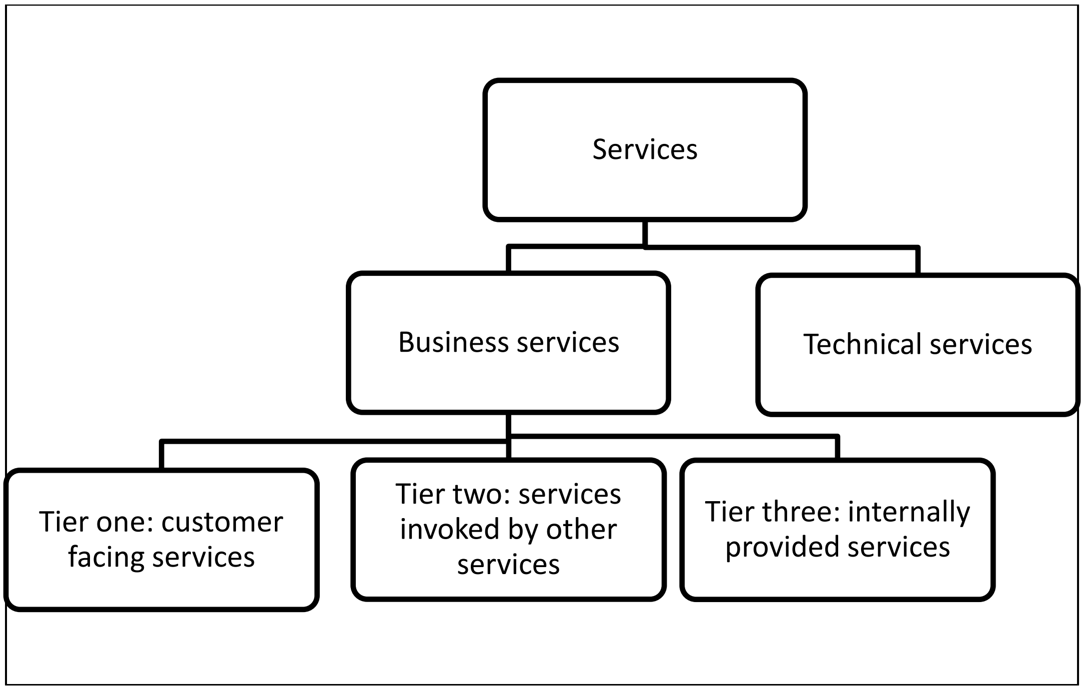


Figure 6.10 Services classification at Dubai Customs

Services, business and technical, were stored in IBM System Architect with other EA elements and their relationships. Reports could be generated to show services, their descriptions, and their relationships with other EA elements such as processes and applications.

6.6.7 Business and IT Collaboration

The business and IT collaboration level influenced SOA's introduction at Dubai Customs. For example, clients and key strategic stakeholders were involved in SOA design. According to one report:

> *Dubai Customs developed Mirsal 2 over a time period of two years in-house. This system has been designed in line with clients and key strategic stakeholders' needs. Seeking to achieve the utmost benefit, Mirsal 2 aims at furthering services delivery standards, at reinforcing cooperation with partners and at providing an advanced electronic environment for Dubai Customs employees* [D-15].

Further, a skilled business and IT team was recruited to drive SOA's introduction to ensure its successful implementation: *"To achieve Dubai Customs' strategy in adopting SOA …. we have acquired highly skilled and qualified resources in implementing SOA solutions"* (Oracle., n.d.).

Moreover, training programs were delivered, organisational restructuring was completed, and the implementation was managed by experts to ensure that SOA was successfully adopted and to facilitate the transition to the new environment [D-12]. Dubai Customs re-assessed its structure and introduced new jobs. Employees were trained where needed to operate effectively in the new operating model [D-11].

Dubai Customs also used a demand management process, which involved the EA team in any demand including SOA. Any demands come to the business architecture team first, and they assess the demand against the architectural standards and as-is information. Then, they produce the business requirement specifications. Next, the demand proceeds to the technical architecture team for review and production of systems requirements specification. If approved, the demand becomes project/projects and EA is involved in monitoring the progress of the project, its deliverables and its adherence to specifications. SOA's introduction was no different from other demands. A senior technical architect [P-8] argued that aligning the business and IT teams was required to drive SOA's introduction. As another senior technical architect [P-7] stated, such collaboration was essential for SOA design and implementation: *"The way it is important for the SOA is if you know the business requirements around a service you can get services design and service orchestration right"*.

6.6.8 Summary

In summary, this section examines the six action-formation generative mechanisms that influenced SOA's introduction at Dubai Customs. First, SOA's introduction was driven by both business and IT perspectives of SOA. It was undertaken to redesign business processes and improve services delivery. Based on the SOA classification introduced in Chapter 4, Dubai Customs' SOA implementation fits within the enterprise service architecture perspective. Second, SOA implementation was associated with perceived benefits at the strategy, process, and IT levels, and most of these benefits were reported as achieved in the SOA implementation documentation and by some participants. Third, SOA's introduction encompassed the whole organisation and lasted for two years. Fourth, SOA's introduction was governed on multiple levels. It was governed by the eServices delivery excellence model and organisational wider governance practices (COBIT). Fifth, SOA was introduced based on a long-term roadmap that employed a defined SOA reference architecture. Services were identified using a top-down approach and were classified into business and technical services. Also, a service repository (IBM System Architect) was employed to track these services in relation to other architectural elements. Finally, SOA's introduction was a large project that involved key business and IT stakeholders during design and implementation. It was supported and driven by the organisation's top management as part of a transformation initiative to improve services delivery. External and internal (business and IT) highly skilled team was involved in SOA implementation. Table 6.7 summarises the SOA introduction-related generative mechanism in the context of Dubai Customs.

Table 6.7 Summary of SOA's introduction at Dubai Customs

Generative mechanisms	Description
View of SOA	• SOA's introduction was driven on both business and IT levels • It was undertaken to redesign business processes and improve services delivery • According to the employed view of SOA classification, SOA's implementation fits in the enterprise service architecture perspective (Business and IT levels)
SOA perceived benefits	• SOA's implementation was associated with perceived benefits at the strategy, process, and IT levels • Most of these benefits were reported as achieved in SOA implementation documentation and by some participants
SOA scope	• SOA's implementation was organisation-wide • It lasted for two years
SOA governance	• SOA's implementation was governed on multiple levels

	• It was governed by the eServices delivery excellence model • It was governed by the adopted IBM SOA reference architecture • It was governed by internal governance practices (COBIT) and EA governance
SOA design	• SOA was implemented using a long term roadmap • It employed a defined SOA reference architecture. Services were identified using a top-down approach • Services were classified into business and technical services • A service repository (IBM System Architect) was adopted to track these services in relation to other architectural elements
Business and IT collaboration	• Business and IT stakeholders were involved during design and implementation • It was supported and driven by the top management of the organisation as part of a transformation initiative to improve services delivery • Diverse skilled teams were involved in SOA's implementation • External vendors and consultants were involved in SOA's implementation

Section 6.7 examines the architectural elaboration (outcomes) that resulted due to SOA's introduction.

6.7 Architectural Elaboration: Reproduction or Transformation

Using the last phase of the theoretical model, this section discusses architectural elaboration (EA evolution outcomes). The pre-existing architectural settings are either reproduced or transformed on five levels as discussed in previous chapters. These levels are business architecture, Information Systems architecture, technology architecture, EA governance, and EA methods and tools. Following the data analysis, the observed architectural elaboration (outcomes) in Dubai Customs are summarised in Table 6.8.

Table 6.8 The observed architectural evolution

Architectural transformation	Description

level	
Business architecture (transformed)	• Process layer became "business layer" to incorporate business services besides the other elements of the business architecture • Re-design of the organisation in terms of domains, and each domain has its provided services • Design of business architecture in terms of services • New SOA-related elements were added to business architecture, such as business services, their descriptions, supported channels, client groups, service scenarios, and owners • Business services were mapped to other business architecture elements • Business services viewpoints were added
IS architecture (transformed)	• Applications were designed and documented in terms of technical services that support business processes and services • A technical service was represented, which had a schema, used a service operation, and had a service realisation diagram • Technical services were aligned and used by business processes and services on the business architecture • Granularity of technical services was considered at the design level to ensure proper reuse • Services were used to integrate internal systems and external systems such as external payment services • Use of SOAP protocols, WSDL for services description and XSD for services schema definitions • Technical services were mapped to business processes and supporting infrastructure
Technology architecture (transformed)	• SOA infrastructure such as BPEL engine, web services manager, and ESB documented using technology environment, instance, interface, interface messaging, and message structure • Use of services-related communication protocols such as SOAP and services security protocols such as WS-security to document used SOA protocols • Service repository (integrated into IBM System Architect) that hosts the meta-data of services and related information

	• Services/infrastructure mapping to show the infrastructure that supports services • Services SLAs were configured and monitored at the application and the infrastructure layer to make sure that the SLAs were met
EA governance (transformed)	• EA covered governance aspects regarding demands management and alignment with strategy and architectural standards • SOA (and its projects) had its own governance frameworks that were aligned with the overarching EA governance • EA governed service documentation, service identification and service delivery • Services were monitored using the orchestration engine • SOA demands were also governed by EA, similar to any other demands, against the architectural standards and strategy • Every service was governed by technical and business SLAs • Every business service had an owner
EA methods and tools (transformed)	• EA was integrated with demands/projects, which include SOA projects New SOA-related elements and new relationships were created in the used EA tools. • New views were created in used EA tools to support services and associated elements • Service identification methods and services were identified using EA products (repository)

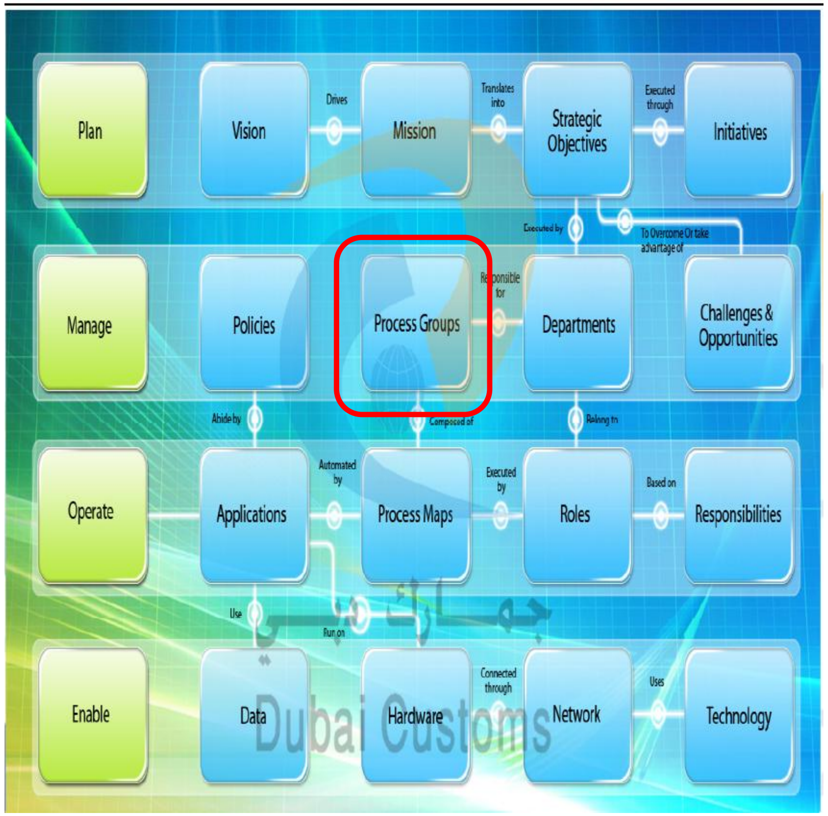
6.7.1 Business Architecture (Transformed)

This section discusses architectural elaboration on the business architecture level. This level of EA was transformed after SOA's introduction (SOA was integrated into the business architecture).

In Dubai Customs, SOA's introduction was an organisation-wide initiative and business driven. Dubai Customs' EA structure slightly changed after SOA's introduction. The "process layer" became the "business layer", which included services in addition to business processes. At the business architecture layer, business services were identified and then aligned through business processes to the technical services at the technical architecture layer. This approach for services identification and alignment was chosen to achieve better service delivery to Dubai Customs' partners and customers, according to senior business architect [P2]. He argued that focusing completely on the small technical components would result in only marginally achieving the bigger picture, *"the end-to-end service delivery"*. Moreover, for improved service delivery, business services were grouped into domains. The organisation was organised into core domains (client, declaration, intelligence, compliance, and enforcement) and supporting domains (planning, IT, HR, and governance). For example, the client domain had services targeting clients such as client registration and client licensing.

Business architecture incorporated elements related to SOA such as business services, their classification, descriptions, owners, and client groups. Additionally, the enterprise connected view (ECV) had services added to them as a major element (see Figure 6.11).

Snapshot of ECV in 2007 [D-2]



Snapshot of ECV in 2011 [D-7]

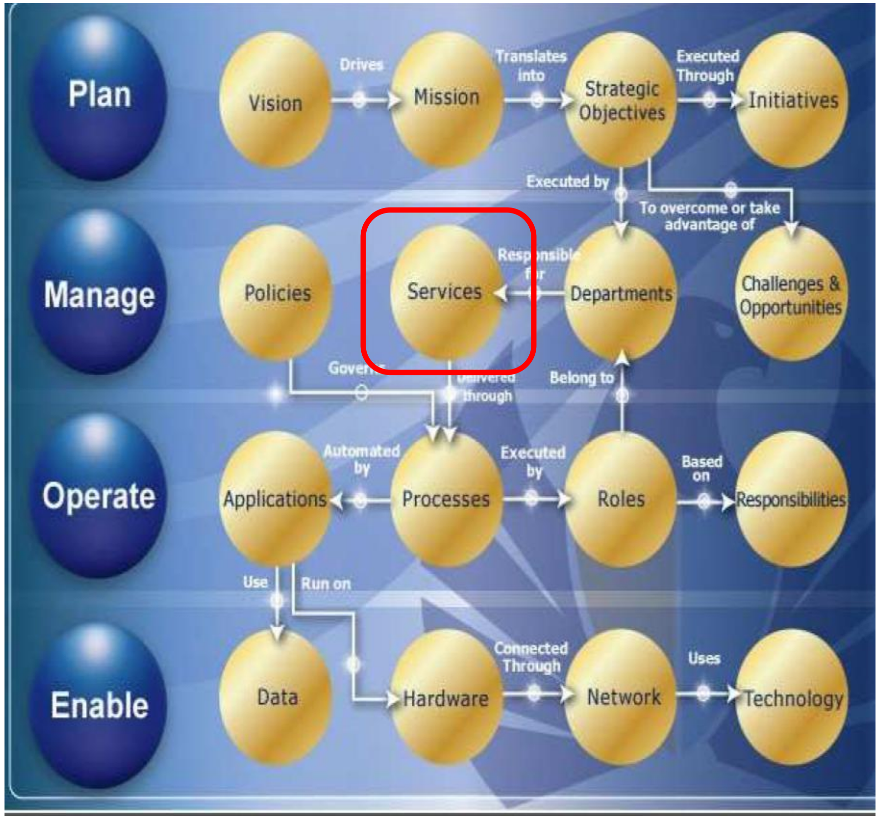


Figure 6.11 Enterprise-connected view changes

Each business service had a description, an owner (the business unit that provides the service), the location of the service, and the supported channels (touch points). Service variations were documented as well. If two services delivered the same outcome but done differently, this was considered a different service scenario: *"If two similar things are being done and they deliver the same output then it is the same service that has two scenarios"* [P-3].

An example of a service with multiple scenarios was the declaration service. Clients used the declaration service to obtain a clearance (an outcome). However, there were various scenarios to obtain the clearance. One scenario was when a client imported something from somewhere else in the world to the local market. The other scenario was when a client imported something from somewhere in the world to the free zone. For each scenario, there could be different duty structures, different documents required, different procedures, or different people involved to obtain the clearance.

As discussed earlier, business services were classified into three tiers. An example of a customer-facing service was declaration, an example of services invoked by other services was inspection, and an example of internally provided service was EA maturity assessment (see Figure 6.12).

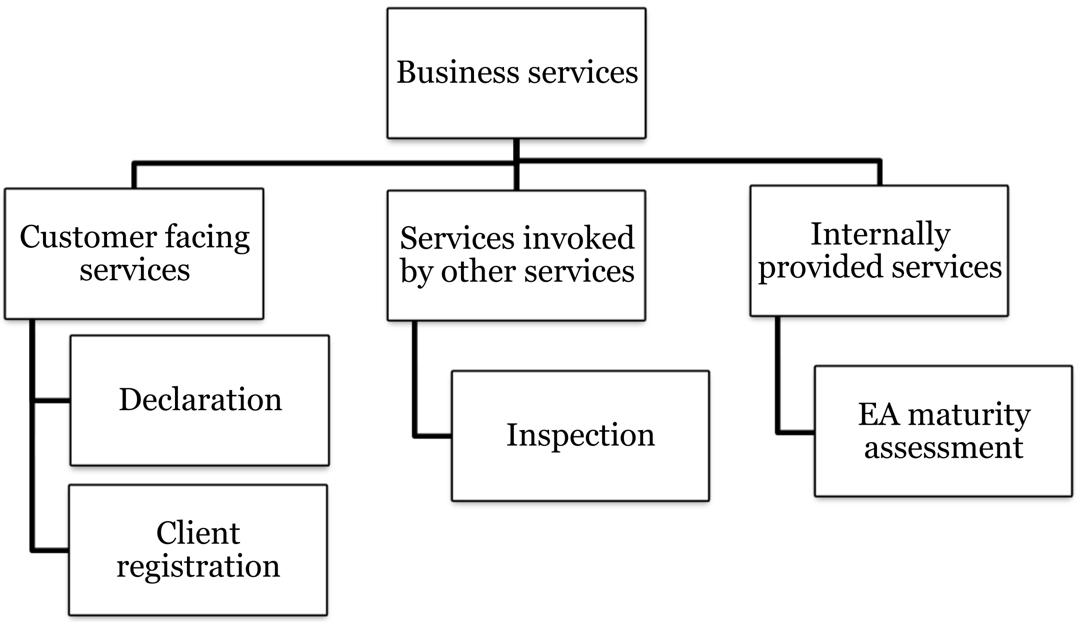


Figure 6.12 Business services types

According to the head of IT planning and business architecture [P-1], business services were designed in such a way that they could be accessed by multiple channels such as user interfaces, business-to-government, and couriers. Business processes were exposed as reusable and loosely coupled services and orchestrated using BPEL. Figure 6.13 shows a snapshot of the Dubai Customs' business services accessible from their website channel.

Figure 6.13 Business services accessible on the Customs website

In summary, the business architecture was transformed (evolved) by SOA's introduction. Table 6.9 summarises the aspects of the business architecture transformation.

Table 6.9 Business architecture

Business architecture (transformed)
• Process layer became "business layer" to incorporate business services besides other business architecture elements.
• Organisation redesigned in terms of domains and each domain has its provided services.
• Design of business architecture in terms of services.
• New SOA-related elements were added to business architecture, such as business services, their descriptions, supported channels, client groups, service scenarios and owners.
• Business services were mapped to other business architecture elements.
• Business services viewpoints were added.

The following two sections examine SOA's integration into the information systems and technology architectures. Although the applications, data, and infrastructure are all labelled "technology architecture" in this case, they are separated for purposes of analysis and comparison, following the decision stated earlier in Chapter three.

6.7.2 Information Systems Architecture (Transformed)

This section presents the integration of SOA elements into the information systems (IS) architecture (applications and data). The IS architecture was transformed due to SOA's introduction. SOA's introduction added some elements to the IS architecture. According to the head of IT planning and enterprise architecture [I-1], technical services were designed in such a way to ensure proper granularity to help in the re-use and re-orchestration of the services and to enhance or develop business processes.

In the meta-model, a technical service was represented as having a schema, using a service operation, and possessing a service realisation diagram. These elements were integrated with other architectural elements. These technical services support business services through business processes. For example, the above-mentioned business service of customs declaration was mapped to business processes and then implemented using multiple technical services such as submit declaration, validate declaration, calculate charges, and identify risks.

Integration with third party systems was also defined using, for example, web services and messaging to build truly decoupled systems. Clients, through partner systems, submitted declarations electronically. The message structure was also determined, and Dubai Customers ensured that it adhered to the organisation's XML Schema. In response to the posted declaration, declaration confirmation, and detailed declaration response, messages were sent to the declarant system. The response included details of duty, deposit, declaration status, and other relevant information. Then, cargo clearance or hold information was sent to the cargo declarant and handler.

In summary, using the criteria prepared in advance, the IS architecture integrated SOA-related elements. The main element of SOA in the IS architecture was the technical service, which was associated with the other architectural elements. Table 6.10 summarises the aspects of SOA's integration into the IS architecture.

Table 6.10 Information systems architecture (transformed)

Information systems architecture
• Applications were designed in terms of technical services that support business processes and services • A technical service had a schema, used a service operation, and had service realisation diagram • Technical services were aligned and used by business processes and services on the business architecture • Granularity of technical services was considered at the design level to ensure proper reuse • Services were used to integrate internal systems and external systems such as external payment services • SOAP protocols, WSDL for services description, and XSD were used for services schema definitions • Technical services were mapped to business processes and supporting infrastructure

6.7.3 Technology Architecture (Transformed)

This section presents aspects of SOA's integration into the technology architecture. The technology architecture had elements to represent SOA and other existing infrastructure such as technology environment, instance, interface, interface messaging, and message structure. These elements had relationships with other business and technology architectural elements. The supporting infrastructure was mapped to business services and technical services.

Dubai Customs also adopted Oracle SOA suite as a major infrastructure of their SOA environment. A senior technology architect [P-7] described Dubai Customs' Oracle SOA suite as: *"A set of all you need to run a SOA environment, so it has application server which is again Oracle application server then we have the Oracle BPEL, Oracle WSM, and Oracle database"*.

BPEL infrastructure was set up to orchestrate business processes by using Oracle BPEL manager. It had BPEL process designer, BPEL process manager runtime, and other related components. Moreover, different protocols are used for communication between services such as SOAP over HTTP(s). The message structure is XML based: *"The Dubai Customs* *supplied WSDL & XSD, which will enable partners to implement a web service, call mechanism to submit Declaration Message"* [D-16].

In summary, SOA was integrated into the technology architecture (see Table 6.11 for detailed aspects).

Table 6.11 Technology architecture (transformed)

Technology architecture
• SOA infrastructure such as BPEL engine, web services manager, and ESB documented using technology environment, instance, interface, interface messaging and message structure. • Services-related communication protocols, such as SOAP, and services security protocols, such as WS-security, were used • Service repository (integrated into IBM System Architect) hosted the meta-data of services and its related information • Services/infrastructure mapping to show the infrastructure that supports services • Services SLAs were configured and monitored at the application as and the infrastructure layers to ensure that the SLAs are met

6.7.4 EA Governance (Transformed)

This section presents SOA governance's integration into EA governance practices. In Dubai Customs, SOA had its own governance practices, which were aligned with existing overall organisational governance (COBIT framework) and EA governance. Participant [P-7] commented on the organisation's use of SOA reference architecture: "*We use SOA reference architecture, to put a set of rules around how to build the architecture and to govern SOA development"*.

According to senior business architect [p-5], EA was strongly involved with demands and projects governance, which included service-oriented demands. EA ensured that demands, including SOA demands, were aligned with Dubai Customs' strategy, business, and technical architectural principles. Senior business architect [P-5] said: *"EA is there to help decisions makers and ensure governance around projects and demands"*. Further: *"So basically SOA governance falls under the enterprise architecture and based on that we govern it, so it is a shared responsibility"* [P-7].

EA also had all the standards and principals that applied to services, processes, and IT. Senior business architect [P-6] described the integration at this level:

> *From a business side, we have process standards and process guidelines. We have service standards and service guidelines... so these govern how to document the processes and services in the organisation.... They govern services documentation, services identification and services delivery.*

Participant [P-7] elaborated on the role of EA in relation to the Service lifecycle governance:

> *The way governance is done in the service life cycle is you start with business process decomposition. So what they do is they make sure that the whole process has been documented correctly. Once that is done we review the business requirements specification. We translate these processes into activities then you do SRS where you translate these activities into use cases and those use cases and they're realised in solution architecture.*

Each of the services was governed by various SLAs that were both technical and business in nature. These SLAs were configured and monitored to make sure that the SLAs were met as prescribed by the organisation [P-1]. According to senior technology architect [P-8]: *"Services are monitored using services orchestration engine... So, we know what is the service, how is this service invoked and who invokes it"* [P-8].

In summary, SOA had its own governance practices and frameworks, such as the eServices delivery excellence model and SOA reference architecture. SOA governance was extended, and aligned with the overall EA governance practices. Table 6.12 shows the integration between both SOA and EA governance practices.

Table 6.12 EA Governance

EA governance
• EA covered governance aspects regarding demands management, including SOA demands) and alignment with strategy and architectural standards • SOA (and its projects) had its own governance frameworks that were aligned with the overarching EA governance

- EA governed service documentation, service identification, and service delivery
- Services were monitored using the orchestration engine
- SOA demands were also governed by EA, similar to any other demands against the architectural standards and strategy
- Every service was governed by technical and business SLAs
- Every business service had an owner

6.7.5 EA Methods and Tools (Transformed)

This section presents SOA's methods and tools integration into EA methods and tools.

EA was integrated with project management and solution development practices, which included SOA projects. Services were identified using a top-down approach of the EA repository deliverables (e.g., processes). EA also informed projects/demands, reviewed them, monitored projects during implementation, and made sure they deliver what they promised.

IT strategist [P-2] emphasised the role of EA and its engagement with the projects lifecycle. Projects requirements and specification were determined using the EA repository (IBM System Architect). Projects used the rich content stored in the EA repository to determine their scope, requirements, specifications, and how they were going to impact the architectures or its content. IT strategist [P-2] mentioned that all the project requirements were delivered out of IBM System Architect. The business requirements specification (BRS) document detailed these requirements. The BRS included business requirements at the service, process, and activity level [P-2]. At the technology level, projects used a document called systems requirement specification (SRS), where use cases were generated to be used for the development. Next, a testing phase was conducted to verify that projects were following the specified specifications and to update the content of the architecture if necessary [P-2]. A senior technical architect [P-7] described the process of EA involvement in any project in the organisation:

> *What happens first is the demand, then, the business case, then, business requirements specifications (BRS). After BRS, comes the system requirements specification (SRS) which translates the*

business requirements into system requirements. SRS is mainly consumed by developers, architects and testing teams.

The repository (IBM System Architect) and the ECV tool were also updated to contain the new architectural elements. In IBM System Architect and ECV tools, business services were documented and linked to business processes in an interactive way. For example, users could navigate to a certain business service, discover other associated services and supported channels, and explore business processes that support that specific business service.

In summary, EA was integrated into project management, which included SOA projects. EA develops business requirements, systems requirements, and reviews business cases, and monitored services/solution developments. The EA repository was service-oriented. Table 6.13 summarises SOA's integration into EA methods and tools in Dubai Customs.

Table 6.13 EA methods and tools

EA methods and tools
• New SOA-related elements and new relationships were created in the used EA tools • New views were created in used EA tools to support services and associated elements • Service identification methods: services were identified using EA products (repository) • EA was integrated with demands/projects, which included SOA projects

6.8 **Summary**

This section summarises the results of the case. In this case, the evolution outcomes at (T4) were understood by retrospectively looking at the architectural interaction (T2-T3) and the architectural conditioning (T1) phases. Figure 6.14 summarise the results of the case study using the theoretical model of this thesis. In this figure, the architectural elaboration (T4) represents the SOA's integration into EA outcomes.

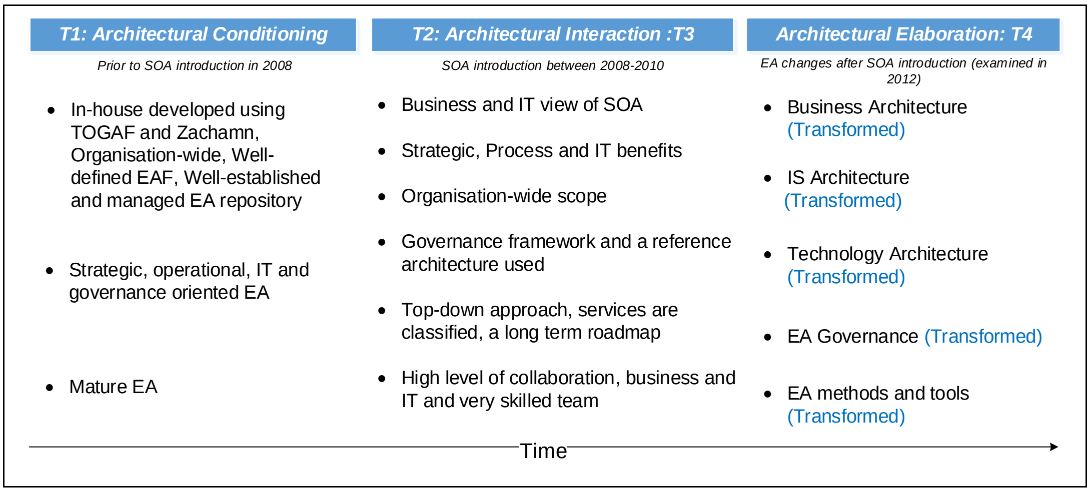


Figure 6.14 The morphogenetic cycle of SOA's integration into Dubai Customs' EA

The rich, mature, well-governed architectural conditioning phase enabled the transformation of the architectural settings due to SOA's introduction. In addition, the actualisation of SOA introduction generative mechanisms contributed to the observed integration outcomes. The introduction of SOA was an organisation-wide project and had a very mature perspective of SOA (enterprise services architecture). It was driven to achieve strategic, process, and IT benefits. It was governed by adhering to a governance framework of eServices delivery, the wider organisational governance practice (COBIT), EA governance, and SOA reference architecture. SOA's introduction was designed according to a long-term roadmap to restructure Dubai Customs' in terms of services by using a top-down approach. Services were clearly defined, classified, and stored in the EA repository. SOA was implemented with strong business and IT support, involved key stakeholders and clients, and was driven by a diverse and skilled team.

As a result, it can be concluded that the architectural conditioning described above enabled the action of SOA's introduction. In addition, introducing SOA in the way described (the actualisation of the generative mechanisms) resulted in SOA's integration into EA on all the identified five architectural levels. Table 6.14 summarises how the outcomes of SOA's integration into EA were generated in the context of the theoretical model.

Table 6.14 Contextualisation of SOA's integration into EA at Dubai Customs

Analytical phases	Generative mechanisms	Actualisation
Architectural conditioning		
	EA framework	In-house developed EA following TOGAF and Zachman
	EA objectives	Strategic-, operational-, IT-, and governance-oriented EA
	EA maturity	Mature EA practices
Architectural interaction		
	View of SOA	Enterprise services architecture view (Business and IT levels)
	SOA scope	Enterprise-wide SOA implementation
	SOA benefits	strategic, business and IT benefits
	SOA governance	SOA was governed against a reference architecture, EA project, and organisational governance
	SOA design	Services were identified top-down, SOA had a well-defined roadmap, services classified and managed (in EA repository)
	Business-IT collaboration	High level of business/IT collaboration, external consultants, skilled and trained SOA team
Architectural elaboration (Outcomes)		
	Business architecture (transformed)	SOA is integrated into the business architecture
	IS architecture (transformed)	SOA is integrated into the IS architecture
	Technology architecture (transformed)	SOA is integrated into the technology architecture
	EA methods and tools (transformed)	SOA methods and tools are integrated into EA methods and tools
	EA Governance (transformed)	SOA governance is integrated into EA governance

Chapter 7: Businesslink Case Study Findings

7.1 Introduction

This chapter introduces the findings of the second case study, which was conducted with the N.S.W. Businesslink Pty. Ltd. (hereafter called Businesslink) in Australia. Businesslink is the only government-owned proprietary company in Australia. It provides shared corporate services to some N.S.W. agencies. It was originally established as a division of the N.S.W. Department of Housing in 2002, and turned into a State Government-owned private company in 2004.

Prior to SOA's introduction in 2010, most of Businesslink's architectural activities were IT-oriented. Businesslink implemented a transformational SOA project called the "next generation service model" to strengthen its presence as a primary supplier of outsourced business services in Australia's public sector. The next generation service model included the transformation of their operating model and the adoption of an innovative service-oriented organisation structure.

This case study was conducted to satisfy the contextualisation stage of the critical realist methodological framework (Danermark et al., 2002). In particular, the theoretical model developed in the previous phases was examined in this case study to explore EA evolution in this specific context. The proposed structures and generative mechanisms, in the developed theoretical model, were further explored to describe EA evolution and explain the observed EA evolution outcomes in Businesslink.

As Chapter 3 presents, the case study design is *a retrospective one*. The three morphogenetic phases (conditioning, interaction, and elaboration) are used to understand how the EA evolution outcomes were generated. The phase of architectural elaboration (evolution outcomes) was identified as the particular point in time that this study would illuminate, and then move backwards through the previous two phases of the model, seeking to uncover the generative mechanisms of the architectural conditioning and interaction phases that have interacted to generate the observed outcomes. In this case, SOA's integration into EA was completed prior to the researcher's engagement with the case, and the EA evolution outcomes were known (based on online empirical evidence) prior to the conduct of this case. The case was selected because Businesslink implemented a business-oriented SOA and received an award for their service-oriented operational model from ICMG, and SOA was integrated into the business architecture (based on online evidence prior to the conduct of the study).

This chapter progresses as follows. First, Section 7.2 describes the data collection and analysis. Section 7.3 describes the background of the organisation, and Section 7.4 shows its organisational structure. Section 7.5 examines the architectural conditioning phase and looks at EA framework, EA objectives, and EA maturity. Section 7.6 examines the architectural interaction (SOA's introduction)—in this case, the implementation of the next generation service model. Section 7.7 represents the architectural elaboration (EA evolution outcomes) that resulted due to the implementation of the next generation service model. Finally, Section 7.8 summarises the findings of this case.

7.2 **Data Collection and Analysis**

In this case study, eleven participants were interviewed at Businesslink's premises in N.S.W, Australia (see Table 7.1). Interviews were conducted following the case study protocol (Appendix B). Each interview lasted between 45 minutes and 1 hour. All the participants were involved in the next generation service model and/or EA. Each interview was recorded, transcribed, and analysed using NVivo 9, following the thematic analysis technique (see Chapter 3).

Table 7.1 Businesslink case participants

Participants	Positions	Years in org.	Years of Experience	Background
P-1	CEO of Businesslink	8	15+	Business and management
P-2	Manager of Service Design & Architecture	8	10+	Business/IT

P-3	Service and Solution Design Lead	5	10+	Business/IT
P-4	General Manager of Business Solutions Development	8	20+	Business/IT
P-5	General Manager, Program Management Office	2	20+	Business/IT
P-6	Enterprise Architect & Project Manager	3	15+	Business
P-7	End User Solutions Manager	2	10+	Technology
P-8, P-9	Lead Enterprise Architect & Enterprise Architect	1,1	15+, 20+	Business/ Technology
P-10	Planning & Innovation Manager	7	10+	Business/ Technology
P-11	Service Design & Architecture	3	10+	Technology

In order to achieve data triangulation, many documents related to EA and SOA were obtained from the department or were discovered on the Internet (see Table 7.2). All of the sources of evidence (interviews and documents) were imported to NVivo 9 and analysed.

Table 7.2 Businesslink obtained documents

ID	Title	Source	Description
D-1	Businesslink - Next Generation Services Model	Online	Presentation of the Next Generation service model presented by the CEO at the IPAA NSW 2012 State Conference, 2012
D-2	Businesslink - ICT Strategic Plan	Internal	The ICT Strategy sets the direction for the use of information and communications technology (ICT) in Businesslink over the period 2011/12 through 2014/15 (3 years)
D-3	Businesslink - Annual Report 2011-12	Online	A report on Businesslink's activities from 1 July 2011 to 30 June 2012
D-4	Enterprise Architecture Capability Uplift & STM Work Plan	Internal	A presentation describing the work plan to improve EA capabilities through strategic transition management initiative
D-5	Strategic Transition Management	Internal	A presentation describing the strategic transition management initiative undertake to increase the readiness and

ID	Title	Source	Description
	Enterprise Architecture Work Stream (v5)		capacity to proactively respond to changes
D-6	Operating Model : Enterprise Architecture	Internal	A document that describes the new service based operating model and their new EA
D-7	Commercial Focus For Sustainable Community Outcomes	Internal	A periodical document that outlines Businesslink operations for clients and stakeholders
D-8	Designing the Next Generation in Shared Services	Internal	A report that outlines and describes the Businesslink transformation program activities and outcomes
D-9	Enterprise Architecture - Technology Architecture Taxonomy v0.4	Internal	Document that describes EA prior to 2010
D-10	Consolidated Technical Artefact Register	Internal	Document that describes the architectural domains to 2010
D-11	Service Catalogue	Internal	A document that has information about Businesslink's services. It is designed to be a single source of consistent information about Businesslink's services
D-12	Service Design Package	Internal	A document that is used to design and develop services. It is shared and developed by many stakeholders who are involved in services design and development, and reviewed by services governance bodies
D-13	Service Ownership Governance Policy	Internal	The "service ownership" governance policy states how Businesslink monitors the "end-to-end" process of delivering and continuously improving services they provide to their clients
D-14	Leading the Way forward	Internal	A presentation that was presented to staff at the beginning of the transformation to explain the transformation initiative and its impact
D-15	Detailed Design of FMS	Internal	Document that describes the design of one internal system and its components and infrastructure
D-	2011/12	Online	A document that outlines the business

ID	Title	Source	Description
16	Statement of Business Intent		intent and objectives in 2011/2012, and what measures are taken to assess the achievements of the intent
D-17	Operating model and the role of new divisions	Internal	A document that describes the new operating model and the new structure of the organisation and its divisions
D-18	Single ERP tabulation of Application function	Internal	A document describes the multiple instances of exiting ERP to move to a single standardised ERP

In this case, the morphogenetic cycle of SOA's integration into EA was determined based on the stability-change-stability approach discussed in Chapter 3. The change is limited to SOA introduction as a trigger of EA evolution. There could be other aspects that cause changes to EA, but they are out of the scope of this study, which focuses only on SOA and EA integration outcomes. The Businesslink's SOA integration into EA morphogenetic cycle is shown in Figure 7.1.

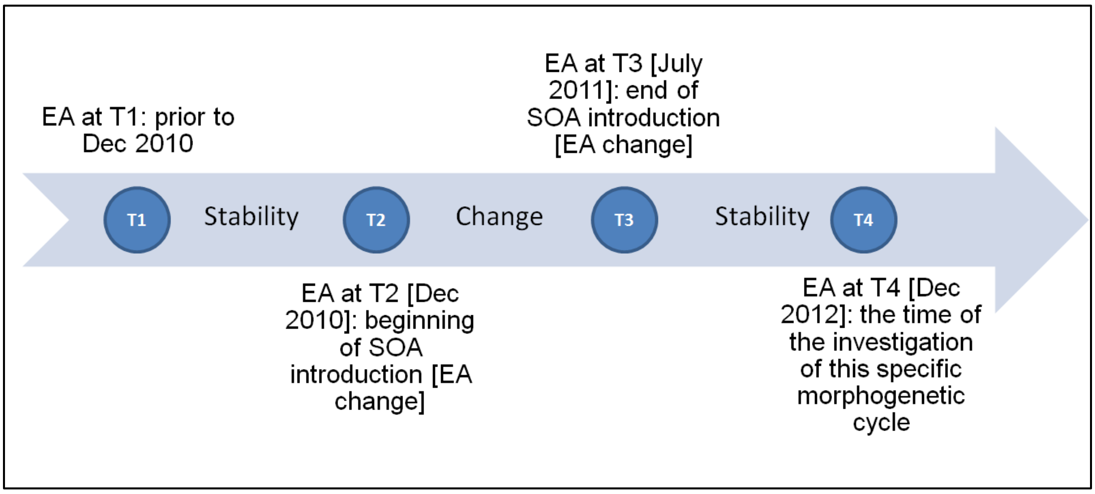


Figure 7.1 The morphogenetic cycle of SOA's integration into EA at Businesslink

The oldest obtained documentation of EA at Businesslink was from 2008. It describes EA (IT-oriented architecture) at that stage. That period until SOA's introduction is considered the architectural conditioning (T1-T2) of the new morphogenetic cycle (SOA's integration into EA). The change period (architectural interaction) began when SOA was introduced in 2010 and finished in mid 2011. This study was conducted in June 2012 during EA's stability period after SOA's introduction.

In order to understand SOA's integration into EA outcomes, the event (SOA's introduction) and the EA prior to SOA's introduction were studied retrospectively. The retrospective analysis was achieved through intensive interviews with people involved in EA and SOA, and was supported by analysing relevant obtained documents (see Figure 7.2).

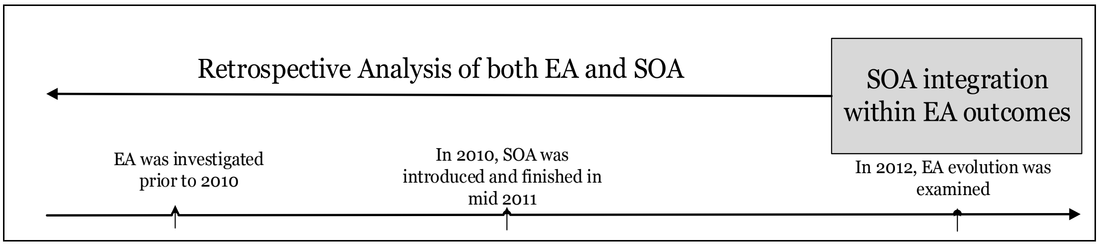


Figure 7.2 Employed retrospective analysis at Businesslink

The collected data was analysed in a manner that was informed by the procedures used to analyse the Dubai Customs case. The interviews were first transcribed. Then, all the interviews and obtained documents were imported into NVivo to prepare them for analysis using the previous case's final codebook. The analysis used the thematic analysis technique (and, more specifically, a deductive approach using only this thesis's theoretical model as a lens). Figure 7.3 shows a snapshot of the used codebook and the analysed text.

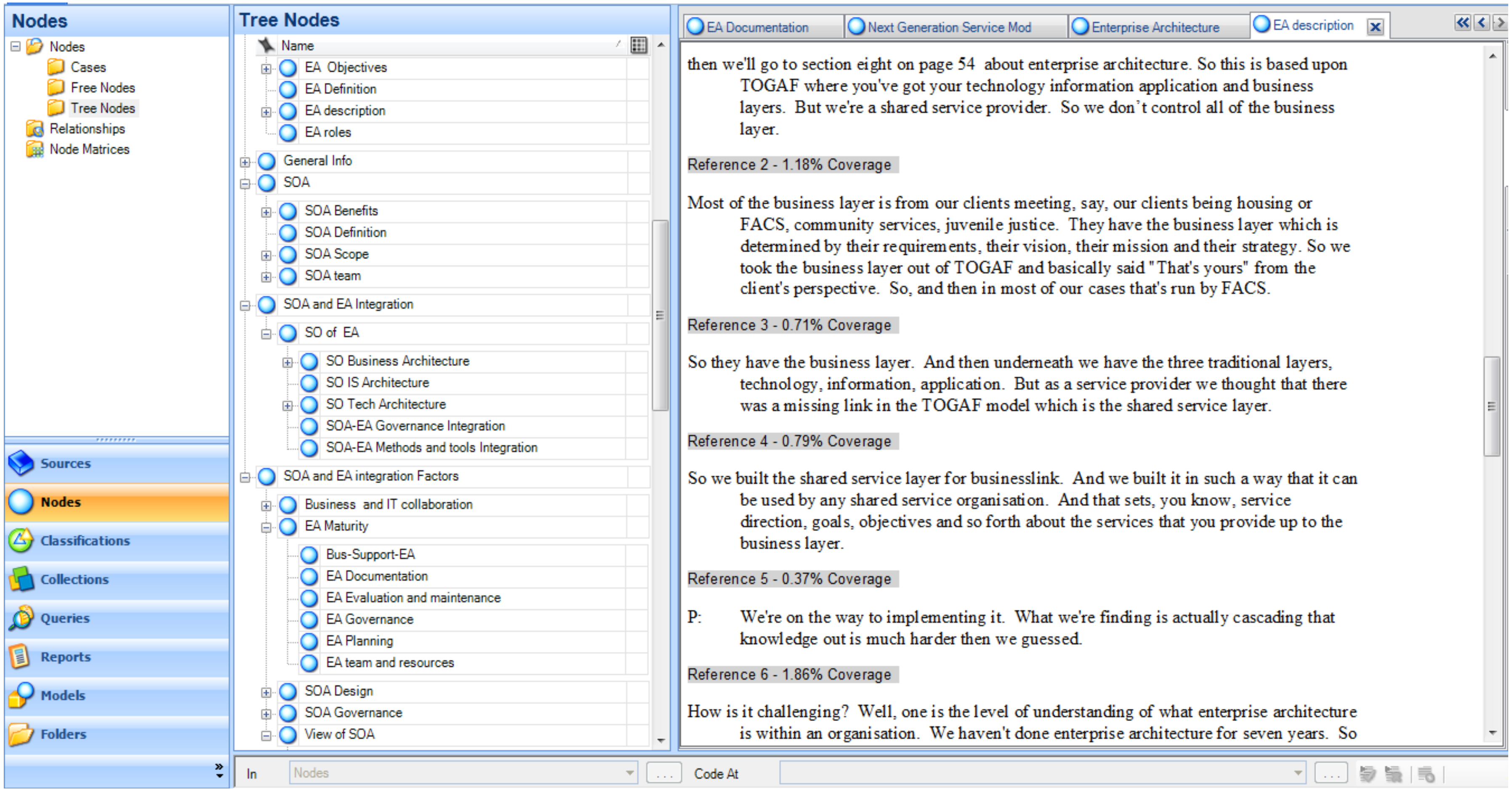


Figure 7.3 Snapshot of the Businesslink Case Analysis using Nvivo

7.3 Case Background

Businesslink is the only government-owned proprietary company in Australia. It provides shared corporate services to some N.S.W. agencies such as the Department of Ageing, Disability, and Home Care (DADHC), and the N.S.W. Department of Community Services. Its goals were to provide outsourced services on behalf of its clients directly to the community, improve corporate service delivery, enable agencies to focus on their core business, realise the benefits of technology, and reduce costs (2004-05 Annual Report).

Businesslink now provides outsourced commercial services to government agencies. Businesslink's mission is "to deliver high quality, secure business services to our clients at the lowest cost in the Australian market". It has a vision *"to be recognised as a centre of excellence for outsourced business services"* [D-15].

Businesslink provides a broad range of core, transactional, and value-added services in the N.S.W. public sector. Businesslink is a specialised provider of services in areas such as human resources, information technology, business services, projects, workforce, and finance [D-3, D-11]. Businesslink provides services to clients in five different clusters in the N.S.W. government, and there are at least ten departments in each cluster. Businesslink clients include [D-3]:

- N.S.W. Department of Family and Community Services
- Aboriginal Affairs – N.S.W. Department of Education and Community
- Early Childhood Education and Care Directorate – N.S.W. Department of Education and Community
- Juvenile Justice – N.S.W. Department of Attorney General and Justice
- Land & Housing Corporation –N.S.W. Department of Finance and Services, and
- Non-Government Organisations (NGOs).

Businesslink provides outsourced services on behalf of its clients directly to the community. These services include call centres, systems and processing support for N.S.W. seniors' cards and criminal checks. Businesslink engages with its clients (other government agencies) to assist them to achieve better community outcomes. It also helps them by maximising value of service delivery with economies of scale and end-to-end service delivery. In turn, Businesslink's clients receive important benefits such as low operation costs and reduced risk. Businesslink also enables improved collaboration across agencies to provide a unified view of clients, which leads to the easier, more-effective, and better-integrated case management of individuals consuming services from multiple agencies [D-15].

Businesslink endeavours to be a reliable and consistent one-stop shop that enables its clients to focus on their core business. Document [D-7] states that:

> *we are moving to a new way of doing business, which will reduce costs and make it easier for customers to access services 24/7. Like the internet banking model, it will be more user friendly and easier for people to interact with us, whenever they want.*

In order to deliver high-quality and low-cost services, Businesslink adopts the most innovative technology and components available in the market. It also buys or builds business solutions to improve the consolidation and standardisation of its provided services [D-7]. Businesslink sources, designs, and builds integrated processes and IT systems that aim to provide high-quality services [D-15].

Businesslink endeavours to generate more value for their clients to enable them to achieve their core mission and deliver their core services at lower costs, and to provide them with opportunities to improve the delivery of frontline services to community members [D-3]. Businesslink's clients are benefited in the form of consistent prices and improved services. In 2010/2011, Businesslink returned around $10 million in benefits to its clients and the people of NSW [D-11].

Businesslink's current (at the time of writing) operations are diverse. For example, Businesslink processes around 512,000 transactions per annum for more than 20,000 of its clients' employees. It receives 7,800 helpdesk calls every week and 3,000 IT change requests per year. It currently manages 3,176 fleet vehicles and 2,200 funding applications [D-15]. Businesslink has invested in scalable infrastructure that can be scaled up by 300% to ensure that it can meet anticipated growth and also to cater for peak demands by providing additional capacity at a much lower marginal cost [D-15].

7.4 Organisational Structure

Businesslink was transformed and restructured when it implemented a service-oriented operating model. The new organisational structure focuses on client engagement, service development, and service delivery. The old organisational structure and the organisation's core business activities and processes have been re-engineered in close engagement with client agencies in order to meet changing client demands and improve business agility [D-3]. According to the CEO, the new service-oriented operating model "represents a fundamental shift from the provision of functions through systems, technology and applications to a business organised around service and process excellence" [D-1].

The new operating model (see Figure 7.4) was organised around services, and Businesslink was structured into six divisions: client engagement, service development, service delivery, people and culture, finance, and corporate strategy and performance.

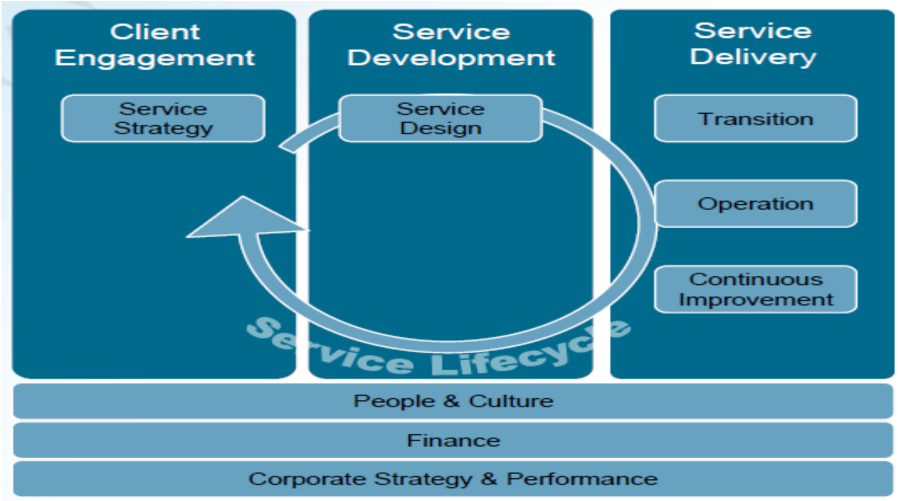


Figure 7.4 Businesslink's operating model

First, the client engagement division engages with clients to understand their business and their needs. The division also develops requirements for new and enhanced services, which include high-level service design. They also conduct high-level assessments of viability and financial costs. The client engagement division manages several responsibilities, such as service standards management, client satisfaction management, account management, requirements analysis and definition, service strategy, service enhancement, and marketing and advertising. The second division is the service development division. Service development teams are responsible for service design, architecture, strategic sourcing (e.g., buying, building, or outsourcing services; testing services, designing solutions, managing project portfolios, and handling supplier relationships). The third division is the service delivery division. They have different teams responsible for service delivery management, service assurance, service knowledge and information, analysis and reporting, service delivery capabilities, and service solutions. The other three divisions are enabler divisions: people and culture, finance, and corporate strategy and performance. These divisions have teams who develop Businesslink's strategy, who recruit, develop, and retain the best talent for Businesslink's services workforce, and who manage Businesslink's finances, risk exposure, corporate communication, and compliance [D-17].

Businesslink also adopts a hybrid workforce of public servants and contractors to meet the demand for services from its clients. Businesslink manages its contract workforce by employing firm policies, such as not engaging contractors for more than three years [D-3]. It employs (or has on contract) approximately 900 people. Around 60 percent of their workforce is permanent; the other 40 percent is contingent labour (either temporary public servants or contractors). The IT workforce is about 25 to 30 percent of the total workforce [P-1].

The previous sections outline the case's background, present the participants' information, and provide a list of the obtained documents. The following sections examine the findings according to the theoretical model of this thesis (see Figure 7.5). The findings of SOA's integration into EA are organised using the theoretical model's three analytical phases: architectural conditioning, architectural interaction, and architectural elaboration.

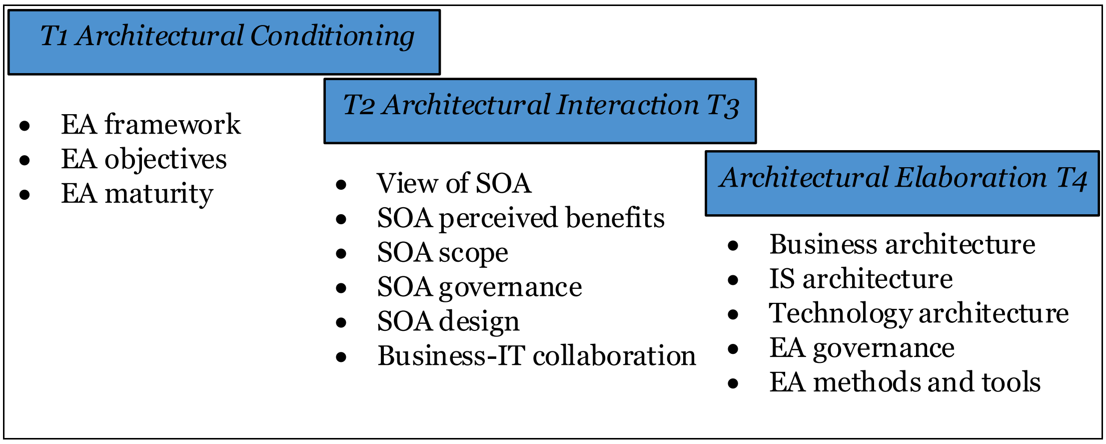


Figure 7.5 The thesis's theoretical model

7.5 Architectural Conditioning

This section examines the architectural conditioning phase prior to architectural interaction (SOA's introduction). It particularly examines the three conditional generative mechanisms conditions (EA framework, objectives, and maturity). Due to the limited documented information about EA priori to SOA's introduction in Businesslink, the obtained evidence was minimal compared to the previous case.

7.5.1 EA Framework

Prior to SOA's introduction, most EA activities in Businesslink were IT oriented. EA was positioned in the IT division. It was called EA although it was predominantly about IT architecture (described in D-9). The architectural work concentrated on the application, data, and infrastructure layers rather than a global organisational perspective. The manager of service design and architecture [P-2] stated:

> *"There was a significant focus on infrastructure and solutions rather than enterprise global architecture…we documented a whole lot of infrastructure standards and that type of thing" [P-2].*

The architecture blueprint was document based. It described two essential aspects of a given technology area: its architecture and its technology standard(s). The focal points were applications, integration between systems, data, platforms, and security, using an unorganised documentation approach in Word and Visio documents. The purpose of the blueprint was to document the current and target state of the IT-oriented architecture and technology standards. The architecture blueprints were grouped together in domains and technology areas. An architecture domain was the primary category for grouping related technologies. Figure 7.6 shows the top-level architectural domains [D-9].

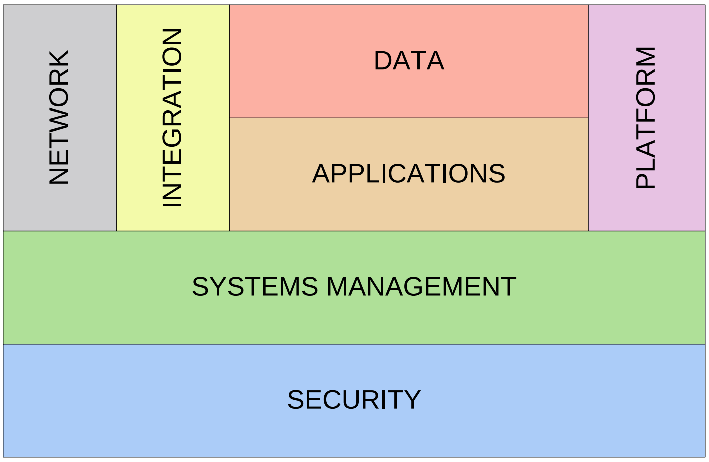


Figure 7.6 Architectural domains [D-9]

These domains defined the supporting technologies and standards [D-10]:

- Network—distributed applications requiring data access and interoperability in a network environment
- Platform—infrastructure platforms and supporting services
- Data—data and information management platforms and services
- Application—business and support applications

- Integration—access and exchange of information between applications and information repositories
- Systems management—the management, monitoring, and support of infrastructure and business applications, and
- Security—the protection of information from a wide range of threats in order to ensure business continuity, compliance, and privacy.

7.5.2 EA Objectives

Prior to SOA's introduction, EA objectives were centred on solutions and were governance oriented. Most of the efforts were concentrated on standards related to solution development to improve aspects such as consistency and compliance. Another side of the architectural objectives was the building blocks' development. These building blocks were documented to accelerate the development of solutions by reusing existing building blocks. To summarise, the objectives were:

> *to develop standards that foster consistency, compliance, efficiency and cohesion in ICT solution design. The goal is to develop a cookie cutter approach to ICT solution architecture where solutions are assembled together from the various building block standards. This not only guarantees conformity to standards, but greatly reduces the time to market for solutions* [D-9].

Some participants reported some issues due to the lack of an organisation-wide architecture established and practised at the organisation. Some of these drawbacks were:

- An ambiguity of vision and unclear roadmaps about how they are planning to move ahead [P-2, P-11]
- Inefficiency with respect to money spending and implementation of the right projects [P-2,P-11]
- A diverse range of systems and applications [P-6];
- A limited understanding of how to improve leverage of the common information that should be common but is not [P-6], and
- Inconsistency [P-7].

7.5.3 EA Maturity

The EA maturity assessment, prior to SOA's introduction, was measured based on: (1) the obtained documents about EA prior to SOA's introduction, (2) interviews, and (3) the EA maturity assessment survey questionnaire, which is shown in Appendix B in the case study protocol.

EA maturity, prior to SOA's introduction, was at level one (informal program) (see Table 7.3). The details of some of the maturity dimension are presented in (Sections 7.5.31. to 7.5.3.). Due to the low level of maturity and the lack of formal EA practices, some of these dimensions were not officially used in Businesslink.

Table 7.3 EA at level one of maturity (informal program)

Level 1: informal program
• Documentation processes are ad hoc and informal • EA activities are informal and unstructured • The need for organised committees to define the architectural standards and processes has been identified • The organisation has identified a need for capable EA team • Evaluation processes are ad-hoc and informal

7.5.3.1 EA Documentation

The obtained documents were IT oriented. There was no meta-model nor a unified repository to store and manage EA artefacts. There was some fragmented documentation of IT architecture artefacts in the form of technology roadmaps and building blocks. These artefacts were stored in Word, Excel, and Visio documents that were distributed across the organisation. Participant [P-2] reported *"there was a significant focus on infrastructure and solutions rather than enterprise global architecture"* and participant [P-6] stated "*The documentation we have, this was really, really high level. It only goes down to the two pictures, if you like".*

The architectural building blocks documentation was called the architecture building block (ABB) or solution building block (SBB). It was used to increase reuse and accelerate solutions delivery. It was:

> *a documented solution or technique to a common or recurring problem. Building blocks are developed by subject matter experts*

> *through the application of industry and organisational best practice, as well as lessons learned… it offers a succinct solution summary outside of formal design documentation* [D-9].

The ABB was the smallest technical solution standard. ABBs were aggregated to form SBBs. Participant [P-10] noted: "*The architects go and write the relevant architecture building blocks that sit at the side that we can reuse. So when that came through, we've now used the IPsec design three times. So it's building on the reuse*" [P-10].

Table 7.4 shows an example of the workstation platform artefacts that belonged to the platform domain. It had a roadmap for desktops. It also had an example of the related ABBs and SBBs [D-10].

Table 7.4 Workstation platform artefacts [D-10]

Classification	Artefacts	Stage	Last update
Architecture roadmap	Roadmap - desktop	Stage 1: Work in Progress	Jun-09
Architecture building blocks	A.2.3.2.1 - Managed laptop hardware	Stage 1: Work in Progress	Jan-08
	A.2.3.2.2 - Shared laptop hardware	Stage 1: Work in Progress	Jan-08
	A.2.3.2.4 - Novel Zenworks client	Stage 1: Work in Progress	Jan-08
	A.2.3.2.5 - Symantec Antivirus	Stage 1: Work in Progress	Jan-08
	A.2.3.2.6 - Managed desktop hardware	Stage 1: Work in Progress	Jul-09
	A.2.3.2.7 - Workstation operating system	Stage 1: Work in Progress	Jul-09
	A.2.3.2.8 - iPrint client	Stage 1: Work in Progress	Jul-09
	A.2.3.2.10 - Web browser	Stage 3: ADS Approved/Final	Jul-09
Solution Building Blocks	S.2.3.2.1 - Workstation SOE	Stage 1: Work in Progress	Jul-09

7.5.3.2 EA Planning

The most obvious form of architecture planning was the development of technology roadmaps. Table 7.5 shows the template of the roadmap as provided in [D-9]. It provides information about the current, tactical, and strategic architectural standards, designs, and methodologies used in a particular technology area. The documentation and planning according to [P-3] was project oriented and affected by the silo structure:

> *I think previously it was more projects-centric and more silo-centric rather than having more a service-oriented approach to it. So if you look at most of our design documents are very, you know, IT and just focused on specific tasks rather than having a broader aspect of end-to-end delivery approach.*

Table 7.5 Architecture forecast template [D-9]

Current	Tactical (2 years)	Strategic (2+ years)
Summarise the current architectural standards, designs, and methodologies employed in this technology area. All solution designs are expected to comply to these standards unless otherwise instructed	Summarise the architectural standards, designs, and methodologies that this technology area is expected to employ in two years	Summarise the architectural standards, designs, and methodologies that this technology area is expected to employ beyond two years

7.5.3.3 EA Governance

Participant [P-2] stated that some governance practices were established at the technical level: *"There are governance arrangements already in place at that infrastructure layer"* [P-2].

Further, the process for review and approval of the IT-oriented architecture artefacts, including building blocks and architecture blueprints, was described in [D-9]. There were some defined governance committees. A high-level process of artefact change was also described. With regard to the evaluation dimension, EA artefact definitions were stored in an Excel file with the name, status, and owner of the artefact that was stored in other documents (Word, Visio, or other similar formats). The tracking of changes was manually executed and some artefacts in that repository were outdated.

7.5.4 Summary

Prior to SOA's introduction, the EA framework was IT-oriented and developed in-house. It had seven IT domains (applications, network, integration, systems management, data, platform, and security). EA maturity was very low prior to SOA's introduction.

The EA framework focused on documenting the IT domains using fragmented artefacts such as Excel, Word, and Visio documents. There was no well-defined EA methodology. The documentation was informal and inconsistent. According to some participants, it was project-oriented documentation. It also focused only on high-level technology roadmaps and technology standards. There was a high-level governance approach around the change of the artefacts that were stored in Excel file. However, a copy of that repository showed that some artefacts had not been updated for a long time. Information about some artefacts was also missing.

The focus on applications and technology artefacts could be due to the fact that the organisation was an IT-oriented shared service provider. Participant P-8 stated: *"Businesslink is almost 90%, no, it wasn't, 86% of its assets are ICT assets, so it's first and foremost an IT-enabled organisation"*.

However, the IT-oriented organisation has extended its presence to be a service-oriented enterprise. For example, participant [P-2] reported that the organisation wanted to move beyond being an IT- and infrastructure-based organisation to a service-oriented organisation*: "[We're] trying to be more of a service-oriented organisation… and looking at business processes, so higher up the value chain than the traditional we're just hosting your IT infrastructure"*.

This section discusses the architectural conditioning phase. Section 7.6 examines SOA's introduction (next generation service model).

7.6 Architectural Interaction: SOA's Introduction

This section presents the findings relevant to the second analytical phase of the theoretical model, the introduction of SOA into the organisation. It provides an overview of SOA's introduction and examines the findings using the six action-formation generative mechanisms (view of SOA, SOA perceived benefits, SOA scope, SOA governance, SOA design, and business and IT collaboration).

7.6.1 Overview of SOA's Introduction

In late 2010, Businesslink introduced a transformational project called the next generation service model (see Figure 7.7). Businesslink wanted to strengthen its presence as a primary supplier of outsourced business services in the Australian public sector, which it has accomplished by changing its operating model through implementing the next generation service model [D-3].

The transformation was driven by many drivers. One of the major drivers was the whole-of-government corporate and shared services reform program. It was adopted to restructure the organisation, its information flows, and its business processes from a functional operating model to a service-based model organised around client engagement and improved service delivery [D-8].

During the transformation, many implementations were performed. For example, a new service portal for non-government organisations (NGOs) was provided through a purchased software-as-a-service (SaaS). The deployment of my virtual office (MVO) was also completed for most of the agencies. MVO provides a simple model for virtually deploying and using applications normally delivered on traditional desktops. An invoice management solution (IMS) was also provided to two clients. IMS is an automated system for processing invoices for clients [D-3]. Integration platforms were used, such as SAP process integration and web services for point-to-point integration.

The transformation was also driven by technology drivers. Mobility was a driver to improve efficiency for a mobile workforce. For example, it enabled field workers to use SAP mobility solutions delivered on mobile devices. Cloud computing, particularly private clouds, was another driver to reduce costs and standardise technology solutions. Self-service was another driver to meet the greater expectation of citizens and to enable them to access services online. It also enabled agencies to improve budget utilisation by widely adopting self-service platforms.

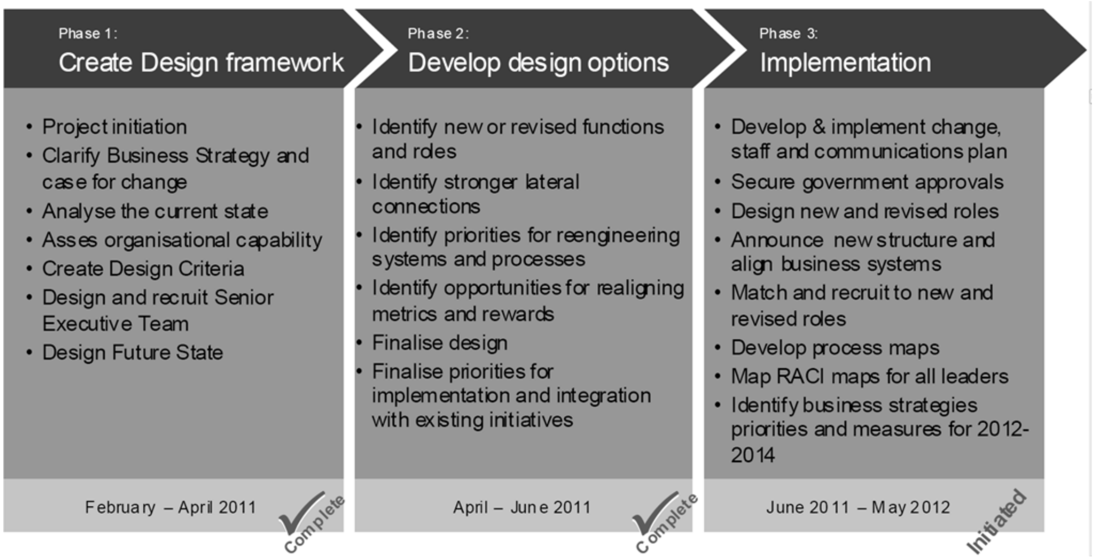


Figure 7.7 Next-generation service model initiative [D-8]

The next generation service model was implemented to improve efficiency, standardise and simplify processes, and reduce the cost of business services delivery to the public sector [D-8]. The organisation was re-organised into a service-centric organisation to improve clients' access to services, increase the use of mobile computing, increase one-step self service access, and make doing business easier via a range of client retail, e-commerce, and mobility access [D-3].

The organisation was operating in silos. It was structured as a traditional corporation. It had a finance division, HR division, IT division, and a division that managed everything else. The manager of service design and architecture [P-2] stated: *"We're trying to organise along service lines, so our history, we're a shared corporate services provider and now we're trying to position ourselves more as provider of outsourced business services"* [P-2].

The CEO reported that the unsuccessful implementation of end-to-end processes while operating on the old model was a driver to move to a service-oriented operating model. He noted: *"In that model we had tried unsuccessfully to implement end-to-end processes like hire to retire, purchase to pay"* [P-1].

Therefore, the organisation was structured in terms of the services it provided (see Figure 7.8). Each service had an owner and many components. These components are people, processes, and technology [D-8, D-12].

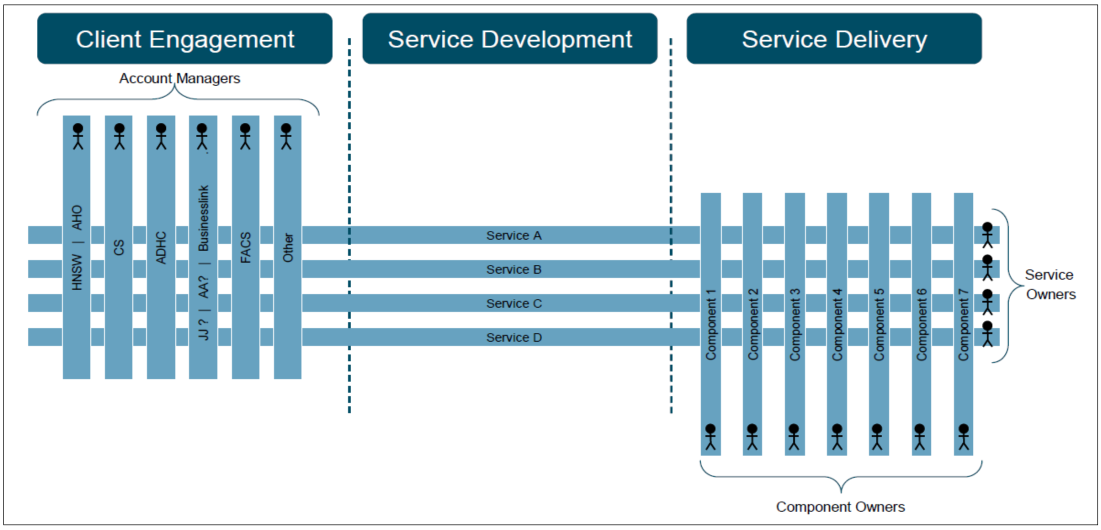


Figure 7.8 End-to-end service delivery [D-8]

7.6.2 View of SOA

In this case study, SOA's introduction was business oriented fitting under the enterprise services architecture view presented in Chapter 2. SOA was driven with a high business focus to move the organisation toward a service-oriented organisation. Such a perspective of SOA fits under the enterprise service architecture view of SOA that was discussed in Chapter 2 because of the following two points. First, the organisation was restructured and some of its business processes were redesigned to improve service delivery. Second, services were also defined in terms of business needs in advance of their use in processes. However, there was less emphasis on the technology levels of the organisation throughout this initiative. The objective of the initiative was to transform and re-organise the organisation in terms of the services it provided.

The service *"is the offering and/or consumption of a type of transaction which adds value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks"* [D-11], and a service *"is comprised of a combination of components—people, processes and technology"* [D-12]. Participant [P-6] offered some examples of their services:

> *So the services are provided by us, consumed by their business. So that could be payroll. It could be a whole financial, from accounts payable, accounts receivable or it could be just desktop. So you've got your PC with Microsoft applications, email, storage, network communications and so forth as a service that we offer* [P-6].

7.6.3 SOA Perceived Benefits

This section examines SOA perceived benefits that are associated with SOA introduction. The main benefit of the next-generation service model was the delivery of end-to-end services to its clients. It reduced service delivery costs and risks. It also added flexibility to the organisation's service delivery, which was constrained by the old silo-based operating model. The partial adoption of cloud computing (SaaS) to deliver some services also provided quicker times to market, quicker responses to changes, and improved agility. Examples of the supporting quotes of reported benefits are presented in Table 7.6.

Table 7.6 Quotes regarding the next generation service model benefits

Reported benefits	Quotes
Reduce costs	"Allows for the provision of significant benefits to clients through lower operating costs" [D-1] "One major component is the move toward cloud technologies, as they offer the opportunity to lower costs and enhance agencies ability to respond quickly to emerging needs" [D-3]
Agility	"So that way we can decouple down the line and give us the flexibility, better cohesion within the service and more service-centric rather than a specific technology" [P-3] "It would give us a whole lot more flexibility. Less vendor lock-in, ability to pick and choose" [P-2]
Increased customer satisfaction	"Better value for money has accompanied and accentuated an improved service experience for our clients" [D-3]
Communication	"It gives the business a degree of transparency that it typically doesn't have around the efficiency, the effectiveness of its processes; who is doing what" [P-1]
Process improvement	"To build continuous improvement...through standardisation, streamlining, automation.... efficiencies and economies of scale through end to end process improvement" [D-3]

In summary, the benefits that participants reported, or that were in the obtained documents, are flagged using the adopted SOA benefits classification in Table 7.7. The reported benefits belong mostly to the strategy and process categories and less to the IT category.

Table 7.7 SOA perceived benefits

SOA benefits layers	SOA benefits	Reported
IT level	Reuse	
	Facilitation of software development	
	Improved IT integration	
	Reduce complexity	
	Improved project management	
	Better assets utilisation	
	Reduce maintenance costs	x
Process (operational) level	Increased availability of information	
	Reduce maintenance costs	x
	Increased customer satisfaction	x
	Business process improvement	x
Strategy level	Agility	x
	Business-IT alignment	x
	Enablement of new functionality	
	Improve communication	x
	Reduce time to market	x

7.6.4 SOA Scope

The next generation service model was an organisation-wide transformational initiative with a high emphasis on the business side of the organisation. In particular, the organisation, its business processes, and its information flows were structured in terms of services to support the design, development, and delivery of services to clients. According to the CEO [P-1] *"We are well down the path. So we set ourselves a three year journey to restructure and shift the organisation. That was two years ago pretty much next week"* [P-1].

It focused on a top-down design of the business side of the organisation in terms of services. Some other projects were concerned with standardising, streamlining, and automating efforts. These projects include service management programs, single ERP programs/projects, and core infrastructure projects. Additionally, end-to-end client services were continuously improved and developed. During the transformation, many service-oriented projects were implemented (e.g., a service portal using cloud computing (SaaS) was adopted to provide services to NGOs:

> *The new Service Portal for non-government organisations (NGOs) is provided through a purchased software-as-a-service (SaaS). By using this software there is the added benefit of consistent functionality and look and feel as the user base of the product expands. It is easy and quick to deploy, with a quick roll out to NGO end users and agility to introduce system changes quickly* [D-3].

7.6.5 SOA Governance

Businesslink implemented a project-based governance framework around the next generation service model implementation [D-3]. It used no specific SOA reference architecture. The EA governance practices were still emerging and SOA governance was not aligned with them.

The next generation service model was governed using the traditional organisational project-based governance. Key stakeholders were involved and supported the transformation. The service transition group was in charge of the implementation [D-8]. Different governance committees were established. For example, a new service and services enhancement committee, and a governance committee responsible for new services development or existing services improvements were established. These committees worked as controlling gates along the services lifecycle:

> *We have a committee called new service and enhancement committee. This committee is like the governance committee for the services and there are some key gates along which we track them* [P-5].

A design authority, which took care of the design aspects of the service (e.g., processes, applications, and infrastructure) was established [D-8]. A governance policy was developed to assign responsibilities and provide authority to the positions that were accountable for delivering services and service components [D-13]. Service owner responsibilities were also integrated into the new operating model and organisational structure. The model redefined the role of service owners to meet the requirements of outsourced business services providers [D-1].

Further, service monitoring for compliance to industry standards such as the Service Delivery Management Standards (ISO 20000) was established to identify requirements for service providers to plan, establish, implement, operate, monitor, review, maintain, and improve service management systems [D-16, P-7].

In summary, the governance aspects were business oriented. The organisation was governed using the existing organisational governance practices. The service-oriented transformation was managed and supported by key stakeholders. Multiple governance committees stretching over the service lifecycle were established to manage and monitor services.

7.6.6 SOA Design

This section addresses the SOA design aspect. Services were identified based on client requirements:

> *Service design is carried out using a collaborative approach with our clients. The focus is on the benefits to be achieved and a clear understanding of the change that is required to ensure that the new end-to-end service can be effectively utilised* [D-3].

Each service had its own smaller components: processes, people, and technology. Services were grouped in five categories. They were grouped into IT services, HR services, finance services, business services, and client-managed services [D-13, P-3]. The CEO described the service lifecycle [P-1]. The lifecycle starts by obtaining the requirement from the client (a concept), then:

> *detailed design, source, build, test, implement, run and continuously improve. So it's an ongoing cycle rather than just a one-off event. [P-1].*

The design also covered the solutions development if the service needed application and technology components. The development also included principals that determined how these components were brought together to enable flexibility, shorter time of delivery, and reduced costs:

> *Service solution development follows the principle of "buying" and integrating externally provided service components, "reuse" of existing standardised components, and only building customised "in-house" solutions if there is no viable alternative. This not only drives to a lowest cost solution, but enables innovative and flexible solutions to be delivered in a shorter timeframe* [D-3].

The organisation also developed a service catalogue (static PDF file) that stored information about provided services. It was developed to inform clients of Businesslink's services, including business outcomes, costs, obligations, and prerequisites across the service lifecycle [D-8]. Further, each service had a service design package. It built on the service level requirements (SLR). It also specified client's requirements and defined how the requirements were essentially satisfied from a technical and business point of view, and provided the service roadmap. The service roadmap outlined the current state and planned enhancements or changes to a service over a period of time [D-12]. The CEO [P-1] described the service roadmap:

> *[It is] rather than a 'here today' view… what is the forward profile of changes, enhancements or development…. So across every service line we have what's the future plans for development. [P-1].*

7.6.7 Business and IT Collaboration

The next generation service model was a highly business-driven and business-oriented project. It was supported by the CEO and key executives. More than 100 Businesslink leaders and staff at all levels, and the Businesslink Chair and client agency executives, were consulted during the design and implementation of the new next-generation service model [D-8]. A linkage was established:

> *from the executive level to the operational level of the organisation with a strong focus on planning and monitoring investment directed at implementing the new operational model [D-2].*

Client engagement was a core component of the process of services design and development [D-3]. According to participant [P-5]:

> *What's changed here is that we work much more together in terms of delivering a common business outcome… you involve the people in*

> *the run, that service delivery right up front so you define, design your system for how it is going to be used not about how you're going to develop it.*

7.6.8 Summary

In summary, SOA's introduction (the next-generation service model) was examined in terms of the six action-formation generative mechanisms presented in the theoretical model. First, SOA's introduction was driven by the business with a strong business focus. It focused on the redesign of the organisation in terms of services. It was also associated with small projects to implement individual services on the technology level, but was not as extensive as the re-design of the organisation on the business level. Second, SOA's introduction was associated with perceived benefits, mainly at the strategy and process levels, such as agility, cost reduction, and customer satisfaction improvement. Third, SOA's introduction was organisation-wide. Yet, it emphasised the business aspects and business restructuring in terms of services. There were some implementations on the technology level, such as cloud-based service delivery, but these were not as extensive as the transformation of the business architecture. Fourth, the governance of SOA's introduction was business oriented and based on existing organisational (project-based) governance practices. Multiple governance committees were established alongside the service lifecycle. Finally, SOA's introduction was initiated by the business and required the involvement of IT stakeholders and key clients. It was supported by the top management of the organisation to shift the focus from technology to services delivery.

The next section examines the architectural elaboration that resulted from introducing SOA.

7.7 Architectural Elaboration: Reproduction or Transformation

This section addresses the architectural elaboration (reproduction or transformation) of the exiting EA practices due to the SOA's introduction. Before discussing the architectural elaborations, it provides an overview of EA improvements that happened during SOA's introduction.

During the next generation service model implementation, Businesslink expanded its pre-exiting EA by adopting TOGAF, which was customised to meet Businesslink's needs. EA was defined as *"an essential strategic activity required for the successful planning and delivery of Businesslink's current and forecast obligations"* [D-5].

EA was extended to deliver a sustainable and relevant EA capability that meet Businesslink strategic and operational expectations [D-4]. It was considered an essential strategic activity for the successful planning and delivery of Businesslink's current and future objectives. Enterprise architect [P-6] described Businesslink EA's approach:

> *At one level they want to use it for the strategic side of things. At a lower level they want to understand the impact upon components of the services that we provide", [P-6].*

The CEO commented on Businesslink's level of EA awareness by saying that it was implicitly integrated into the discussions around Businesslink's operating model:

> *We've probably more couched it in the context of our operating model, so used a conversation around the operating model to talk about what we do, how we do it, and articulate the architecture in that way rather than brining more concepts to the table* [P-1].

According to the lead enterprise architect [P-8], awareness of EA and understanding of its role across the organisation were unclear:

> *So Businesslink's Enterprise Architecture capability at this point is a one, really it's a one [level one of maturity], right. We're two architects and we've got to build this capability so that it's relevant to Businesslink. Businesslink as well in understanding what it needs in Enterprise Architecture is also a one* [P-8].

EA was a function that resided in the service design and architecture team in the service development division. The EA team was under development and there were some positions vacant. According to the CEO

> *We have in the architectural team a lot of technology architects, but don't have, we've still got to pick up a good enterprise architect and some good business architects in that team [P-1].*

The previous paragraphs briefly describe EA improvements during the next generation service model implementation. The next sections describe the architectural elaboration on the five adopted levels: business architecture, information systems architecture, technology architecture, EA governance as well as EA methods and tools architecture. Table 7.8 summarises the architectural elaboration (evolution outcomes).

Table 7.8 The architectural elaboration

The observed architectural elaboration	Description
Business architecture (transformed)	• The business architecture was largely considered an external architectural piece owned by Businesslink's clients due to being a shared service provider • A shared service layer was introduced between business and the three lower architectures • The service layer included services, services direction, and goals. It also provided the context for the development of services, and improved the alignment between services and its supporting process, applications, and infrastructure • Services are grouped under five main categories (service groups) (i.e., finance services, IT services, business services and client managed services • Each service was (or was going to be) mapped to its main components
Information systems architecture (reproduced)	• This level of architecture was reproduced. • Monolithic systems, integration, and master data management were obstacles that confronted the efforts to move the IS architecture to a complete service-orientated environment • Some applications/services mapping • Documentations of IS architecture were still based on fragmented Visio and Word files
Technology architecture (reproduced)	• This level of architecture was reproduced. • Documentations were still based on fragmented Visio and Word files
EA governance (reproduced)	• This level of architecture was reproduced. EA governance practices were as they were prior to the architectural interaction.
EA methods and tools (reproduced)	• This level of architecture was reproduced. EA methods and tools were as they were prior to the architectural interaction.

7.7.1 Business Architecture (Transformed)

This section discuses the architectural elaboration on the business architecture level. It covers the extension of the IT-oriented architecture after the adoption of TOGAF during SOA's introduction.

The previous EA, described in the architectural conditioning phase, was developed internally, IT-oriented, and of low maturity. During SOA's introduction, it was clear that the IT-oriented architecture was not supporting the transformation. Thus, the organisation sought to extend the IT-oriented architecture by adopting TOGAF to enable SOA's implementation [D-3]. The CEO [P-1] explained the purpose of this initiative:

> *… we took a methodological approach.… We went through a design exercise around how should we, how do we architect the organisation to do that [the new operating model].*

And participant [P-2] stated that

> *… as part of the organisational transformation, there was some EA work [P-2].*

Some aspects of the business architecture were considered external architectural pieces owned by Businesslink's clients due to being a shared service provider. According to some participants, some of the business architecture aspects such as general strategy, directions, and objectives were owned by major clients such as the Department of Family and Community Services and other organisational agencies that provided the context for the delivery of the shared services. According to service and solution design leader [P-3]:

> *The organisation is part of a super cluster, so FACS [Department of Family and Community Services] they have their own vision that our enterprise architecture is always aligned with them, so we don't want to deviate as much.*

Another participant commented: "We're a shared service provider. So we don't control the entire business layer. Most of the business layer is from our clients" [P-6].

Participant [P-6] reported that Businesslink, as a shared services provider, needed a services layer (considered an internal business layer on top of the other layers: application, information, and technology):

> *As a service provider, we thought that there was a missing link in the TOGAF model which is the shared service layer. So we built the shared service layer for Businesslink. And we built it in such a way that it can be used by any shared service organisation. And that sets, you know, service direction, goals, objectives and so forth about the services that you provide up to the business layer* [P-6].

The CEO [P-1] added: "*We inserted between the business layer, between the first and second layer a translation layer, if you like, that converted customer need, business need into service"* [P-1].

They added a shared services layer (an internal business layer) on top of their internal EA (see Figure 7.9). The shared service layer included services, services direction, goals, and processes. The service architecture was developed to allow for the better planning and implementation of the capability, processes, and systems needed to deliver the organisation's services [D-5].

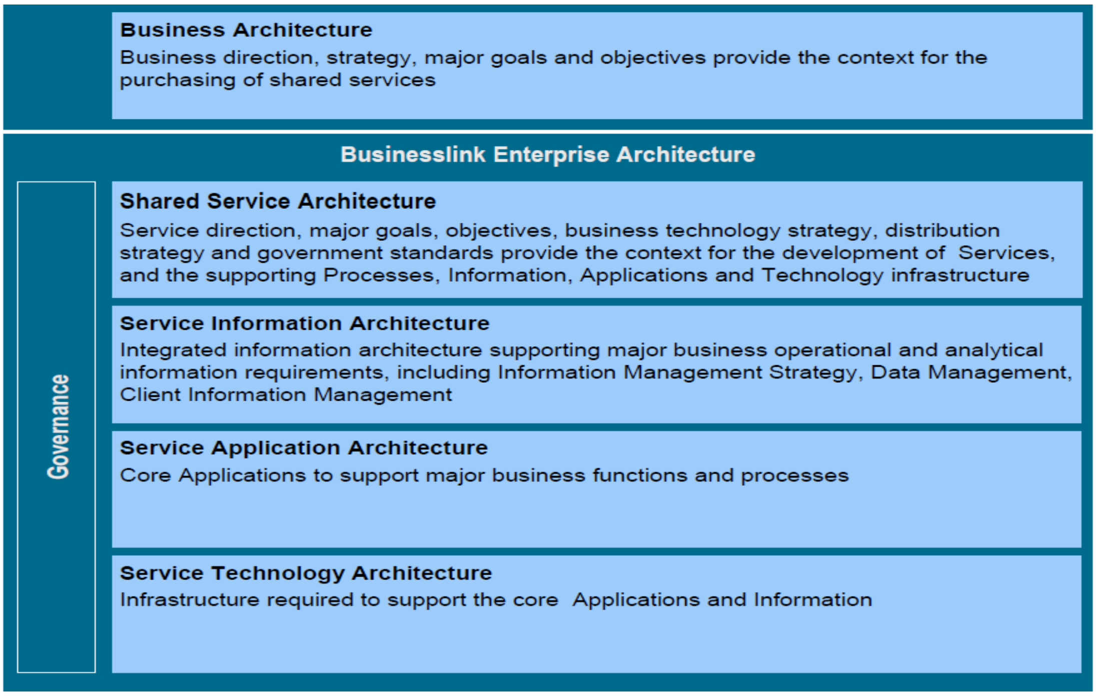


Figure 7.9 Businesslink's new EA framework [D-6]

Services were grouped under five main categories (service groups): HR services, finance services, IT services, business services, and client-managed services. Table 7.9 shows an example of the service groups, its services, and each service component [D-13].

Table 7.9 Services groups [D-13]

Service group	Service	Service components
Finance services	Accounts payable	Petty cash management
		Purchase order management
		Invoice processing
		Client specific AP transactions
		Expenses and card management

In summary, Table 7.10 presents the architectural elaboration (transformation) aspects of the business architecture.

Table 7.10 Elaboration of the business architecture of Businesslink

The observed architectural elaboration	Description
Business architecture	• Some aspects of the business architecture were considered external architectural pieces owned by Businesslink's clients due to being a shared service provider • A shared service layer was introduced between external business architecture of clients and the three lower architectures. • The service layer included services, services direction, and goals. It also provided the context for developing services and improved the alignment between services and its supporting process, applications, and infrastructure. • Services were grouped under five main categories (service groups): HR services, Finance services, IT services, business services, and client-managed services • Each service was or was going to be mapped to its main components

7.7.2 Information Systems Architecture (Reproduced)

This section presents the architectural elaboration on the information systems architecture level. In this case, most of the service-orientation activities occurred on the business architecture level and, in particular, through the organisation's restructuring in terms of services. The IS architecture was largely reproduced in terms of SOA elements. There was some SOA implementation such as the cloud-based service delivery (SaaS), but this had not transformed the pre-existing IS architecture elements; transformation would have involved the design of the IS architecture in terms of services (adding services and their relationships to the IS architecture). Instead, the existing application architecture was reproduced and labelled "service application architecture".

According to some participants, the big monolithic systems, integration, and master data management were obstacles that confronted the efforts to move to a complete service-orientated environment. For example, participant [P-2] noted:

> *great difficulty unpicking various things, to just try and pick one service and outsource it, you will have a lot of constraints there, you've got a whole lot of integration problems that you need to solve. You've got master data problems that you need to solve... the technology and the monolithic systems like ERP don't support the organisational vision of picking services and getting them from wherever they might best come from.*

The mapping between the business services and its supporting applications varied from service to service:

> *It depends on the service and its maturity.... FMS [Funding Management Service] is one where it is really well articulated, everything that's in there. There are others where we are now articulating all of the components that make up the service. Because we've got people that have moved into positions and they're now going to be accountable for the services. They need to know what their components are* [P-10].

The documentation was project based in Visio and Word documents. For example, Table 7.11 shows the mapping of two services and their supporting applications and components associated with the "single ERP project". It was a project to merge three ERP systems into single ERP to standardise the application landscape [D-18].

Table 7.11 Service to application mapping [D-18]

Service	Application	Functional description
KIDS	Siebel Public Sector 7.8	Community services uses the KIDS system to provide adoptions, child protection, and out of home care, early intervention, service provider, allowances, performance of service information and management Integration with SAP ECC via SAP PI KIDS system triggers allowance payment and re-imbursement payment to foster parents in the SAP ECC system KIDS system triggers vendor master data maintenance to the SAP ECC system SAP ECC sends payment status to KIDS Technical component in use: SAP ECC – FI/AP, SAP PI
COMS	Siebel Public Sector 7.8	Community services uses COMS (commercial online management system) to manage the payment of grants to NGO's and community groups COMS triggers payment files to SAP SAP ECC sends payment status to COMS Sap ECC sends vendor creation/changes to COMS. Technical component in used: SAP ECC – FI/AP, SAP PI

In summary, the information systems architecture was reproduced after the implementation of the next-generation service model. SOA's integration into this layer of EA was very limited.

Table 7.12 Elaboration of the information systems architecture of Businesslink

The observed architectural elaboration	Description
Information Systems Architecture	• This level of architecture was reproduced. It had the large monolithic applications that supported the delivery of business services • Monolithic systems, integration, and master data management were obstacles that confronted the efforts to move to a complete service-orientated environment

	• Some applications/services mapping • Documentations were still based on fragmented Visio and Word files

7.7.3 Technology Architecture (Reproduced)

This section discusses the architectural elaboration on the technology layer of EA. This layer (exiting technologies prior to the next generation service model) was reproduced. SOA was not integrated into the technology architecture. There was no standardised service enabling infrastructure platforms such as ESB. There was also little focus on service automation and self-service or enabling workflow technologies [D-2]. The supporting infrastructure models were still as they were prior to SOA's introduction. For example, participant [P-10] explained that the lower layers of EA were still represented in EA in terms of large applications and supporting infrastructure:

> *Let's use, for example, we have something called the Funding Administration Service, where we look after an application called FMS. It sits on Solaris boxes, it uses sand storage, it uses backup products, [and] it uses the network*

7.7.4 EA Governance (Reproduced)

This section addresses the architectural elaboration on the EA governance level. There was noticeable integration of service governance with the wider organisational governance practices; for example, the restructuring of service and service components-related responsibilities and accountabilities. Participant [P-10] noted:

> *So the whole idea around our shift in the enterprise model is to give someone the accountability, but highlight on the way through you provide these components. So it's shifting it from that way to that way, and it's an end to end, so that the projects that are delivering something from the design, the sourcing, to what's the end service, there's clear accountability and responsibility.*

Also, some committees were established alongside the service development lifecycle. Participant [P-2] said:

> *There's a service, a new service and enhancements committee which actually will review all service designs and service roadmaps, but what we, what we haven't really done is formally establish an architecture review board.*

However, the integration of services governance with the architectural governance practices was under development. The CEO reported that their EA governance is still in its early stages [P-1].

In summary, there was a large overlap between services governance and the wider organisational and project governance practices. Yet, EA governance practices were still in development. It was not clear how the service development and its related activities affected EA and vice versa.

7.7.5 EA Methods and Tools (Reproduced)

This section presents the architectural elaboration at the EA methods and tools level. First, EA did not have specific formal repository or tools. It was manually conducted in a fragmented approach using Word, Visio, and other similar formats to document the architectural artefacts. The old EA methods and tools were reproduced. There were no noticeable changes to this aspect of EA due to SOA's introduction.

However, the service development was aligned with the new operating model and the project delivery approach. For example, one of the new guidelines was that new and enhanced services needed to be developed in accordance with the operating model and service design [D-17]. The next-generation service model implementation changed the way services were developed or improved according to some participants. For example:

> *So it's not a traditional application development because you are talking about service development, and it is possible that that particular component of the service, the solution to that is better sourced externally because it has matured already in the cloud or it will deliver more cost efficiency if we're going to do that* [P-4].

And:

> *What's changed here is that we work much more together in terms of delivering a common business outcome, so what that means is in terms of benefits, you involve the people in the run, that service*

delivery right up front so you define, design your system for how it is going to be used not about how you're going to develop it [P-5].

7.7.6 Summary

In summary, the business architecture (internally called "service architecture") layer was the one EA layer that was transformed. The business architecture integrated SOA and had services groups, services, and service components. The other layers of EA were largely reproduced, which could be because: (1) EA during the architectural conditioning phase was at a very basic level of maturity (level one), (2) the extended EA during SOA's introduction was strategically-oriented towards the development of high-level roadmaps for the delivered services, and/or (3) SOA's introduction was mostly focused on the business side of the organisation.

7.8 Chapter Summary

This section summarises the results of this case. In this case, the evolution outcomes at (T4) were understood by retrospectively looking at the architectural interaction (T2-T3) and the architectural conditioning (T1) phases. Figure 7.10 summarise the results of this case using the theoretical model of this thesis. In this figure, the architectural elaboration (T4) represents the SOA's integration into EA outcomes.

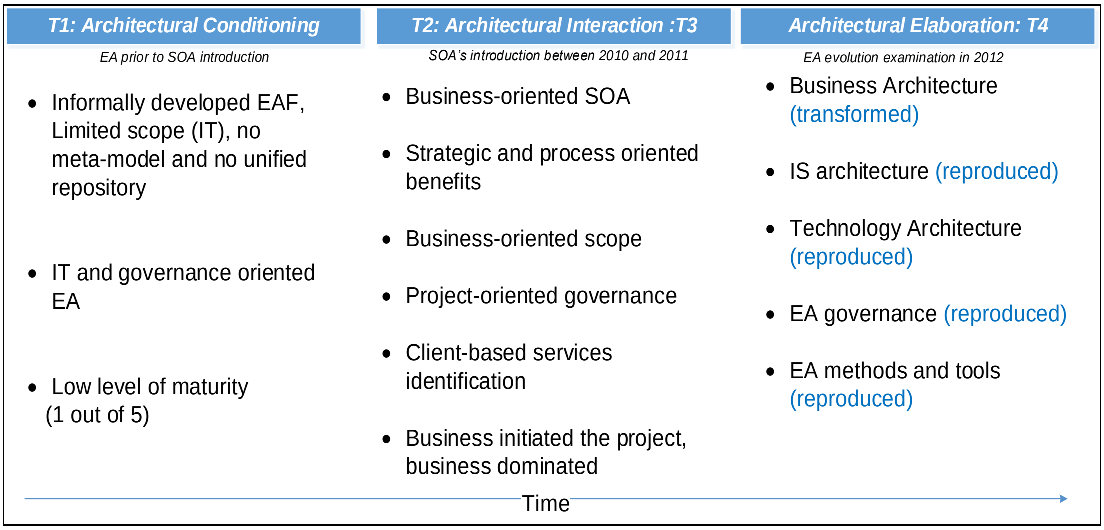


Figure 7.10 The morphogenetic cycle of SOA's integration into EA at Businesslink

First, the architectural conditioning phase was considered the period prior to SOA's introduction (the implementation of the next generation service model). This architectural conditioning is described in Section 7.5 according to three conditional generative mechanisms: EA framework, EA objectives, and EA maturity. EA was implemented using an in-house developed IT-oriented framework. EA's maturity was very low. The primary architectural artefacts were solutions, systems, and infrastructure. They were documented in loosely managed distributed documents. It was mostly project-oriented documentation. Although there was a defined governance approach around the change of the artefacts, they were inconsistently updated. The focus on applications and technology artefacts could be due to the fact that the organisation was mostly a host of its clients' infrastructures and solutions.

Second, Businesslink introduced SOA (the next-generation service model implementation) in late 2010 as described in Section 7.6. The project was undertaken to strengthen Businesslink's presence as a major provider of outsourced business services in the Australian public sector. The scope of the project was very large and included the transformation of the organisation's operating model to become service-oriented. It was influenced by the six action-formation mechanisms presented in Table 7.13.

Table 7.13 Contextualisation of SOA's integration into EA at Businesslink

Analytical phases	Generative mechanisms	Actualisation
Architectural conditioning		
	EA maturity	low maturity level (1)
	EA framework	In-house developed EA framework and methods
	EA objectives	IT and IT governance oriented
Architectural interaction		
	View of SOA	Enterprise services architecture view (mostly business view, not traditional SOA)
	SOA scope	Organisation wide (with high concentration on the business/service architecture)
	SOA benefits	Driven by strategic and business benefits
	SOA governance	SOA was governed against existing organisational governance
	SOA design	Services were based on clients requirements, each service had or was going to have a roadmap, services classified and managed in PDF files

	Business-IT collaboration	Business-driven transformation project, implemented mostly by business team
Architectural elaboration (Outcomes)		
	Business architecture (transformed)	SOA was integrated into the business (service) architecture. It had the business services as a major SOA's element
	IS architecture (reproduced)	No integration
	Technology architecture (reproduced)	No integration
	EA governance (reproduced)	No integration
	EA methods and tools (reproduced)	No integration

Third, SOA's introduction resulted in the following architectural elaboration outcomes. The business architecture was the only transformed level (SOA was integrated). The old IT-oriented architecture at (T1) constrained SOA's integration into EA and, therefore, TOGAF was adopted to extend the IT-oriented EA and support SOA's introduction. The business (service) architecture was service-oriented and was added on top of their old IT-oriented architecture. It was transformed by redesigning the organisation in terms of services, and by adding the SOA-related elements such as services, services direction, and goals.

The IS and technology layers of EA were not transformed by SOA's introduction. There was limited integration between SOA and the lower layers of EA, which could be due to the fact that: (1) EA was at a very basic level of maturity (level 1), (2) even with the expansion of the IT-oriented EA through the adoption of customised TOGAF, it was strategically-oriented towards the development of high level roadmaps for services, and (3) SOA's introduction was business-oriented to re-organise the organisation to be service-oriented.

With respect to the EA governance level, the next-generation service model's implementation was not integrated into EA governance and that could be due to the fact that EA governance was still emerging and/or because of the low level of EA maturity. The fifth level of the architectural elaboration was the transformation of EA methods and tools. There was no clear integration of SOA methods and tools into EA methods and tools. This could be due to the lack of well-defined EA method and tools. EA tools were not yet established.. Yet, the traditional solution procedures made aware of services and design for services.

Chapter 8: Cross-case Analysis

8.1 Introduction

This chapter compares the two case studies presented in Chapters 6 and 7. The results of two cases are compared to understand SOA's integration into EA and its outcomes in the two cases and their similarities and differences using the developed theoretical model's three analytical lenses.

The chapter progresses as follows. First, Section 8.2 provides the context for both cases. Section 8.3 presents and compares the architectural conditioning phase for both cases. In particular, it compares the cases based on the three conditional generative mechanisms: EA framework, EA objectives, and EA maturity. Section 8.4 compares the two cases' architectural interaction (SOA's introduction) phases. In particular, it compares the two cases in terms of the six action-formation generative mechanisms related to SOA's introduction. Section 8.5 compares the architectural elaboration due to SOA's introduction (either transformation or reproduction of the pre-existing architectural settings) on five architectural levels. Section 8.6 uses this thesis's theoretical model to collectively compare the three phases of both cases. Section 8.7 summarises this chapter.

8.2 Contextual Description

This section briefly describes the two organisations (cases) included in this thesis (see Chapters 6 and 7 for more information).

First, Dubai Customs is a government agency in Dubai. It was established in 1920 and has a long history. In 2001, Dubai Customs, Dubai Ports, and Free Zone Corporation merged. In 2006, Dubai Customs was one of the first government organisations to undertake the reform and modernisation program (RMP), which was designed to help Dubai Customs achieve its strategic objectives.

Second, Businesslink is a company that provides shared corporate services to many N.S.W. agencies, such as the Department of Ageing, Disability, and Home Care (DADHC) and the N.S.W. Department of

Community Services. It was originally established as a division of the N.S.W. Department of Housing in 2002. Afterwards, it became Government-owned private company in 2004.

Both Dubai Customs and Businesslink implemented SOA initiatives. Prior to that, both were silo-based organisations. Table 8.1 shows their objectives.

Table 8.1 Case study profiles

Criteria	Dubai Customs	Businesslink
Type of organisation	• Government organisation (Dubai)	• State Government-owned private company (Australia)
Number of employees	• Around 3000	• Around 900
Years of establishment	• Well-established organisation in 1920. • In 2001, Dubai Customs, Dubai ports and Free Zone Corporation merged	• In 2002, Businesslink was established from an established government agency • In 2004, it became the a private company
Operating model	• Started an initiative in 2008 to adopt a service-oriented operating model • Prior to 2008, a traditional silo-based operating model	• Started an initiative in 2010 to adopt a service-oriented operating model. • Prior to 2010, a silo-based operating model
Objectives	• Contribute to Dubai's economic and social development • Adopt and share best practice in terms of business processes and systems • Provide best human and technical resources, and • Improve customer satisfaction and loyalty	• Provide outsourced services on behalf of its clients (government agencies) directly to the community • Enable agencies to focus on their core business, • Realise the benefits of technology • Reduce agencies' costs
Enterprise architecture	• Positioned in the customs development division • EA included strategy, people, processes, information, and technology	• Positioned in the IT division prior to the service-oriented transformation in 2010. Then, positioned in the service development division • EA included applications,

		infrastructure, data, and security

The next sections (8.4 to 8.6) compare the two cases using these three analytical lenses of the theoretical model (see Figure 8.1) in order to understand EA evolution's similarities and differences in each case.

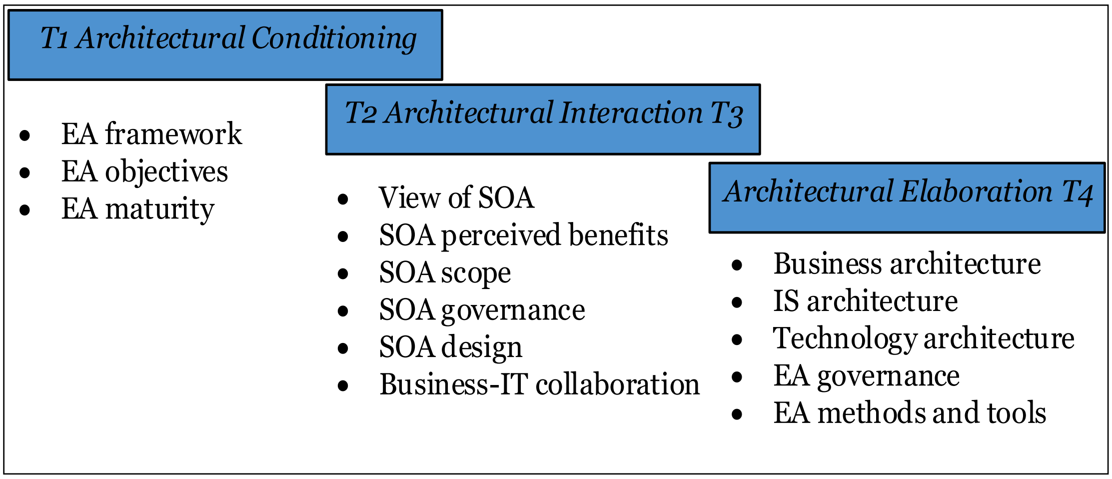


Figure 8.1 This thesis's' theoretical model

8.3 Architectural Conditioning

The first EA evolution phase, architectural conditioning, is the period of EA stability prior to SOA's introduction. This architectural conditioning phase is described according to three conditional generative mechanisms: EA framework, EA objectives, and EA maturity. The next sections discuss each generative mechanism and the similarities and differences in each context to build a bigger picture of EA evolution (SOA's integration into EA).

8.3.1 EA Framework

This section compares the first generative mechanism of the architectural conditioning phase, EA framework. Dubai Customs' EA framework was established in 2006 as part of the organisation's reform and modernisation initiative. It was an organisation-wide EA. By the end of 2007, most of the organisational artefacts, such as strategists, processes, applications, and their relationships, were stored in the EA repository (IBM Systems Architect). The EA framework was developed in-house with external consultants' engagement. It was built based on the TOGAF and the Zachman Framework and was aligned with the TOGAF's structure.

Businesslink's EA framework was also developed in-house. It had seven IT domains: applications, network, integration, systems management, data, platform, and security. It focused on documenting the IT domains using fragmented artefacts such as Excel, Word, and Visio documents. There was neither a well-defined EA methodology nor a very comprehensive EA framework. Table 8.2 summarises the two cases' EA framework characteristics.

Table 8.2 Case study EA frameworks

EA framework characteristics	Dubai Customs	Businesslink
Architectural base	• In-house developed based on the TOGAF and the Zachman Framework • EA was part of a large transformation initiative	• In-house developed partial framework • EA was developed by the IT division (internal project by itself)
Architectural layers (domains)	• Strategy, resources, process, information, and technology	• Seven IT-oriented domains: applications, network, integration, systems management, data, platform, and security
Architectural scope	• Organisation-wide EA program	• EA was an IT-oriented architecture
EA methodology	• Well-defined methodology with five main steps (i.e., assessment, planning, execution, monitoring, and governance)	• No specific methodology

As Table 8.2 shows, the two organisations' EA frameworks noticeably differed. First, the analysed data indicated that Dubai Customs' architectural base was developed using well-established existing EA frameworks. It was comprehensive EA that included strategy, business, and IT. On the other hand, Businesslink's EA was partial (IT-oriented), internally developed, and had a structure that was not based on an existing EA framework. Further, EA at Dubai Customs was part of a large transformation project to help transform the organisation. On the contrary, the EA of Businesslink was an internal project with limited scope. In addition, Dubai Customs had a well-defined EA methodology and well-established EA tools, while Businesslink had neither a defined EA methodology nor well-established EA tools.

8.3.2 EA Objectives

This section compares the second generative mechanism of the architectural conditioning phase, EA objectives. Dubai Customs' EA was adopted to achieve strategic, governance, IT, and operational objectives. Examples of the realised benefits were improved business and IT alignment, improved decision-making processes, reduced costs, and reduced complexity of the business and IT landscapes.

Businesslink's EA was oriented towards IT and IT governance. EA's objectives were less defined. In Businesslink, the most noticeable objectives of EA were the development of IT standards, high-level technology roadmaps, and architectural building blocks. Table 8.3 summarises the two organisations' EA objectives.

Table 8.3 Case study objectives

EA Objectives	Dubai Customs	Businesslink
Strategic	• To align Dubai Customs with the wider Dubai strategies • To align business and IT • For strategic alignment • For change management • To identify gaps and build roadmaps	• There were no explicit strategic objectives of EA
Operational	• Single source of truth • for daily activities (accessible to both business and IT personnel) • For impact analysis, discovery of duplications and to improve standardisation	• There were no explicit operational objectives
Governance	• Demands/projects are evaluated against the strategy and both business and IT architectures • EA includes strategy domain, which was used to guide the	• Ad-hoc IT governance-oriented objectives • Focused on managing IT standards and

	other domains and EA's activities • EA includes architectural standards	technology roadmaps (high level)
IT	• Provide solutions' requirements • Reduce IT duplications • Reduce IT complexity	• Ad-hoc and focused on the development of architectural building blocks to standardise solutions and infrastructure

As Table 8.3 shows, the two organisations' EA objectives noticeably differed. Dubai Customs' EA was adopted to achieve diverse objectives at the strategic, operational, IT, and governance levels. Businesslink's EA was adopted to achieve only IT and IT governance-related objectives. The two organisations differed regarding the formality of their EA objectives. Dubai Customs' EA was formally established as a part of a transformation project and was driven by high expectations. Businesslink's EA was informally established by the IT division, and its value was tied to the IT domain.

8.3.3 EA Maturity

This section compares the third generative mechanism of the architectural conditioning phase, EA maturity.

The two organisations' EA maturity level noticeably differed. Dubai Customs' EA was mature (between levels 3 and 4 out of 5) prior to SOA's introduction. The documentation of the whole organisation was comprehensive and it was stored in the EA repository. EA had a well-defined methodology. EA governance practices were established and integrated with organisational governance. EA was involved in demand and project governance to ensure that projects were aligned with both strategy and architectural (business and IT) standards. EA's content and its meta-model were kept up-to date. EA had diverse and skilled team members. There were about twelve strategists, business architects, and technology architects. EA was supported by the top management.

On the other hand, Businesslink's EA was at level one (informal program) of maturity. The documentation level was superficial and focused on IT aspects only. There was neither a unified repository nor a unified methodology for doing EA activities. A high-level governance approach around the change of the artefacts was documented. However, a copy of that repository showed that some artefacts had not been updated for a long time and/or had missing information. There was neither a formal EA team nor sufficient resources. Table 8.4 summarises the two organisations' EA maturity.

Table 8.4 EA maturity comparison

EA maturity	Dubai Customs	Businesslink
Documentation	• EA documentation was comprehensive • Strategy, business, IS, & infrastructure information were captured and stored in a repository • An interface tool was built to browse, query, and navigate the repository and its content by business and IT personnel	• High-level documentation of IT artefacts • Fragmented documentation • Focused on standards and roadmaps
Planning	• EA program was well defined and had a structured framework and timeline for developing the EA • EA planning was well integrated with major strategic initiatives to help the organisation to achieve these initiatives objectives • EA was involved in building roadmaps for organisational improvements • EA enabled assessment of the current situation, identification of gaps, and development of roadmaps and action plans	• EA activities were informal and unstructured • There were no explicit EA methodology • EA development was ad-hoc • There were some inconsistent activities for documenting technology roadmaps
Governance	• EA governance standards, processes, and procedures were established and employed. EA roles were defined, and review committee was established.	• There were no well-established governance practices • The was a need for organised committees to define the architectural standards and processes
Evaluation	• EA and its products were evaluated on two sides. First, EA, its methodology, & meta-model were reviewed and assessed periodically every two	• Evaluation processes were ad-hoc and informal • Out-dated artefacts

	years. Second, EA was reviewed and changed when needed (e.g., new trends or requirements)	
Team and resources	• EA team was defined. Tools, frameworks, and resources were available for EA team to support their activities	• No official EA team • The need for a capable EA team was identified
Business Support	• EA and its activities were supported by the top management. EA was engaged with business and IT	• Limited or non-existent business support

8.3.4 Summary of the Architectural Conditioning Phase

The findings show that Dubai Customs adopted a framework that follows well-established frameworks; namely, the TOGAF and the Zachman Framework. On the other hand, Businesslink adopted its own EA framework that focused only on IT domains. Dubai Customs adopted EA to achieve strategic, operational, IT, and governance objectives, while Businesslink adopted it to document and govern the IT landscape. Moreover, Dubai Customs' EA was mature while Businesslink's EA was not.

There seem to be associations between these three generative mechanisms. In Dubai Customs, the objectives were comprehensive, which we can associate with the well-established EA framework and mature level of EA practices. In Businesslink, the objectives were limited: the EA framework was internally developed to deal with the IT domain, and EA practices had low maturity. Businesslink's EA objectives were "to develop standards that foster consistency, compliance, efficiency and cohesion in ICT solution design" [Businesslink D-9] and a participant in the Businesslink study stated that "there was a significant focus on infrastructure and solutions rather than enterprise global architecture" [Businesslink P-2].

These three conditional generative mechanisms describe the state of the architectural conditioning of each case and provide a clear picture of their EA settings prior to SOA's introduction. Using Archer's (1995) argument of the conditional impact of the architectural conditioning phase (see Chapter 4 for details), pre-existing EA settings condition but do not determine the outcomes of SOA's integration into EA. The architectural conditioning phase enables or constrains an organisation's ability to transform its existing EA (e.g., through SOA integration). Both cases had a different architectural conditioning phase, which may have influenced SOA's integration into EA. Section 8.6 discusses the impact of this phase on SOA's integration into EA and the next Section 8.4 compares the architectural interaction (SOA's introduction) phase of the two cases.

Table 8.5 Case study architectural conditioning

Architectural conditioning	Dubai Customs	Businesslink
EA Framework	• Well-developed based on the TOGAF and the Zachman Framework • Well-described using layers and meta-model • Implemented using a tool (IBM Systems Architect)	• Internally developed to cover the IT domain only • No defined meta-model • Not implemented. It was based on Word, Excel, and Visio files
EA objectives	• Strategic, operational, governance and IT (realised)	• Governance and IT (informal)
EA maturity	• Mature	• Low maturity

8.4 Architectural Interaction (SOA's Introduction)

This section provides an overview of SOA's introduction in both cases. Then, it compares the two cases in terms of the six action-formation generative mechanisms that were identified to have an influence on SOA introduction.

In both cases, there were issues with the delivery of end-to-end services prior to SOA's introduction, and the action (SOA's introduction) was undertaken in response to the need to move to a service-oriented organisation to improve service delivery and client satisfaction.

Dubai Customs' SOA was introduced in 2008 to implement a web-based, scalable, and feature-rich business-to-government suite to improve service delivery. The implementation included an online risk engine for risk assessment. The SOA suite was introduced in response to a government-wide initiative of service delivery improvement. It was accompanied by an internal restructuring to embrace an outside-in strategy with respect to service delivery thinking. It was implemented to provide paperless services that are supported by IT services, processes, and resources.

Businesslink's SOA was implemented in late 2010 to reorganise the organisation in terms of services that Businesslink provides to its clients. Businesslink implemented a transformational SOA project called the "next generation service model" to strengthen its presence as a primary supplier of outsourced business services in the Australian public sector. The next generation service model included the transformation of their operating model and the adoption of an innovative service-oriented organisation structure.

8.4.1 View of SOA

The first action-formation mechanism that influences SOA introduction is view of SOA. Both cases viewed SOA differently, which influenced the way SOA was introduced. SOA at Dubai Customs encompassed both business and IT (enterprise services architecture view as presented in Chapter 2), and was considered as a business strategy and an architectural style. Dubai Customs' business processes were redesigned, and business services were identified and aligned with IT services. On the other hand, SOA at Businesslink was a business strategy to redesign the organisation in terms of services. Businesslink did not emphasise services at the technology level as much as Dubai Customs did.

8.4.2 SOA Perceived Benefits

The second action-formation mechanism that influences SOA introduction is SOA perceived benefits. The two cases have shared the perceived benefits of SOA at the strategy and process levels. For example, both cases had very similar strategy-related benefits of SOA (i.e., both organisations were restructured to be service-oriented organisations, to improve business processes and to increase customer stratification).

However, Dubai Customs had very strong IT-related benefits associated with its SOA implementation, whereas Businesslink had very minimal IT-related benefits. The findings show that Dubai Customs implemented SOA increase reuse, improve IT integration, reduce IT complexity, and reduce IT maintenance. On the other hand, the findings show that Businesslink did not achieve nearly as many IT-related benefits as Dubai Customs did.

8.4.3 SOA Scope

The third action-formation generative mechanism that influences SOA introduction is SOA scope. Both cases had very large SOA projects. The analysis shows that Dubai Customs had an enterprise-wide project that lasted for around two years. During the project, processes were automated, and SOA was aligned with strategy, business, and IT. Services were managed in a service portfolio. SOA best practices were adopted and used. The scope of Businesslink's SOA was comprehensive: it focused mostly on redesigning the organisation's business architecture to be service-oriented. The organisations were restructured using services as the major structuring element. There were some implementations at the technology level, such as cloud-based service delivery, but these were not as extensive as the transformation of the business side of the organisation.

8.4.4 SOA Governance

The fourth action-formation generative mechanism that influences SOA introduction is SOA governance. There was some governance practised during SOA's introduction in both cases. However, the application and approach was different in each case.

The findings show that, at Dubai Customs, SOA's introduction was governed by and aligned with the wide-organisational governance framework COBIT. It was governed and enabled by the eservices delivery excellence model, which provides guidelines and evaluation criteria for services delivery quality. SOA's introduction was governed against the wider EA governance and through the use of the adopted IBM SOA reference architecture. On the other hand, at Businesslink, SOA's introduction was governed using traditional (project-based) organisational governance. Some governance committees were established to manage services through their lifecycle. However, there was neither a specific SOA governance framework such as the one used in Dubai Customs nor explicit EA governance governing SOA's introduction.

8.4.5 SOA Design

The fifth action-formation generative mechanism that influences SOA introduction is SOA design. The findings show that SOA design at the two organisations was different. Both cases had a long-term plan for SOA. However, Dubai Customs' roadmap was at SOA initiative level (two years), while Businesslink's roadmaps were partially created based on each service having its own roadmap for future improvement.

Dubai Customs' services identification was based on a top-down approach, while Businesslink's was driven by clients' requirements (restricted by the delivery of services that certain clients require. Dubai Customs classified services into business and technical services, Businesslink had business service where each service had service components (processes, people, and IT).

Dubai Customs' services were stored in a repository (IBM System Architect) with other architectural elements. Businesslink's services were kept in a static (PDF) file as a service portfolio that had only services-related information.

8.4.6 Business and IT Collaboration

The sixth action-formation generative mechanism that influences SOA's introduction is the level of business and IT collaboration. The findings show that Dubai Customs clearly had a high level of business, IT, and external vendors' engagement. SOA implementation in Dubai Customs was supported by key stakeholders, and clients were also engaged. The implementation was executed by experts from inside the organisation and from external vendors. Businesslink had a high level of business engagement but less IT engagement because its SOA introduction focused more on business architecture. It was supported by the organisation's top management as part of a transformation initiative to shift the organisation's focus from technology to services delivery. Key stakeholders were involved in SOA's implementation.

Dubai Customs' SOA was driven by a highly skilled and diverse team of internal and external stakeholders and large vendors. However, Businesslink's team was mostly internal and involved key stakeholders. Table 8.6 summarises the findings.

Table 8.6 SOA's introduction comparison

Generative mechanism	Dubai Customs	Businesslink
View of SOA	• Fits within enterprise services architecture view of SOA' view classification (Business and IT levels • Dubai customs moved beyond the IT-focused management of services toward defining services driven by business and clients requirements • Service definition was directly tied to business requirements • Business services were defined, identified, and aligned with IT • Business processes were redesigned to achieve agility	• Fits within enterprise services architecture view of SOA' view classification (mostly business and not traditional SOA) • Business-oriented view (redesign of the organisation) • The organisation was restructured to improve services delivery
SOA perceived benefits	• It was driven by strategy, process, and IT benefits • Shift in organisational thinking (shift from internally focused design, inside-out thinking to a customer-oriented design, outside-in thinking) • Dubai Government initiative to deliver eservices, "Dubai model for government services" • Improved agility to accommodate existing and anticipated levels of trade • Reduced maintenance, increased information availability and reuse	• SOA's introduction was driven mostly by strategy and process benefits. There was less emphasis on the IT benefits • Shift in organisational thinking (shift from internally focused design and silos to a customer-oriented design, service-oriented organisation) • Delivery of end-to-end services to improve customer experience • Improved agility to accommodate existing and anticipated demands • Reduced maintenance, and improved communication and business processes

SOA scope	• Organisation-wide initiative • Processes were automated • SOA was aligned with business strategy • SOA's implementation involved business people • Service portfolio was established and managed as part of EA • SOA best practices were adopted and promoted with the help of large vendors such as Oracle and IBM • SOA was strategically adopted and implemented	• High emphasis on the redesign of the business side of the organisation in terms of services • SOA was aligned with business strategy • SOA was strategically adopted and implemented • Organisational alignments were considered
SOA governance	• SOA governance was established and practised • The introduction of Dubai eservices delivery excellence model (EDEM) • Services standards and guidelines were established • Services were governed through different stages: enablement, delivery, and evaluation • SOA projects were governed against EA architectural practices	• SOA governance was business oriented, not much on the technical side of SOA • Project-based governance practice was used to govern SOA implementation. It was related to the change management and project governance in general • Key stakeholders were involved and new services' governance committees were also established
SOA design	• Services were identified using a top-down approach. Business services were identified and linked to technical services • An internal classification of business services into three tiers was employed • Services were kept with other EA elements in a repository that was accessible to browse, query, and search for info about business and IT services and how they are associated with other EA	• Business services were identified (developed or in-sourced) based on customers' requirements (top-down) • An internal classification of services into components (process, people, and IT) • Business services were classified into different types • Each service was developed according to a defined service lifecycle which includes the evaluation of services

	elements • IBM SOA reference architecture was used as a guideline for SOA's environment at Dubai Customs • Dubai customs had a clear roadmap (strategy) for SOA adoption	• Some services had roadmaps (2-3 years) • A static document (service catalogue) was developed. It had information about the provided services such as description of the service and its pricing
Business and IT collaboration	• Business and IT collaborated to drive SOA. Key stakeholders, external vendors and clients were involved. • The demand management process was designed to engage business and IT when a new demand is initiated that includes SOA projects • Dubai Customs had and continues to acquire qualified, skilled people to drive its SOA	• The next-generation service model was business-driven and supported by both business and IT • Internal and external stakeholders were involved

8.4.7 Summary

In summary, SOA's introduction in Dubai Customs and Businesslink had both similarities and differences. The six action-formation generative mechanisms presented in Table 8.7 influenced the way SOA was introduced in both organisations. Differences in organisations' orientation, interests, and resources influenced the architectural interaction (SOA's introduction).

Table 8.7 Summary of SOA introduction in both cases

Action generative mechanism	Similar or different	Dubai Customs	Businesslink
View of SOA	Different	Enterprise services architecture view (Business and IT levels)	Enterprise services architecture view (mostly Business level, not traditional SOA)
SOA perceived benefits	Different	Strategy, business and IT benefits	Strategy and business benefits
SOA scope	Different	Organisation-wide	Mostly business side of the organisation
SOA governance	Different	SOA was governed against a reference architecture, EA project, and organisational governance	SOA was governed against organisational and project governance
SOA design	Different	Services were identified top-down, SOA had a well-defined roadmap, services classified and managed (in EA repository)	Services were based on clients requirements, each service had or was going to have a roadmap, services classified and managed in PDF files
Business/IT collaboration	Different	High level of business/IT collaboration, external consultants, skilled and trained SOA team	Business-driven services transformation project, implemented mostly by business team
Summary of SOA's introduction			
SOA's introduction into Dubai Customs	• SOA was implemented in 2008 • It was a web-based, scalable, and feature-rich business-to-government suite to improve service delivery • It was in response to a government-wide initiative of service delivery improvement • It was accompanied by an internal restructuring to embrace an outside-in strategy with respect to service-delivery thinking		
SOA's introduction into Businesslink	• SOA was implemented in late 2010 • It was a transformational project called the "next generation service model" • It included the transformation of their operating model and the adoption of an innovative service-oriented organisation structure to o be a primary supplier of outsourced business services in the Australian public sector.		

8.5 Architectural Elaboration

This section compares the architectural elaboration (the outcomes of SOA's integration into EA) in Dubai Customs and Businesslink. The elaboration could be a transformation or reproduction of the pre-existing EA settings. As this thesis's theoretical model shows (see Section 8.2), there are five possible architectural elaboration (evolution) outcomes: business architecture, information systems architecture, technology architecture, EA governance, and EA methods and tools. Applying Archer's (1995) terms, the architectural elaboration (outcomes) are explained by retrospectively examining the architectural interaction (SOA's introduction) and the architectural conditioning (T1) impact. These elaborations in both cases represent the EA evolution outcomes at (T4).

The findings show that the architectural elaboration of the two cases differed. Dubai Customs' architectural elaboration was a transformation of the EA settings at all five of the architectural levels. On the other hand, Businesslink's architectural elaboration was a transformation of the business architecture and a reproduction of the other architectural levels. The following paragraphs compare the architectural elaboration at these levels.

8.5.1 Business Architecture

The first architectural elaboration level is business architecture. Both Dubai Customs' and Businesslink's business architectures were transformed; that is, SOA was integrated into the business architecture of both organisations. In particular, Dubai Customs integrated SOA-related elements such as business services, their descriptions, channels and owners into the business architecture. These elements were part of the EA meta-model and their instances are captured in the EA repository (IBM System Architect). For Businesslink, the business architecture (internally called "service architecture") had service groups and services, but there was no formal meta-model of their EA.

8.5.2 Information Systems Architecture

The second architectural elaboration level is information systems architecture. The cases were different at this level. Dubai Customs transformed its information systems architecture. The IS architecture integrated SOA-related elements such as technical services, service operation, and service realisation. These elements were added to the meta-model and integrated with the other architectural elements. On the other hand, Businesslink's information system architecture was reproduced. The same practices that were used prior to SOA's introduction were still used.

8.5.3 Technology Architecture

The third architectural elaboration level is the technology architecture. The cases were also different at this level. Dubai Customs transformed its technology architecture. This architecture integrated and supported SOA elements such as technology environment, instance, interface, interface messaging, and message structure. It also supported mapping between services and their supporting infrastructure. In contrast, Businesslink's technology architecture was reproduced. SOA's introduction did not transform the technology architecture.

8.5.4 EA Governance

The fourth architectural elaboration level is EA governance. The cases were again different at this level. Dubai Customs transformed its pre-existing EA governance. In particular, SOA governance had its own governance practices, which extended EA governance. The service lifecycle was integrated with EA because of EA engagement's with project/solution management. EA had standards and principles that apply to services and the other architectural elements such as processes and applications. Each service was governed using various SLAs, which were both technical and business in nature. Businesslink reproduced its pre-existing informal EA governance and there was no explicit governance practices integration.

8.5.5 EA Methods and Tools

The last architectural elaboration level is EA methods and tools. The cases were again different at this level. Dubai Customs' EA and SOA had overlapping methods and tools. For example, business services were identified on the business architecture and mapped to technical services using a top-down approach. EA was integrated with projects/solutions, which included SOA solutions. EA deliverables were used to deliver (SOA) project requirements. EA also reviewed SOA projects, monitored their implementation, and ensured they delivered their objectives. Further, requirements, design, and development documents were generated by EA during the design and implementation of projects, including SOA projects. In return, these projects provided any required architectural changes into EA. In contrast, Businesslink had no specific EA method prior to SOA's introduction, and thus there was no such integration. It also showed that there was neither a specific EA tool (repository) nor an EA meta-model, and thus there was no integration. The pre-existing ad-hoc processes and fragmented EA documentation (repositories) were used and allowed only limited integration of SOA. Yet, there was some project-specific documentation about certain services and their components. Table 8.8 summarises the two organisations' architectural elaborations.

Table 8.8 Architectural elaboration comparison

Architectural elaboration T4	Dubai Customs	Businesslink
Business architecture	• Process layer became "business layer" to incorporate business services besides other business architecture elements • Redesign of the organisation in terms of domains and each domain has its provided services • Design of business architecture in terms of services • New SOA-related elements were added to business architecture such as business services, their descriptions, supported channels, client groups, service scenarios, and owners • Business services were mapped to other business architecture elements • Business services viewpoints were added	• The business architecture was largely considered an external architectural piece, owned by Businesslink's clients, due to being a shared service provider • A shared service layer was introduced between business and the three lower architectures • The service layer included services, services direction, and goals. It also provided the context for developing services and improved the alignment between services and their supporting process, applications and infrastructure • Services are grouped under five main categories (service groups). They are HR services, Finance services, IT services, business services, and client-managed services • Each service was or was going to be mapped to its main components
Information systems architecture	• Applications were designed and documented in terms of technical services that support business processes and services. • A technical service was represented that had a schema, used a service operation, and had service realisation diagram. • Technical services were aligned and used by	• This level of architecture was reproduced. It had large monolithic applications that support the delivery of business services • Monolithic systems, integration, and master data management were obstacles that confronted efforts to move to a complete service-orientated environment

Architectural elaboration T4	Dubai Customs	Businesslink
	business processes and services in the business architecture. • Granularity of technical services was considered at the design level to ensure proper reuse. • Services were used to integrate internal systems and external systems such as external payment services. • Use of SOAP protocols, WSDL for services description, and XSD for services schema definitions. • Technical services were mapped to business processes and supporting infrastructure.	• Some applications/services mapping • Documentation was still based on fragmented Visio and Word files
Technology architecture	• SOA infrastructure such as BPEL engine, web services manager, and ESB documented using technology environment, instance, interface, interface messaging, and message structure. • Use of services-related communication protocols such as SOAP and services security protocols such as WS-security • Service repository (integrated into IBM System Architect) that hosted the meta-data of services and related information • Services/infrastructure mapped to show the infrastructure that supported services • Services SLAs were configured and monitored at the application and the infrastructure layers to ensure that the SLAs were met	• This level of architecture was reproduced. • Documentations were still based on fragmented Visio and Word files using the same practices prior to SOA' introduction. •

Architectural elaboration T4	Dubai Customs	Businesslink
EA governance	• EA covered governance aspects regarding demands management (including SOA demands) and demands alignment with strategy and architectural standards. • SOA (and its projects) had its own governance frameworks that were aligned with the overarching EA governance. • EA governs service documentation, service identification, and service delivery • Services were monitored using the orchestration engine • SOA demands were also governed by EA, similar to any other demands against the architectural standards and strategy	• This level of architecture was reproduced. EA governance practices remained as they were prior to the architectural interaction
EA methods and tools	• New SOA-related elements and new relationships were created in the used EA tools (IBM System Architect and the connected view) • New views were created in used EA tools to support services and associated elements • Service identification methods & services were identified using EA products (repository) • EA was integrated with demands/projects, including SOA projects	• This level of architecture was reproduced. EA methods and tools remained as they were prior to the architectural interaction

8.6 Overall Comparison Using the Three Phases of the Model

This section uses the developed theoretical model to understand how EA evolution outcomes (the architectural elaboration phase) was produced in Dubai Customs and Businesslink by retrospectively examining the architectural and interaction phases. It first provides a high-level summary of the three phases and then examines them in detail.

8.6.1 High-level Comparison

As Section 8.5 discusses, the elaboration outcomes of Dubai Customs and Businesslink were different. Using Archer's (1995) morphogenetic theory as a lens, each organisation's architectural elaboration can be explained by looking retrospectively at both the interaction (SOA's introduction) and the architectural conditioning phases. In other words, the architectural conditioning phase conditions the architectural interaction (T2-T3) that generates the evolution outcomes. The architectural interaction (T2-T3) is also influenced by action-formation generative mechanisms. This thesis's findings suggest that, of the two cases, both the conditional generative mechanisms and the action-formation mechanisms influence SOA's integration into EA outcomes.

Figure 8.2 shows a high-level summary of Dubai Customs' three analytical phases. It shows that an enabling context (the architectural conditioning phase) enabled the architectural interaction (the implementation of the Customs Suite), which resulted in EA evolution on all EA five levels.

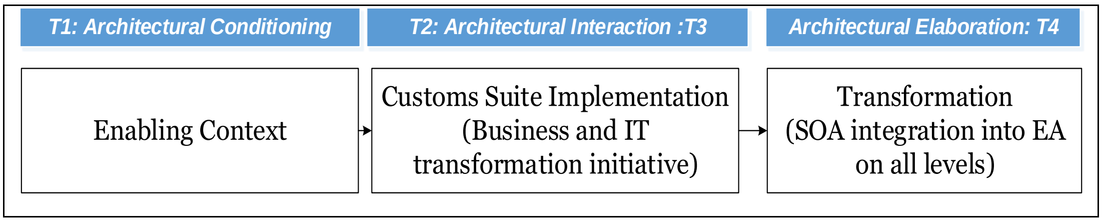


Figure 8.2 High-level view of the morphogenetic cycle of SOA's integration into EA at Dubai Customs

On the other hand, Figure 8.3 shows a high-level summary of Businesslink's EA evolution using the three analytical phases. It shows that a constraining context (architectural conditioning phase) restrained the implementation of the next generation service model (architectural interaction phase), which resulted in a concurrent effort to improve EA (due to the constraining impact) and limited EA evolution.

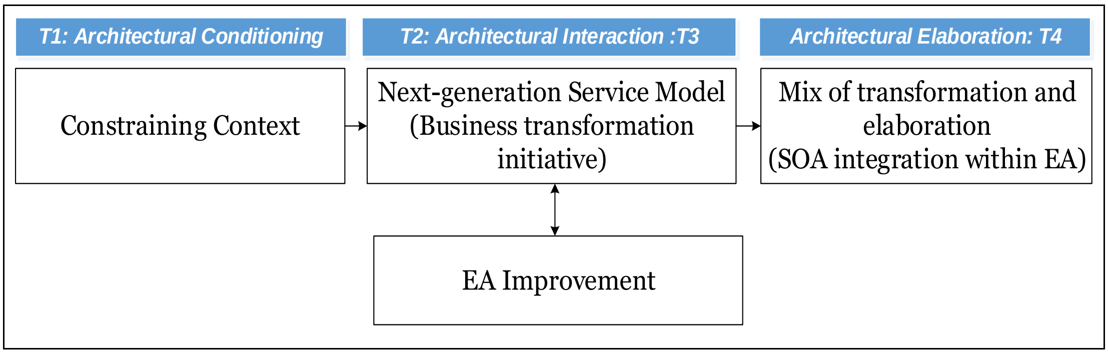


Figure 8.3 High-level view of the morphogenetic cycle of SOA's integration into EA at Businesslink

8.6.2 Detailed Comparison

This section compares the two organisations in detail using the three analytical phases of the theoretical model. First, both organisations presented examples of the conditional generative mechanisms influence SOA's introduction. In Dubai Customs, the three conditional mechanisms created an enabling context for EA evolution. They enabled the EA team to play a major role in SOA's introduction and SOA integration into EA outcomes. EA framework was organisation-wide, and had a well-defined meta-model, a well-defined methodology, and well-maintained deliverables. EA objectives were strategic, business, and IT oriented. Dubai Customs realised EA benefits and appreciated them. In addition, EA practices were mature on many dimensions. For example, a well-established architectural governance facilitated the EA's team engagement with SOA's introduction and, thus, SOA introduction's compliance with EA. The well-established and populated EA repository contributed to SOA's introduction through the use of existing EA models and information to design and implement SOA. The three conditional mechanisms collectively facilitated SOA's integration into EA. Archer (1995) call this context, marked by the three conditional mechanisms in this thesis, an "enabling context".

On the contrary, Businesslink's architectural conditioning constrained the architectural interaction (SOA's introduction). There was no organisation-wide defined EA framework. It was internally developed with no specific meta-model and no defined methodology. EA was IT-oriented and was at very low level of maturity. This conditioning phase is an example of what Archer (1995) calls “a constraining context” of agents. It led to a concurrent effort to improve EA in sync with SOA’s introduction. Upon SOA’s introduction, Businesslink agents realised that they needed to improve EA, which constrained them in their efforts to implement SOA and integrate it into EA. The organisation extended its EA (the IT-oriented EA) by adopting the TOGAF to enable SOA’s implementation and SOA’s integration into EA.

Second, Archer (1995) argued that the conditional phase does not determine the outcomes. She recognises the agent’s orientation and ability to overcome the constraining conditional influence during the action (T2-T3). Businesslink’s case presented an example of Archer’s previous argument. Businesslink’s actors found themselves in a constraining context upon SOA’s introduction. Without considering the action specific characteristics and the agents’ abilities to overcome this constraining context, the integration outcomes would have been negatively determined based on the conditioning phase’s aspects. However, Businesslink’s actors had chosen to improve their EA (the conditioning aspects) in order to improve the integration outcomes, which Archer (1995) describes as an opportunity cost. In other words, the opportunity to implement SOA and to better integrate it with EA had associated costs (elevating the constraining conditional influence) that could have been avoided if the conditioning phase was enabling.

Further, in both cases, the actors had different orientations, interests and resources when they introduced SOA (action-formation mechanisms), which influenced the elaboration outcomes at T4. As presented earlier in Section 8.4, both organisations’ were similar in some aspects and different in others with respect to the six action-formation mechanisms: view of SOA, perceived benefits, and SOA scope, SOA design, SOA governance, and level of business and IT collaboration.

Third, the interplay between the conditional generative mechanisms (conditioning phase T1) and the action-formation mechanisms (architectural interaction T2-T3) resulted in different elaboration outcomes. In Dubai Customs, EA was transformed on the five architectural levels due to the

SOA's introduction (in the form described in Section 8.4) in the enabling context (architectural conditioning phase) (see Figure 8.4).

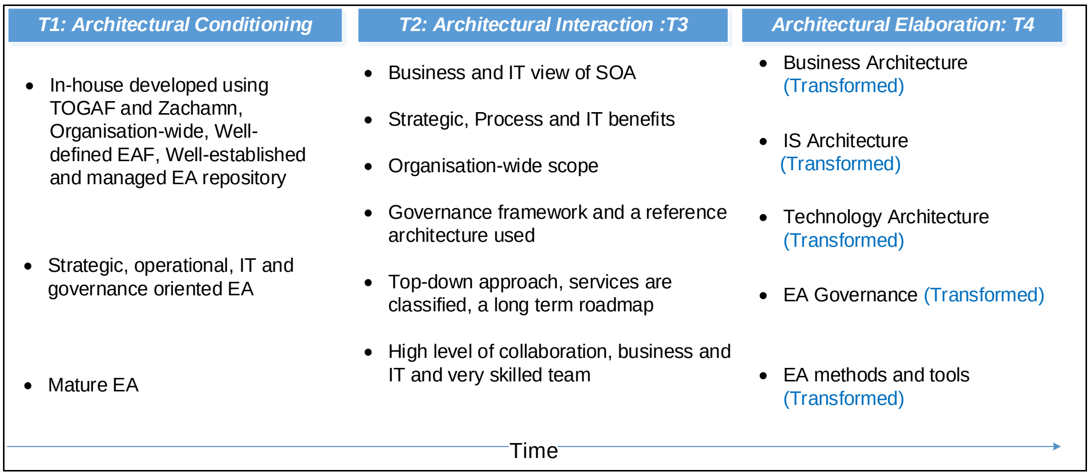


Figure 8.4 The morphogenetic cycle of SOA's integration into EA at Dubai Customs

In Businesslink, there was transformation only of the business architecture and reproduction on the other four levels. These elaboration outcomes were due to the SOA's introduction (in the form described in Section 8.4) and the influence of constraining context (architectural conditioning phase) (see Figure 8.5).

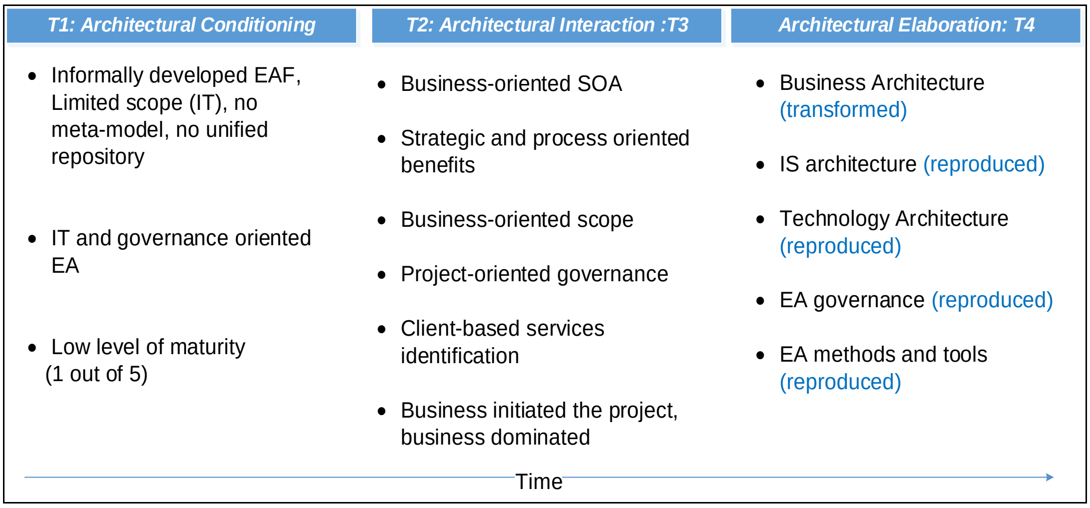


Figure 8.5 The morphogenetic cycle of SOA's integration into EA at Businesslink

8.7 Summary

This chapter compares the findings of the two cases in this thesis. It compares the findings along the three analytical lenses of Archer's (1995) morphogenetic theory.

It first compares the architectural conditioning phase (T1) of both organisations, the phase that preceded SOA's introduction (T2-T3). It describes this phase by comparing the two organisations based on three conditional generative mechanisms: EA framework, EA objectives, and EA maturity. The comparison shows wide differences in terms of EA framework (structure and coverage). It indicates the two organisations' very different levels of maturity of EA during the architectural conditioning phase. It shows the different EA objectives and their use. For example, Dubai Customs adopted EA for strategic, operational, IT, and governance-related benefits, while Businesslink adopted EA to manage their IT on a very high level. These different architectural conditioning phases with respect to pre-existing EA created an enabling context in Dubai Customs and a constraining one in Businesslink. These contexts influence but do not determine the integration outcomes at (T4). Businesslink's constraining context was associated with what Archer (1995) calls an opportunity cost. The opportunity (SOA's introduction) encountered an extra cost of improving the pre-existing EA because of its constraining conditions on SOA's introduction.

Second, it compares the two organisations based on the second phase of Archer's theory; namely, the architectural interaction (SOA's introduction). SOA's introduction is influenced by agents' orientations, interests and resources. In other words, it is influenced by six action-formation generative mechanisms (view of SOA, SOA perceived benefits and SOA scope, SOA design, governance, and business/IT collaboration). The combination of these action-formation generative mechanisms influences SOA introduction and thus the way it's integrated into EA.

The two organisations shared similarities and some differences in the action-formation generative mechanisms that influence SOA's introduction.

First (view of SOA), Dubai Customs actioned the view of SOA at a deeper level than Businesslink did. For Dubai Customs, SOA was a strategy and architectural style that it used to transform its operations. Dubai Customs also used SOA to transform its IT landscape. On the other hand, Businesslink had the same view of SOA compared to Dubai Customs on the business level; SOA was seen as a strategy to transform the organisation into a service-oriented one. Yet, the transformational impact on the organisation's IT levels was minimal except for some individual implementation of cloud-based services.

Second (SOA scope), the two organisations adopted SOA at an organisation-wide level. However, Dubai Customs emphasised business and IT equally, while Businesslink emphasised SOA's business side more than its IT side.

Third (SOA perceived benefits), the two organisations adopted SOA for strategic and process-based benefits. In addition, Dubai Customs also adopted SOA for IT-related benefits.

Fourth (SOA governance), the two organisations adopted a different governance approach. Dubai Customs adopted an SOA governance practice using a specific SOA reference architecture. SOA's introduction was also governed against the wider organisational COBIT governance and EA governance. In contrast, Businesslink's SOA governance was more like project governance and governance around the lifecycle of service development.

Fifth (SOA design), the two organisations had differences in their SOA's design. Dubai Customs used a top-down approach to identify its services, while Businesslink used client requirements for service identification. Dubai Customs also used IBM reference architecture to guide the design of their SOA, while Businesslink adopted an internal design approach. The services classification was also different. Dubai Customs employed types of business services and technical services, while Businesslink used services and service components.

Sixth (Business/IT collaboration), both organisations engaged business. Dubai Customs involved business and IT stakeholders equally. The team to implement SOA was very diverse and skilled (internally and externally). However, Businesslink's SOA was business-driven and business-dominated.

The third analytical level is the architectural elaboration. The outcomes of SOA's integration into EA were different in the two organisations. In Dubai Customs, EA settings were transformed at all five architectural levels.

In Businesslink, EA settings were transformed on the business architecture level only and reproduced on all the other levels. Table 8.9 summarises the two organisations' based on the three analytical phases of the Archer's (1995) morphogenetic theory. These integration outcomes result from SOA's introduction (had similarities and differences in both cases), which was conditioned by the architectural conditioning phase (the influence of its conditional generative mechanisms)

Table 8.9 Contextualisation of the two organisations compared

Analytical phases	Generative mechanisms	Dubai Customs	Businesslink
Architectural conditioning			
	EA framework	In-house developed EA framework based on TOGAF and Zachman	In-house developed EA framework and methods
	EA objectives	Strategic, operational, IT and governance oriented EA	IT and IT governance oriented EA
	EA maturity	High level of maturity (between 3 and 4 out of 5), (see Appendix B for	Low maturity level (1) out of 5
Architectural interaction			
	View of SOA	Enterprise service architecture view of SOA (includes business and IT), SOA as a business strategy and architectural style	Enterprise services architecture view (mostly Business level) not traditional SOA)
	SOA scope	Enterprise-wide with equal emphasis on business and IT	Enterprise wide (with high concentration on the business/service architecture)
	SOA perceived benefits	Driven by strategic, process and IT benefits	Driven by strategic and process benefits
	SOA governance	SOA governance was adopted (using reference architecture) SOA was governed against the wider organisational governance (COBIT) and the architectural governance	SOA's implementation was governed using project-oriented governance Service lifecycle governance was established
	SOA design	Wider SOA design framework was adopted (IBM reference architecture) SOA had a long term roadmap Services were identified using a top-down approach Services were classified into business and	Service identification was driven from clients requirements. Services design and guidelines were established Services have/will have roadmaps (2-3 years) Services were classified into services (have many types) and have components (process,

Analytical phases	Generative mechanisms	Dubai Customs	Businesslink
		technical services	people and IT)
	Business-IT collaboration	Business and IT-driven SOA It was supported by top management Very skilled (internal and external) team implemented SOA Large vendors were involved EA was involved	Very business driven and supported SOA transformational project There was no formal EA team involved
Architectural elaboration (outcomes)			
	Business architecture	Transformed	Transformed
	IS architecture	Transformed	Reproduced
	Technology architecture	Transformed	Reproduced
	EA methods and tools	Transformed	Reproduced
	EA Governance	Transformed	Reproduced

Chapter 9: Discussion

9.1 Introduction

This chapter presents the key insights of the thesis through synthesising the findings from the literature study, interviews, and case studies. It discusses the research findings using the morphogenetic analysis of the EA evolution (and, in particular, the integration of SOA into EA). The findings show that Archer's (1995) theory facilitates a useful analysis of EA evolution to accommodate new emerging business and IT capabilities.

The chapter progresses as follows. Section 9.2 summarises the research topic and its research questions. Section 9.3 discusses the architectural conditioning phase. Section 9.4 discuses the second analytical phase, the architectural interaction (SOA introduction). In particular, it discusses the influence of the six action-formation generative mechanisms related to SOA introduction. Section 9.5 discuses the architectural elaboration due to SOA introduction (either transformation or reproduction of the pre-existing architectural settings) on five architectural levels. Section 9.6 summarises the three analytical phases using the developed theoretical model, while Section 9.7 summarises the chapter.

9.2 EA Evolution

The wider subject of this thesis is EA evolution. This thesis specifically focuses on the introduction of service-oriented architecture (SOA) into organisations as one exemplary trigger of EA evolution. Thus, throughout the thesis, SOA's integration into EA is used to represent EA evolution after SOA's introduction.

The following section syntheses the key empirical findings from the interviews and case studies (Chapters 5, 6, 7 and 8) in order to improve the literature-based insights presented in Chapters 2 and 4 to answer this thesis research questions.

The literature review presented in Chapter 2 outlines how the outcomes of SOA's integration into EA vary. To reiterate, that situation is described by the following points: (1) there is no clear unanimity on an integration strategy of services into EA (Traverson, 2008), (2) EA frameworks and languages have not adequately addressed SOA elements and viewpoints (Postina, et al., 2010), and (3) more studies are needed in order to comprehend SOA's impact on EA (Dico, 2012; Kistasamy, et al., 2012; Viering, et al., 2009). The review shows that EA evolution due to introducing new business or IT capabilities into an organisation is undeveloped research area. Specifically, it shows that EA evolution after introducing SOA is underemphasised. This situation is articulated well in a quote from an EA Consultant describing the need for EA-driven SOA implementations [I-1]:

> *There's an awful lot of people who think it's actually simplistic and don't understand the complexities associated with developing an SOA architecture... there's a massive recognition out there about the fact that SOA is failing... but also is starting to emerge in the last year or so a recognition that enterprise architecture is the solution.*

Thus, this thesis is dedicated to empirically understanding EA evolution and to explaining EA evolution outcomes through examining SOA's integration into EA after SOA's introduction in organisations. As Chapter one presents, this thesis has two research questions:

RQ1: *How does EA evolve as a result of the introduction of SOA?*

RQ2: *What are the factors that influence EA evolution as a result of the introduction of SOA?*

In the light of the research objectives and research questions, this thesis uses Archer's (1995) morphogenetic theory as an analytical lens to improve understanding of EA evolution. In Archer's theory, every morphogenetic cycle distinguishes three analytical phases, consisting of (1) a particular structure (here: EA), which conditions but does not determine (2) architectural interaction (here: SOA introduction); (2) in turn leads to (3), architectural elaboration (here: EA evolution outcomes).

Archer's (1995) morphogenetic theory underpins the development of the a-priori model of this thesis (Chapter 4). The developed a-priori model was used to provide early insights to answer the research questions with the literature review's findings. The morphogenetic theory is used to re-describe the components of EA evolution following the "theoretical re-description stage" of the Danermark et al.'s (2002) methodological framework (Chapter 3). The research questions *were answered* iteratively using different strategies (literature review, theory, interviews, and case studies) (See Chapter 3 and Table 9.1).

Table 9.1 Triangulation of methodological approaches

Sources of data	Their use in this thesis (triangulation)
Literature review (Chapter 2) and the a-priori model's development (Chapter 4)	Used to understand the research context, investigate SOA's integration into EA (macro-level) and identify possible conditional mechanisms, action-formation mechanisms, and integration outcomes (micro-level). The findings were re-described using Archer's (1995) morphogenetic theory. SOA's integration into EA was scoped, and the findings were represented using Archer's morphogenetic theory along the three analytical phases.
Explorative interviews (Chapter 5)	The explorative interview phase (20 participants) was used to refine and extend the a-priori model developed in the previous phase. As a result, two more conditional generative mechanisms (EA framework and EA objectives), three more action-formation mechanisms (SOA governance, SOA design, and business/IT collaboration), and two levels of the integration levels (EA governance and EA methods and tools) were identified.
Two case studies (Chapters 6 and 7)	The developed theoretical model in the previous phases was contextualised. In other words, it was further explored in two contexts (case studies) to explore the interplay between the generative mechanisms and the observed evolution outcomes. The results support the effects of the generative mechanisms and their interplay on the outcomes in different contexts.
Cross case analysis (Chapter 8)	The two cases were compared in order to understand similarities and differences of EA evolution process (the three phases) and outcomes

The next sections discuss EA evolution using the three analytical phases of the theoretical model (see Figure 9.1).

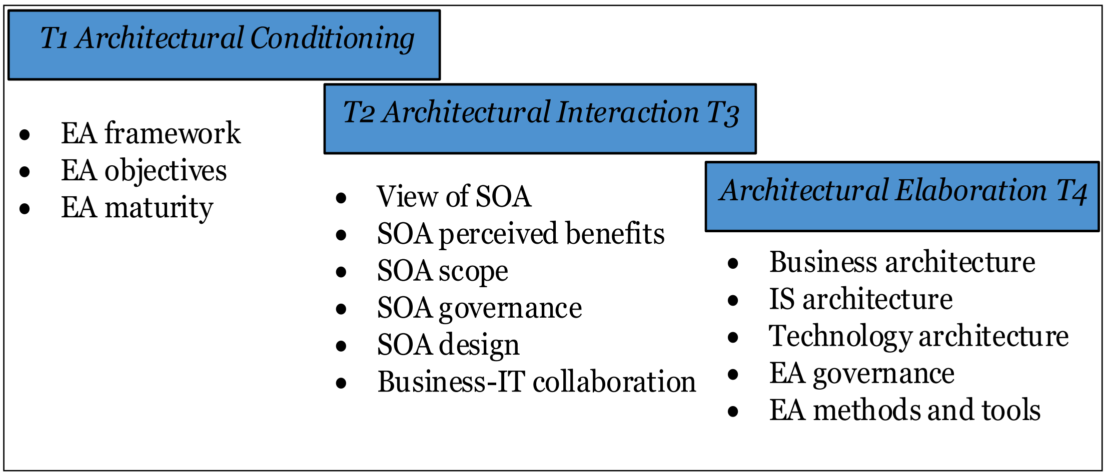


Figure 9.1 This thesis's theoretical model

9.3 Architectural Conditioning (T1)

This section concentrates on the architectural conditioning phase and how its conditional generative mechanisms condition the subsequent architectural interaction phase, which leads to the architectural elaboration. This phase is considered in isolation following Archer's (1995) morphogenetic theory. In reality, however, the architectural conditioning phase that has been isolated co-exists with a variety of other contextual factors (out of this thesis's scope), which could influence the architectural interaction phase.

As Chapter 4 presents, the basic argument for considering the architectural conditioning phase that precedes the action (here: SOA's introduction) is that EA evolution cannot be fully explained without reference to antecedent architectural conditioning (Archer, 1995). Archer's work is based on the notion of emergent properties first developed by Bhaskar (1975), who argues that, while a given structure is obviously the product of human actions, it is not necessarily the product of those "here-and-present" agents. In other words, the previous activities of agents create structures that then both constrain and enable actors in the next round of action (Mutch, Delbridge, & Ventresca, 2006).

This thesis presumes this argument to be true for integrating SOA into EA. For instance, actors who integrate SOA into EA start the integration in a context (pre-existing EA), which enables certain outcomes and makes others difficult. This context is described in terms of three generative mechanisms that, together, conditionally influence EA evolution.

Chapter 2 identifies one generative mechanism based on the literature: EA maturity. The empirical data supports EA maturity as an important conditional generative mechanism. They led to the identification of two more conditional generative mechanisms: EA framework and EA objectives. Table 9.2 summarises the key insights related to these generative and Sections (9.3.1-9.3.4) discuss them. These generative mechanisms may have interrelationships between them, and one may affect others; however, this level of analysis is considered beyond the scope of this thesis because, as Hedström and Ylikoski (2010, p. 52) explain:

> *For a mechanism to be explanatory it is not required that the entities, properties, and activities that it appeals to are themselves explained. The only requirement is that such entities, properties, and activities really exist; their explanation is a separate question.*

Table 9.2 Summary of findings related to the architectural conditioning phase

Generative mechanism	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Chapter 8: cross-case analysis	Implications for this thesis
EA framework (empirically identified)	The interview analysis shows diverse EA frameworks (structure, scope, focus, and meta-models) and the effects of previous cycles of change on these EA frameworks have created a conditional influence on SOA's integration into EA.	Well-established EA framework, formally defined and used, employed well-defined methodology using well-known EA frameworks (enabler)	Internally developed, informally defined, not based on existing EA frameworks, no defined EA methodology, and covered the IT domain only (constraint)	Supported conditional mechanism. EA framework (has different characteristics and is often shaped by previous morphogenetic cycles). created an enabling context in Dubai Customs and constraining context in Businesslink for SOA's integration into EA.	Supported conditional generative mechanism added to the theoretical model
EA objectives (empirically identified)	The interview analysis shows that EAs were adopted for different purposes and classified into: strategic, operational, IT, and governance oriented. These different objectives were suggested as a conditional influence on further EA-related actions such as SOA's integration into EA.	EA was established to achieve strategic, operational, IT, and governance objectives (enabler)	EA was established to achieve IT and IT governance objectives only (constraint)	Supported conditional mechanism. EA objectives created enabling context (need to be sustained in Dubai Customs) and a constraining context in Businesslink (needs to be improved)	Supported conditional generative mechanism added to the research model
EA maturity (literature-based)	Evidence from about 14 interviews emphasise the role of mature EA for better and evolving	Mature EA practices (enabler of SOA integration)	Immature EA practices (constraint of SOA integration)	Strongly supported mechanism. The cross-case analysis showed the very different conditional	Well-supported conditional generative mechanism

Generative mechanism	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Chapter 8: cross-case analysis	Implications for this thesis
	EA practices in general and SOA's integration in particular			influence of EA maturity in the two cases (enabling in Dubai Customs and constraining in Businesslink)	

9.3.1 EA Framework

This conditional generative mechanism concerns the influence of the EA framework on SOA integration. It was inductively identified from the interview analysis in Chapter 5. EA frameworks have different structures and scopes. They may or may not include a defined meta-model, scope, and methodology. Implemented EA frameworks have often been changed from their original shape by previous cycles of changes prior to introducing SOA. This thesis proposes that EA frameworks characteristics have a conditional influence (enabling or constraining) for the next activities related to EA and specifically EA evolution.

Chapter 5 reported on the various EA frameworks with different characteristics. Some of these frameworks were developed using well-known EA frameworks, others were modified, and some were in-house developed frameworks. The scope of the use of these frameworks also varied (partial, light, and full adoption). The interview data shows that these EA frameworks were shaped by previous cycles of change. EA is often implemented in organisations prior to SOA's introduction. Most of these frameworks were changed, adapted, and modified. They create a conditional context for EA evolution (Chapter 5) by either being an enabler or a constraint. Two interviewees noted that their organisation's old EA framework was replaced by a newer one because it created a constraining context for the organisation (e.g., not supporting the decision-making process, and/or becoming out-dated). Furthermore, some participants [I-1 and I-20] state that their organisations existing frameworks do not provide the necessary artefacts to better integrate SOA.

Moreover, the conditional influence of this generative mechanism was supported in the case studies. Dubai Customs' (Chapter 6) EA framework had an enabling conditional influence on SOA's integration into EA. The EA framework was built on well-known EA frameworks. It had a comprehensive scope, a well-defined meta-model, and a well-defined structure. On the other hand, Businesslink's (Chapter 7) EA framework constrained SOA's integration. It was internally developed in the organisation's IT department, had no meta-model, included only IT domain artefacts, and had not been managed in accordance with the organisational evolution (changes).

In summary, the EA framework becomes a conditional generative mechanism for the next iteration of action related to EA. This thesis emphasises the significance of having an organisational-wide, a well-defined EA framework, and a comprehensive meta-model to create an enabling context of EA-related activities such as EA evolution. These findings support recent studies proposing that EA framework characteristics influence EA implementations (Bui, 2012). Having an organisationally aligned EA framework and a well-defined meta-model increase the efficiency and effectiveness of EA practices (Lange, 2012) and enables EA evolution.

9.3.2 EA Objectives

The second conditional generative mechanism, EA objectives, was inductively identified from the interview analysis in Chapter 5. EA objectives are classified into strategic, operational, IT, and governance. The interview findings show the diversity of EA objectives that have driven EA implementations. For example, fifteen interviewees reported adopting EA to achieve strategic benefits, while three interviewees reported adopting EA to realise IT-oriented benefits. EA objectives were supported as a conditional generative mechanism in the two case studies (chapters 6 and 7); it created an enabling context in Dubai Customs and a constraining context in Businesslink prior to SOA introduction.

The importance of this conditional generative mechanism stems from the fact that EA is adopted for different objectives. EA could be adopted for one or more of four classes of objectives: strategic, operational, IT, and governance. These objectives drive the way an organisation implements and use EA. As a result, these objectives become a conditional factor for the next round of EA activities (e.g., SOA's integration into EA— see Chapters 6 and 7).

The case studies provided two insights regarding the conditional influence of EA objectives on EA evolution. The first insight is that the comprehensiveness of EA objectives on strategic, operational, IT, and governance aspects at Dubai Customs enabled EA evolution. Such comprehensive objectives stemmed from a comprehensive EA initiative, close engagement of business and IT, and a long term vision of EA. Most of EA's objectives were realised in Dubai Customs prior to SOA's introduction, which had demonstrated the value of EA and created an enabling context (Archer, 1995) for EA evolution. As such, Dubai Customs ensured that its EA will evolve with any organisational change.

The second insight is that Businesslink's limited EA objectives contributed to a constraining context that didn't support EA evolution. EA was predominantly focused on IT and IT governance objectives. EA value was not visible to the organisation and thus its integration into organisational activities was missing. Therefore, the organisation extended its EA by adopting a TOGAF-based EA encompassing comprehensive strategic and operational objectives. EA became *"an essential strategic activity required for the successful planning and delivery of Businesslink's current and forecast obligations"* [Businesslink D-5]. *"So at one level they want to use it for the strategic side of things. At a lower level they want to understand the impact upon components of the services that we provide"* [Businesslink P-6].

Thus, one can argue that, based on the interviews and the two case studies, EA objectives have a conditional influence on EA evolution (enabling in Dubai Customs and constraining in Businesslink). As such, comprehensive EA objectives should be emphasised and maintained via business and IT stakeholders' participation, a long-term EA vision, and the establishment of a common understanding about EA (e.g., Dubai Customs). If that is missing, EA is more likely to be isolated from organisational development and thus do not evolve. These findings support recent studies that suggest organisations follow different EA development approaches based on the architecture objectives, which, in turn, may affect EA activities in later stages (Haki, et al., 2012; Lapalme, 2012).

9.3.3 EA Maturity

The findings from Chapter 2, which were used to build the a-priori model in Chapter 4, suggest that the maturity level of EA has a conditional influence on EA evolution. A low maturity level leads to difficulties in establishing an EA function that is effectively integrated into existing organisational practices, and to difficulties in motivating effective collaboration between architects and other stakeholders. As a consequence, a fragmented and poorly integrated EA function typically fails to achieve expectations (Raadt & Vliet, 2008). This argument applies to SOA integration, too. Mature EA is an enabler of SOA implementation (O'Brien, 2009), improves SOA implementation's alignment with organisational objectives (Brooks, 2009), and facilitates SOA integration (Postina, et al., 2010).

The empirical findings further support the influence of EA maturity on SOA's integration into EA. The interview phase emphasises the significance of mature EA settings for advantageous EA practices in general and for EA sustainability and SOA's integration into EA (Chapter 5) in particular. Moreover, the case studies provide an intimate understanding of EA maturity's impact as a conditional generative mechanism. Dubai Customs' high level of EA maturity enabled it to comprehensively integrate SOA into EA, while Businesslink's low level of maturity restrained its EA evolution efforts.

Chapters 6 and 7 provide insights about EA maturity's influence on the subsequent EA-related activities. At Dubai Customs, EA was mature, well managed, and *"the governance [was] ensured during the whole EA process"* [Dubai Customs P-3]. Such a high level of EA maturity was achieved through defined documentation, a well-established repository, continuous governance, business support, and a diverse and skilled team. Thus, Dubai Customs' mature practices efficiently enabled SOA's integration into EA. On the contrary, Businesslink's low-level EA maturity, the result of the opposite conditions to those specified above, restricted EA evolution. Recent literature supports EA maturity's influence on EA activities in general. For example, mature EA is expected to lead to greater business-IT alignment and facilitates the realisation of business objectives (Bradley, et al., 2012). EA maturity has a profound influence on EA's overall effectiveness (Gartner, 2012b; Lagerstrom, et al., 2011; Roth, et al., 2013).

9.3.4 Summary

Reflecting on the findings of this section, the architectural conditioning phase has a causal influence on SOA's integration into EA through the effects of the three generative mechanisms. They either enable or constrain SOA's integration into EA. The two case studies illustrate different architectural conditioning influences, which had a different conditional impact (enabling in Dubai Customs and constraining in Businesslink) on SOA's integration into EA.

In summary, the architectural conditioning phase influences SOA's integration into EA based on the actualisations (values) of the identified three generative mechanisms. Nevertheless, treating these conditional generative mechanisms as the primary factors in EA evolution process is not suitable according to the findings and Archer's (1995) argument that states neither the structure (EA) nor the action alone determines the outcomes. Thus, the next section discusses SOA's introduction as an action that triggers EA evolution.

9.4 Architectural Interaction (T2 SOA's Introduction T3)

The previous section concentrates on the architectural conditioning phase and how its generative mechanisms condition the architectural interaction (SOA introduction) which leads to the architectural elaboration (EA evolution outcomes). The previous phase's impact is conditional. Archer (1995) acknowledges the ability of actors who take the action (e.g., introduce SOA) and their capacity for innovative responses to challenge the conditional influence.

This section focuses on the interaction phase and its (action-formation) generative mechanisms. As Chapter 4 discusses, this type of mechanism describes how actions are influenced by agents' orientations, interests, and resources.

The six action-formation mechanisms are presented here to explore their causal impact on SOA's integration into EA through their combined effects on SOA introduction. The interplay between the architectural conditioning (T1) and the architectural interaction (T2-T3) occurs in a morphogenetic cycle (SOA's integration into EA), which defines how the architectural changes occur (Archer, 1995; Cuellar, 2010). In other words, the integration outcomes occur due to the interplay between the two sets of generative mechanisms related respectively to the architectural conditioning (T1) and the architectural interaction (T2-T3) phases. Table 9.3 summarises the key insights in relation to each generative mechanism.

Table 9.3 Summary of findings related to the architectural interaction phase

Generative mechanism	Chapter 5: interviews	Chapter 6: Dubai Customs	Chapter 7: Businesslink	Implications for this thesis
View of SOA (literature based)	Evidence from the interviews support the diversity of perspectives of SOA and its potential impact on SOA introduction	Business- and IT-oriented perspective of SOA	Business-oriented view of SOA	Supported action-formation generative mechanism
SOA perceived benefits (literature based)	Evidence from the interviews support the diversity of SOA benefits and its potential impact on SOA introduction	SOA was implemented to achieve strategy, process and IT benefits	SOA was implemented to achieve strategy and process benefits	Supported action-formation generative mechanism
SOA scope (literature based)	Evidence from the interviews support the different scopes of SOA implementations and its potential impact on SOA introduction	Enterprise-wide implementations on both the business and IT levels	Business-oriented implementation, high emphasis on the business side of the organisation and less on the IT	Supported action-formation generative mechanism
SOA governance (empirically identified)	Emerged from eight interviews as another action mechanism that influences SOA introduction	SOA introduction was governed against SOA reference architecture, EA and wider organisational (projects) governance practices	SOA introduction was governed using traditional project governance practices	Emerged action-formation generative mechanism
SOA design (empirically identified)	Emerged from fifteen interviews as another action mechanism that influences SOA introduction and	SOA design was considered on many dimensions (top-down services identifications, services classified, reference architecture was used, a long-term road map was developed and EA repository	SOA design was considered on many dimensions (client-based service identification, services classified, each service has a service design reference, every service has a roadmap and static repository was used)	Emerged action-formation generative mechanism

Generative mechanism	Chapter 5: interviews	Chapter 6: Dubai Customs	Chapter 7: Businesslink	Implications for this thesis
		was used)		
Business and IT collaboration (empirically identified)	Emerged from thirteen interviews as another action mechanism that influences SOA introduction	High level of business and IT collaboration. Key stakeholders and external vendors were involved. Highly skilled and diverse team drove SOA implementations	Dominated business project and supported by IT. Internal and external stakeholders were involved	Emerged action-formation generative mechanism

9.4.1 View of SOA

The first action-formation generative mechanism is the view of SOA. There are widely differing perspectives of SOA that are suggested to influence its introduction in organisations. Here, these views are classified into five perspectives: fine-grained service components, emerged software architecture, business process support, enterprise service architecture, and adaptive architecture (Hirschheim, et al., 2010; Welke, et al., 2011).

SOA is introduced depending on how it is perceived (Hirschheim, et al., 2010; Lee, et al., 2010; Stein, et al., 2008; Welke, et al., 2011), and Archer (1995) argues that actions are influenced by perceptions. The interview data further supports the diversity of how SOA is perceived and consequently implemented. Nevertheless, no participant reported the last perspective of SOA, the adaptive architecture (see chapter 5). The interview findings suggest that an adopted view of SOA shapes SOA introduction. Some participants argued that the technical views are considered undeveloped perspectives of SOA and adopting SOA from such a perspective does not represent SOA's wider aspects nor attain its ultimate potential.

The case study findings further support the diversity of SOA perspectives and show a conceivable link between the view of SOA and its implementations (Chapters 6 and 7). Both cases have a different perspective of SOA. *SOA at Dubai Customs encompasses the business and IT*. It was seen as a strategy and an architectural style. On the other hand, *SOA at Businesslink was a business strategy to re-design the organisation's services only*. Businesslink placed less emphasis on traditional SOA and more emphasis on the move to a service-oriented organisation.

The view of SOA is a key action-formation mechanism that influences SOA introduction. Introducing SOA is often complex and involves many actors from different areas of an organisation. Thus, SOA introduction requires a consistent and aligned perspective of SOA across the organisation introducing it (Koumaditis, Themistocleous, & Da Cunha, 2013).

9.4.2 SOA Perceived Benefits

The second action-formation mechanism is SOA perceived benefits. The literature review suggests that SOA is adopted for various benefits at different levels (strategy, process, and IT) (Becker, et al., 2009; Mueller, et al., 2007). Some SOA implementations pay attention to these benefits at all levels and others are limited to achieve SOA benefits at one or two levels. Based on the literature review, it seems that these various benefits affect SOA introduction (Joachim, et al., 2009; Welke, et al., 2011). The interview findings support the diversity of SOA's perceived benefits and its potential influence on SOA introduction.

The case study findings also support the diversity of the perceived benefits of SOA and its influence on SOA introduction. For example, Dubai Customs drove its SOA to achieve strategic, process, and IT benefits. The resultant SOA implementation equally affected the business and technology sides of the organisation. On the other hand, Businesslink drove its SOA to achieve strategic and process-oriented benefits, and thus most of the implementation activities were on the business side of the organisation.

Based on these findings, it seems that SOA perceived benefits is a relevant action-formation generative mechanism that influences, in combination with other action-formation generative mechanisms, SOA introduction. Thus, it is important for organisation to comprehensively understand SOA's benefits (strategy, process, and IT) and, based on that, set up in advance well-defined perceived benefits of SOA to better drive SOA introduction (Koumaditis, et al., 2013; Lee, et al., 2010).

9.4.3 SOA Scope

The third action-formation generative mechanism is SOA scope. Chapter 2 suggests that there are three different scoping options chosen to implement SOA (Campbell & Mohun, 2007). Each scope has certain objectives, requires different resources, skills, and methods, and has different organisational impacts. The interviews support the varying scoping options and suggest that these scoping options influence SOA introduction. Participants report different scopes (small projects, portfolio level, and organisation wide) that influence the way SOA is introduced. They also report that a higher scoping achieves wider benefits but requires a longer commitment. On the other hand, project-based SOA implementations are usually fragmented and are conducted without being aligned to EA.

The case studies also show the different scoping options for SOA introduction. For example, SOA introduction was organisation wide in Dubai Customs, and was mostly business-oriented in Businesslink case. As a consequence, Dubai Customs' implemented SOA affected both its business and IT, while Businesslink's implemented SOA had a very minimal impact on the IT side of the organisation.

9.4.4 SOA Governance

The fourth action-formation generative mechanism is SOA governance. It was inductively identified from the interview analysis in Chapter 5. It is defined in this thesis as the planning of a SOA's direction, the management of services lifecycle, and the establishment of standards, policies, roles, and responsibilities related to SOA introduction.

The findings reveal insights about whether SOA governance influences SOA introduction. First, the interview findings suggest that well-established SOA governance is important for SOA introduction. It keeps SOA implementation on track and avoids inconsistencies. Second, SOA introduction needs a well-defined identification of roles and responsibilities and the establishment of a governance committee to monitor SOA implementation. Third, there are varied practices of SOA governance that impact SOA introduction (see Chapter 5). For example, the use or lack of SOA standards and policies, the establishment or lack of SOA specific roles and designated responsibilities, and the use or deficiency of an established SOA reference architecture all impact SOA introduction.

Furthermore, the case studies reveal different approaches to SOA governance. Dubai Customs' SOA introduction was governed using SOA-specific reference architecture, EA governance, project governance, and the wider organisational governance (COBIT). On the other hand, Businesslink did not employ a specific-SOA governance framework nor did it align SOA with EA governance. It was governed using traditional project governance practices, possibly due to (1) the lack of established EA governance practices prior to and during SOA's implementation, and (2) the limited focus on the technology aspects of SOA's implementation.

Such diverse SOA governance practices, collectively with other action-generative mechanisms, influence SOA introduction. The way an organisation controls SOA introduction through the service lifecycle management, the establishment of SOA-specific roles, the monitoring of the progress of SOA, and the alignment of SOA governance with existing governance practices influences SOA introduction (Joachim, Beimborn, & Weitzel, 2013; Koumaditis, et al., 2013).

9.4.5 SOA Design

The fifth action-formation generative mechanism is SOA design. It was inductively identified from the interview analysis in Chapter 5. It concerns the way SOA is designed; that is, its reference architecture, roadmaps, service identification methodology, and services classifications. The interview data suggest that SOA design practices affect SOA introduction.

The two cases show that SOA design influences SOA introduction. For example, Dubai Customs used a top-down service identification approach and classified its services as either business or technical. The services were stored in a repository (IBM System Architect) with other architectural elements, and SOA's roadmap was based on a long-term initiative. On the other hand, Businesslink's services were determined by its clients' requirements. Each service had or was going to have service components (processes, people, and IT), and each was classified under a domain such as HR or finance services. Services were kept in a static (PDF) file used as a service portfolio that had only service-related information. The roadmaps were partially completed based on each developed service. Thus, these SOA design practices influenced each organisation's SOA introduction.

Such variations in SOA design are contributed to (1) the lack of empirically validated guidelines and/or (2) limited practitioners' experiences of best SOA design that could lead to successful SOA implementation (Aier & Gleichauf, 2009).

9.4.6 Business and IT Collaboration

The last action-formation generative mechanism is business and IT collaboration. It was inductively identified from the interview analysis in Chapter 5. It refers to the level of business support, the SOA team (business, IT, or mixed), and its members' skills that may influence SOA introduction. The interview findings suggest that the level of business support, SOA team settings and the team members' skills influence the way SOA is introduced.

The actualisations of this generative mechanism varied in the two cases studies, which supports the influence of this generative mechanism on SOA introduction. Dubai Customs had a high level of business and IT collaboration, while Businesslink had a very strong business-driven SOA. Dubai Customs had a highly skilled team and involved external vendors in its SOA introduction. Businesslink had a very business-oriented internal team driving its SOA introduction.

As Koumaditis, et al. (2013) argue, the insights of this generative mechanism highlight the importance of engaging business and IT and of having mature and skilled teams when introducing SOA.

9.4.7 Summary

It is apparent that the architectural interaction phase (SOA introduction) is influenced by many action-formation generative mechanisms. This finding supports the theoretical arguments of Chapter 4, of Archer (1995), and of Hedström and Ylikoski (2010) and Cuellar (2010) about the influence that a combination of interests, orientations, and resources can have on the action (SOA introduction).

That is, SOA introduction is influenced by the six action-formation mechanisms (agents' orientations, interests, and resources). The action-formation generative mechanisms collectively (acknowledging that one generative mechanism may counterbalances others) shape the way SOA is introduced. This conclusion is supported by Mutch's (2010) argument that IS implementations could be configured in different ways based on different factors to produce very different outcomes for organisational arrangements. The way SOA is introduced is the result of the different configuration (actualisations) of these generative mechanisms in different contexts. Thus, when introducing SOA, organisations need to consider the implications of the combination of these generative mechanisms on (1) SOA implementation and (2) the organisation and its enterprise architecture in general. Further, these findings shed light on the often overlooked organisational and governance aspects of SOA implementations (Joachim, et al., 2013).

The variations of the action (SOA introduction) are discussed here to understand its impact on SOA's introduction and consequently on EA evolution. The second level of generative mechanisms analysis (the inter-relationships between these generative mechanisms and their influence on each other) is outside this thesis scope. In other words, the understanding of the interrelationships between these generative mechanisms and their influence on each other requires further analysis.

The next section addresses the third analytical phase of EA evolution, the architectural elaboration that results from the action (SOA introduction) in a certain context (architectural conditioning).

9.5 Architectural Elaboration (T4)

This section deals with the final phase of EA evolution. It deals with the architectural elaboration that results from the architectural interaction discussed in Section 9.4. The elaboration outcomes are either transformation or reproduction of the pre-existing EA on five levels.

According to Archer (1995), the point of examining any morphogenetic cycle is to provide an analytical perspective on the emergence of outcomes under investigation. The literature review findings suggest that SOA's integration into EA could happen on three EA levels: business architecture, information systems architecture, and technology architecture. The interview findings support EA evolution on the three levels and suggest two more meta-levels of architectural elaboration: EA governance and EA methods and tools. The pre-existing EA (prior to SOA introduction) is transformed or reproduced on one or many of these five levels. The key insights in relation to each architectural elaboration level are summarised in Table 9.4 and discussed in Sections 9.5.1 to 9.5.6 and summarised in Table 9.4.

Table 9.4 Summary of findings related to the architectural elaboration phase

Architectural elaboration level	Chapter 2: literature review	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Implications for this thesis
Business Architecture (literature based)	Chapter 2 offers some examples of SOA's integration into the business architecture only or in combination with the IS and technology architectures. Yet, these examples have different SOA elements integrated at this level	Evidence from about 7 interviews (out of 20) reported the explicit integration of SOA into business architecture (addition of SOA elements and their relationships with existing EA elements), while 13 interviews did not explicitly report this level of integration	Transformation: SOA was integrated with the business architecture (formally called process architecture). SOA-related elements such as business services, their descriptions, channels, and owners were integrated into the business architecture	Transformation: Pre-existing EA was technology-oriented and lacks the business focus which constrained agents' effort to integrate SOA with the business architecture. Thus, EA was extended using the TOGAF. Service-oriented business architecture (service architecture) was developed. It had services groups and services, but there was no explicit meta-model	Supported level of SOA architectural elaboration.
Information Systems Architecture (literature based)	Chapter 2 provides some examples of SOA's integration into the IS architecture only or in combination with the Business and technology	Evidence from ten interviews (out of 20), the previous seven interviews, and three more reported the explicit integration of SOA into IS architecture	Transformation: the IS architecture integrated SOA-related elements such as technical services, service operation, and service realisation. These elements were added to	Reproduction: the IS architecture was largely reproduced. No SOA elements were integrated due to (1) the lack of an explicit meta-model, (2) the reproduced use of	Supported level of architectural elaboration

Architectural elaboration level	Chapter 2: literature review	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Implications for this thesis
	architectures. These approaches have integrated dissimilar SOA elements into the information systems architecture	(addition of SOA elements and their relationships with existing IS elements), while 10 interviews did not explicitly report this level of integration	the meta-model and integrated with the other architectural elements	fragmented documentations, and (3) the business focus of SOA's introduction	
Technology Architecture (literature based)	The literature review (Chapter 2) suggested cases where SOA is integrated into the technology architecture only or in combination with the two levels presented earlier	The empirical findings support SOA's integration into the technology architecture alone (2 interviews) and with the other architectural levels (10 interviews). There was also one interview that reported that SOA was integrated with the business and IS architectures and had not been integrated with the technology architecture	Transformation: Dubai Customs' technology architecture was transformed. This architecture had integrated and supported SOA elements. It also supported the mapping between services and their supporting infrastructure	Reproduction: No reported changes to the technology architecture. It was reproduced as it was prior to SOA implementation	Supported level of architectural elaboration
EA governance (empirically		Mixed opinions about SOA governance	Transformation: EA and SOA governance	Reproduction: EA and SOA governance were	Supported level of elaboration.

Architectural elaboration level	Chapter 2: literature review	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Implications for this thesis
identified)		integration into EA governance. Six participants (out of 20) explicitly reported that SOA governance needs to be integrated with EA governance, yet with different perspectives on the level of integration required	were integrated. This could have been enabled by the enabling conditioning mechanisms and the actualisations (values) of the action-generative mechanisms. In particular, SOA governance where SOA introduction was well governed and aligned with EA and organisational governance	not integrated. This could have been restricted by the frustrating conditional mechanisms and the lack of very explicit traditional SOA governance framework (combined with other action-formation mechanisms)	Transformation in the first case was enabled by the mature governance practices and the agents' view of the need to integrate both SOA and EA practices. In the second case, reproduction was the result due to the lack of mature EA governance prior to SOA's introduction, despite acknowledgment of the importance of SOA governance's integration into EA governance
EA methods and tools (empirically identified)		Five participants (out of 20) explicitly reported SOA and EA methods and tools are integrated	Transformation: SOA and EA have overlapping methods and tools. That could have been enabled by the enabling conditioning phase and the (defined SOA methodology and	Reproduction: EA methods and tools were reproduced, that could be due to the conditional influence (low maturity, no defined frameworks, no defined methods and no specific EA tool) and	Mixed opinions, could be due to the level of conceptualisations of EA. In some cases, EA was comprehensive and included aspects of solution development and project management, which

Architectural elaboration level	Chapter 2: literature review	Chapter 5: interviews	Chapter 6: Dubai Customs case	Chapter 7: Businesslink case	Implications for this thesis
			design aspects) combined with other action mechanisms	the combined influence of SOA's introduction with no specific SOA methodology or tools	explains the need to change EA methods and tools due to SOA introduction. Whereas in other cases, EA was a very high-level representation of the organisation and thus did not include nor require the changes to the solution and project management dimensions

9.5.1 Business Architecture

The first level of architectural elaboration because of SOA introduction is the transformation or reproduction of the pre-existing business architecture. The business architectural transformation or reproduction is determined based on the integration of SOA-relevant elements, such as business services, service channels, SOA vision, drivers, SLAs, and QoS, into business architecture.

Chapter 2 offers some examples of EA transformation on this level, which are used as a basis to build the a-priori model in Chapter 4. Yet, these examples have different SOA elements integrated at this level. Moreover, the transformation of this architectural level is often accompanied by a transformation of the IS and technology architectures. The interviews findings show examples of business architecture transformation where some participants reported SOA integration into EA. Similar to the literature review, the interview findings show the diversity of the transformation details on this architectural level (different SOA elements were integrated). Transforming the business architecture is also seen in the two case studies but with different emphasis on the integrated SOA elements.

9.5.2 Information Systems Architecture

The second level of architectural elaboration is the transformation or reproduction of the pre-existing information systems architecture. IS architectural transformation or reproduction is determined based on the integration or lack of integration of SOA elements such as application services, service descriptions, and SLAs into information systems architecture.

The literature review provides some examples of EA transformation on this level, which are used as a basis to build the a-priori model in Chapter 4. Yet, these approaches have integrated dissimilar SOA elements into the information systems architecture. The interviews findings further support the report IS architecture transformation where ten participants reported SOA's integration into the IS architecture (three of them integrated SOA into the IS architecture without considering the business architecture). Further, the interview findings reflect the dissimilarity of integrated SOA elements into the IS architecture. For example, services and service components [I-11], enterprise service, component service, service description, and SLAs [I-1], and entity and utility services [I-19]. Further, the two cases show contrary results. The information systems architecture was *transformed* in the Dubai Customs case, while it was *reproduced* in the Businesslink case. In Dubai Customs case, the IS architecture accommodated SOA-related elements such as technical services, service operation, and service realisation. The business service of customs declaration was mapped to business processes and then implemented using multiple technical services such as submit declaration, validate declaration and calculate charges. In Businesslink, there was no changes to the IS architectures. The old IS has not been impacted by SOA's introduction.

9.5.3 Technology Architecture

The third level of the architectural elaboration is the transformation or reproduction of the pre-existing technology architecture. Whether it is transformed or reproduced is based on the integration or lack of integration of SOA elements such as technology services, services monitoring, messaging, services security, and an enterprise service bus (ESB) with the technology architecture.

The literature review suggests cases where SOA is integrated into the technology architecture only or in combination with the two levels presented in Sections 9.5.1 and 9.5.2. The interviews findings describe cases of SOA's integration into the technology architecture alone (2 interviews) or with the other architectural levels (10 interviews). One interview reported that SOA was integrated with the business and IS architectures and not with the technology architecture.

The two case studies show contrary results. At Dubai Customs, the technology architecture was transformed, while, at Businesslink, it was reproduced.

9.5.4 EA Governance

The fourth architectural elaboration level, EA governance, was inductively identified from the interviews. EA governance can be transformed by introducing SOA governance into pre-existing EA governance practices. Similar to the other elaboration levels, participants provided mixed opinions about SOA governance integration into EA governance. Only six participants explicitly reported that SOA governance needs to be integrated with EA governance. They had different perspectives on the level of integration required. Some argued that SOA has its own governance and needs to be aligned with EA governance, not necessarily integrated. Others argued that EA governance practices are capable of handling SOA and thus should be integrated.

The case studies also show differences in their governance integration. EA and SOA governance were integrated in Dubai Customs, whereas they were not integrated in Businesslink. Their comparison reveals some interesting insights. For example, in Businesslink, there was some understanding of the need for SOA governance's integration into EA governance. However, that did not occur because of the conditional generative mechanisms' influence (low EA maturity and lack of formally established EA governance practices). In Dubai Customs, there was integration between SOA and EA governance practices, which was enabled by the conditional generative mechanisms (mature EA, well-established EA framework, and comprehensive EA objectives) and their interplay with the action-formation mechanisms of SOA introduction.

Several recent studies depict a similar discussion about SOA governance's relationship to EA governance. Some organisations have leveraged existing EA governance practices to manage SOA activities (Joachim, et al., 2013) while others defined independent SOA governance management (Hojaji & Shirazi, 2010). Clarke, Hall, and Rapanotti (2013) argued that the link between EA governance and SOA governance is not clearly defined in the literature. Thus, such findings are useful and provide empirical perspectives of the link between EA and SOA governance.

9.5.5 EA Methods and Tools

The fifth architectural elaboration level, EA methods and tools, was inductively identified from the interviews. EA methods and tools can be transformed or reproduced depending on the integration or lack of integration of SOA methods and tools. Only five participants explicitly reported such an elaboration level, which provides only weak evidence for this elaboration level in the model. This could be due to the influence of the conditional generative mechanisms. In some cases, EA is comprehensive and includes aspects of solution development and project management, which explains the need to change EA methods and tools after introducing SOA. Whereas, in other cases, EA is a very high level representation of the organisation (strategically oriented), and thus does not include the solution and project management dimensions.

The case study findings also highlight the different outcomes of EA methods and tools elaboration. In Dubai Customs, this level of EA was transformed, while, in Businesslink, it was reproduced. The data analysis of Dubai Customs shows that SOA and EA had overlapping methods and tools. EA was integrated with projects/solutions prior to SOA's introduction, which enabled the transformation of these methods and tools to support SOA. EA deliverables were used to deliver (SOA) project requirements. At Dubai Customs, EA also reviewed SOA projects, monitored their implementation, and ensured they deliver their objectives. Further, requirements, design, and development documents were generated by EA during the design and implementation of projects, including SOA projects. In return, these projects feed back any required architectural changes into EA.

In contrast, the data analysis shows that there was no specific EA method at Businesslink prior to SOA's introduction, and thus there was no such integration. It also shows that there was neither a specific EA tool (repository) nor an EA meta-model, and thus there was no integration. The pre-existing ad-hoc processes and fragmented EA documentation (repositories) were used, and allowed only limited integration of SOA in the project-specific documentation of certain services.

EA and the operational management activities' lack of integration is supported in literature. EA-based solution architecture activities are of substantial business value and considered a key field of relevance for EA management. Yet, there is a small degree of integration between strategic architecture activities and operational ones such as project management and solution development, which are both key aspects of EA management (Simon, et al., 2013). Further, the action-generative mechanisms influence the integration too, such as in the use (or not) of a defined SOA methodology and tools (SOA design). Through a comparative study of SOA methodologies, Gu and Lago (2011) discovered that some SOA methods are developed in isolation from EA, while others aligned with existing EA frameworks.

9.5.6 Summary

There are five elaboration (evolution) outcomes of SOA's integration into EA. The literature review findings suggest that SOA can be integrated into EA on one or more of three levels: business, information systems, and technology architectures. The empirical findings generally support these three levels of elaboration. Two more levels of architectural elaboration were identified from the interview findings: EA governance, and EA methods and tools. The case study phase supports thee five levels being possible EA evolution outcomes. The findings show the significance of the impact of the new emerging business and IT trends on EA frameworks, methodologies, governance, and tools. Thus, it is essential for organisations to explicitly examine whether these emerging trends require EA evolution and, if so, what level/s of EA needs to be evolved?

9.6 Overall Discussion

This section discusses the three analytical phases of EA evolution theoretical model together (see Figure 9.1). The model provides plausible mechanisms that explain EA evolution. It improves understanding of how EA evolves and how EA evolution outcomes can be produced.

First, in Archer's (1995) terms, the architectural conditioning (T1) phase (conditional mechanisms) conditions but does not determine (1), architectural interaction (T2-T3). The findings support Archer's (1995) argument that a structure of interest (here: EA) has properties that allow it to influence (conditional influence) the action that may transform it. In particular, the conditional influence of the architectural conditioning phase was highlighted in Businesslink's case. It showed an example of what Archer calls "an opportunity cost" and agent's ability to overcome the conditional influence. In Businesslink's case, the constraining state of EA at the conditioning phase has increased the cost of pursuing EA-related activities (here: SOA's integration into EA). The conditioning phase had a negative conditional influence on SOA's integration into EA, which caused Businesslink to spend more money and time to improve the pre-existing EA to enable the integration (implementation of enterprise-wide EA by adopting TOGAF).

Three conditional mechanisms (EA framework, EA objectives, and EA maturity) condition but does not determine SOA introduction. They create an enabling or constraining context for EA-related activities. Thus, such conditional influence emphasises that EA development is not a single activity. Rather, it is a process in which previous activities create an enabling or constraining context for the following ones. It is crucial to pay attention to the longitudinal impact of initial EA development, such as the selection of EA framework and the determination of EA objectives and level of EA maturity on further EA activities such as SOA's integration into EA.

Second, the architectural interaction phase is influenced by actors' orientations, beliefs, interests, activities, and resources (action-formation mechanisms) and, in turn, leads to (2), architectural elaboration (T4); that is, to a change in the relations between parts or to no change.

In this thesis, SOA introduction is influenced by six action-formation mechanisms (view of SOA, SOA perceived benefits, SOA scope, SOA governance, SOA design, and business and IT collaboration). They represent the actors' orientations, beliefs, interests, and resources (Archer, 1995; Hedström & Ylikoski, 2010) that influence SOA introduction. Thus, it is crucial to explicitly pinpoint these mechanisms prior to introducing SOA not only to ensure successful SOA implementation but also to understand their influence on SOA's integration into EA.

Third, the results of SOA's introduction (T2-T3) in a given architectural conditioning (T1) results in an architectural elaboration (evolution outcomes) (T4) based on the interplay between action-formation generative mechanisms and the conditional ones. This interplay and the actualisations of these generative mechanisms were dissimilar in the two case studies, which plausibly explain the observed varied evolution outcomes.

The evolution outcomes (architectural elaboration) are classified into five levels. In other words, EA could be transformed (integrated with SOA) or not (reproduced) on one or more of these levels depending on the interplay between the conditional generative mechanisms and the action-formation generative mechanisms in a given context.

Fourth, the model developed in this thesis does not claim that there has to be single (EA evolution) approach. Rather, it emphasises the complexity of the EA evolution process and provides the means to understand EA evolution. By understanding the EA evolution process, it becomes easier to improve its evolution with current and future emerging capabilities such as cloud computing and enterprise mobility. The first phase (architectural conditioning) and the third phase (architectural elaboration) of the developed theoretical model are generic for EA evolution. They are considered applicable for examining EA evolution due to other emerging trends. The second phase (architectural interaction) is also considered applicable and possibly generic to examine emerging trends such as cloud computing. For example, for cloud computing, the view of SOA, SOA scope, and SOA design could be view of cloud, cloud scope, and cloud design.

In a nutshell, continually evolving EA and having consistent terminology and methods are essential for advancing EA as a discipline and practice (Gartner, 2013; MacLennan & Van Belle, 2012; Shah & Golder, 2011; Short, 2013).

9.7 **Summary**

This thesis uses Archer's morphogenetic theory (1995) and its generative mechanisms-based explanation (Archer, 1995; Hedström & Swedberg, 1998; Hedström & Ylikoski, 2010) to develop a theoretical model of EA evolution. The model is used to describe EA evolution and understand its outcomes. By using the morphogenetic theory's three phases, the existence of multiple generative mechanisms (within phase one and two) are proposed to influence EA evolution outcomes (phase three).

The developed model suggests multiple paths of EA evolution. These multiple paths are due to the interplay between the conditional generative mechanisms and the action-formation generative mechanisms. This thesis finds that the three conditional generative mechanisms conditionally influence EA evolution by creating an enabling or a constraining context. Also, the six action-formation generative mechanisms seem to influence the action that happens between T2 and T3 (in this thesis, SOA introduction). Further still, EA may evolve on one or more of the five levels of EA evolution—this is dependent on the interplay between the aforementioned conditional and action-formation mechanisms.

Chapter 10: Conclusion

10.1 Introduction

This thesis develops a theoretical model that describes the EA evolution process (three phases) and explains EA evolution outcomes. It examines SOA introduction as an exemplarily trigger of EA evolution. This thesis views EA evolution as an interaction between existing structural settings (existing EA) and the action of introducing SOA, which results in EA evolution outcomes. The thesis uses Archer's (1995) morphogenetic theory due to the inherent complexity of EA and the theory's consideration of EA evolution's temporality aspect. The developed model is applicable to investigate EA evolution due to other emerging IT capabilities such as cloud computing or enterprise mobility.

This chapter concludes this thesis. Section 10.2 provides an overview of the study. Section 10.3 presents the theoretical contributions, and Section 10.4 focuses on the domain contributions. Section 10.5 lists this thesis's limitations, and Section 10.6 examines future research opportunities.

10.2 Overview of the Study

The development of EA is not a one-off activity that leads to static descriptions of an organisation. Rather, it is a process that parallels the evolution of the organisation and its strategy (Shah & Golder, 2011). EA changes over time to represent the system of interest and provide value for its stakeholders. This thesis distinguishes between two levels of changes related to EA. First, architectural descriptions changes (EA related elements, relationships, viewpoints (Martin, et al., 2009), methods and/or governance changes (empirically identified in this thesis)). Second, representational (content) changes such as changes of applications and processes details. EA needs to evolve in response to emerging business and IT trends, and it is crucial to plan its evolution (MacLennan & Van Belle, 2012; McKendrick, 2010; Shah & Golder, 2011). Many organisations have to confront the challenge of EA evolution. If EA evolution is not planned, EA is likely to evolve in an uncontrolled manner and becomes out-dated as the organisation evolves isolated from its EA (Lucke, et al., 2010; Mens, et al., 2010). Yet, despite the importance of continually evolving EA, few studies have examined EA evolution. Some studies have focused on the representational changes of EA such as changes to applications or processes (e.g. see Buckl, Ernst, Matthes, & Schweda, 2009; Farwick, et al., 2012) and others have provided examples of EA evolution (e.g., SOA's integration into EA) without considering the underlying process of evolution or what may impact EA's evolution (e.g. see Banerjee & Aziz, 2007; Correia & Silva, 2007; Postina, et al., 2010; Shankararaman & Kazmi, 2011; Sharma, 2013).

Furthermore, EA studies often lack robust theoretical foundations (Schmidt & Buxmann, 2011). Thus, this thesis adopts Archer's (1995) morphogenetic theory (to investigate EA evolution). This thesis, concurring with its adopted theory, recognises the inherent complexity and temporal dimension of EA evolution as well as the need for an analytical lens to make sense of it. The morphogenetic theory is used to explore and understand EA evolution. It has an explicit temporal dimension that enabled this thesis to explore EA evolution's temporal aspects (pre-existing EA, action, and EA elaboration). The time dimension is represented by the three analytical phases of the theory: architectural conditioning, architectural interaction, and architectural elaboration. This thesis conception of EA evolution recognises that the generative mechanisms of the first and second phases have an influence on the evolution outcomes at phase three.

10.3 **Theoretical Contributions**

This thesis is the first in-depth explorative study that examines EA evolution in organisations. This is significant due to the critical role of EA in organisations, and the paucity of academic literature addressing EA evolution and explaining how organisations evolve their EA. In doing so, the thesis extends the EA literature by providing significant theoretical insights into how EA evolves and what may impact its evolution outcomes. It particularly adds a useful contribution to the underemphasised EA evolution and thus enriching EA management.

This thesis derives its theoretical contribution via synthesising a comprehensive literature review, empirical data, and the perspective of Archer's (1995) morphogenetic theory. This triangulation led to the main contribution of this thesis: the development of an empirically derived, theoretically driven model that describes EA evolution (the three phases) and provides a plausible explanation of EA evolution outcomes (phase three) (see Figure 10.1).

The developed model is unique for several reasons. First, it the first instantiated work of Archer's (1995) morphogenetic theory in the EA context. Second, it is the first empirical study that sheds light on the EA evolution process and its outcomes. Third, it identifies the components (the three conditional generative mechanisms, the six action-formation generative mechanisms, and the five levels of evolution outcomes) of the model's three phases and uses them to explain EA evolution and its outcomes in depth.

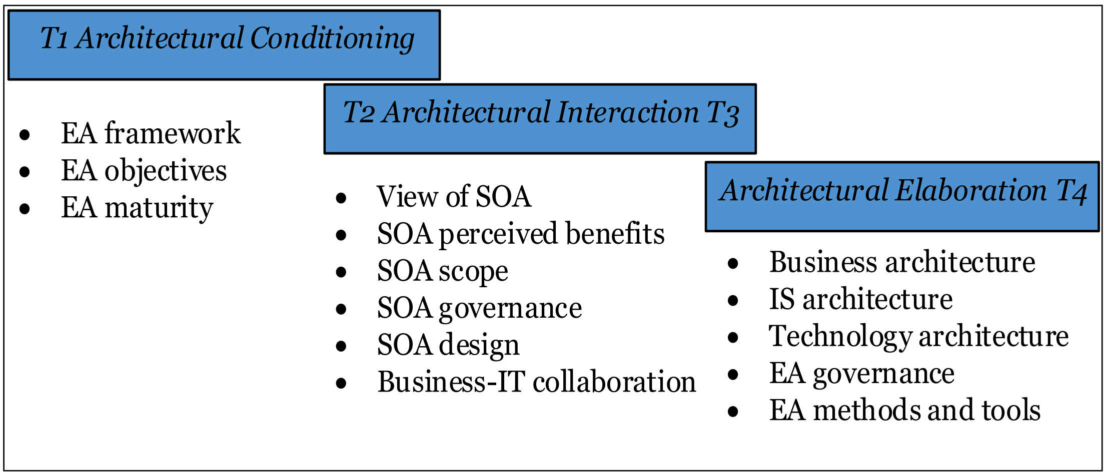


Figure 10.1 EA evolution model

Further, the developed model specifically clarifies SOA's integration into EA as a specific instance of EA evolution and provides guidance on how EA evolves in general due to emergent business and IT capabilities. Following Gregor's (2006) classification of theory, this thesis's theoretical model is considered an analysis and explanation theory. It is an analysis theory in the sense that the three distinctive phases describe the process of EA evolution. It is also reasoned to be an explanation theory because the first two phases of the model have causal mechanisms that provide plausible explanation of the outcomes of EA evolution at phase three. When the three phases are viewed as a combined set, they collectively holistically depict the complex process of EA evolution. The following sections address each phase of the three phases in detail.

10.3.1 Architectural Conditioning (T1)

In the first phase of the model (architectural conditioning), three relevant conditional generative mechanisms (EA framework, EA objectives, and EA maturity) related to EA evolution were identified. These three conditional mechanisms exert causal influence on the EA evolution process in the sense that they create an enabling or a constraining context (contextual conditions) for EA evolution. This phase and its generative mechanisms do not determine EA evolution outcomes, but they condition the action that generates the outcomes.

The thesis found that, when the architectural conditioning (contextual conditions) becomes a constraining factor of EA evolution, there is often an opportunity cost (Archer, 1995) associated with the action to transform EA. In other words, the efforts required to transform EA will increase due to the impact of the constraining contextual conditions. In this situation, more resources and efforts are needed to improve the constraining context to enable EA evolution. Table 10.1 summarises the contributions related to each conditional generative mechanism.

Table 10.1 Architectural conditioning related contributions

Conditional generative mechanism	Implications
EA framework	Having a well-defined, a compressive EA framework and a well-defined meta-model enable EA evolution (Dubai Customs case) and the opposite conditions constrain EA evolution (Businesslink case).
EA objectives	Comprehensive EA objectives should be emphasised and maintained through the participation of business and IT stakeholders, clear EA vision, and the establishment of a common understanding of EA (Dubai Customs) in order to create an enabling context for EA evolution and vice versa (Businesslink).
EA maturity	Mature EA is an enabler of EA evolution (Dubai Customs) and can be a constraint of EA evolution when EA is at a low maturity level (Businesslink). High maturity is achieved through robust governance practices, EA integration with

	demand/project management, business support, and a highly skilled and diverse EA team (Dubai Customs).
Contribution	The three generative mechanisms seem to have a conditional influence on EA evolution. They create either an enabling or constraining context for the action related to EA evolution (in this thesis, SOA's integration into EA). When they create a constrained context, an associated opportunity-cost may be required in order to enable EA evolution.

10.3.2 Architectural Interaction (T2-T3)

In the second phase of the model (the architectural interaction), six relevant action-formation mechanisms related to the action (in this thesis: SOA introduction) were identified. SOA introduction is influenced by agents' orientations, interests, and resources, which are described according to the six generative mechanisms (view of SOA, SOA perceived benefits, SOA scope, SOA design, SOA governance, and business/IT collaboration). These six action-formation mechanisms have a combined influence on SOA introduction and thus the way SOA is integrated into EA. Table 10.2 summarises the contribution of this phase.

Table 10.2 Architectural interaction-related contributions

Action-formation generative mechanism	Implications
View of SOA	There are five diverse views of SOA that influence its introduction. Dubai Customs had a business and IT perspective of SOA, and SOA was implemented on business and IT sides of the organisation. Businesslink had a very business-oriented view and SOA was mainly implemented on the business side of the organisation. SOA implementations involve many categories of actors (e.g., business and IT stakeholders and external vendors in Dubai Customs) with different interests. Thus, a consistent and agreed-on view is a critical factor that should be considered when introducing SOA.
SOA perceived benefits	SOA perceived benefits are classified into strategy, processes, and IT, which influence SOA introduction. A deep understanding of these benefits is critical to understand their implications on SOA introduction as these benefits should be set up in advance to drive SOA implementations and measure the implementation outcomes.

SOA scope	There are three different SOA scoping options that influence SOA introduction: project, portfolio, mostly business-oriented (Businesslink) and organisation level (e.g., Dubai Customs). Each scope has certain objectives, requires different resources, skills, methods, and different organisational impacts.
SOA governance	SOA governance is another generative mechanism that influences SOA introduction. In particular, the service lifecycle management, roles establishment, SOA implementation monitoring, and SOA alignment with existing governance practices influences SOA introduction.
SOA design	SOA design is additional generative mechanism that influences SOA introduction. SOA is designed in different ways using well-defined, loosely defined or un-defined design criteria such as reference architecture, roadmaps, methodology, and services classification frameworks.
Business/IT collaboration	Business and IT collaboration is an additional generative mechanism that influence SOA introduction. The level of business-IT engagement during SOA implementation and the settings and skills of the SOA implementation team (IT, business or mixed) influence how SOA is introduced.
Contribution	The six generative mechanisms seem to have an influence on SOA introduction (the architectural interaction phase). Understanding the implications of these generative mechanisms is essential in implementing and developing SOA. Further, these findings shed light on the often-overlooked organisational and governance aspects of SOA implementations because the majority of the SOA literature concentrates on SOA's technical aspects (Joachim, et al., 2013).

10.3.3 Architectural Elaboration (T4)

According to Archer (1995), the point of examining any morphogenetic cycle (EA evolution) is that it provides an analytical perspective on the emergence of outcomes under investigation. This thesis identified five levels of EA evolution outcomes (business architecture, information systems architecture, technology architecture, EA governance, and EA methods and tools). The interplay between the generative mechanisms of the previous two phases of the theoretical model causes EA to be transformed (in this thesis SOA integration) or reproduced on one or more of these five levels. Table 10.3 summarises the contributions related to this phase.

Table 10.3 Architectural elaboration related contributions

EA evolution outcomes	Implications
Business architecture	EA can evolve on the business architecture level only. The business architecture transformation or reproduction is determined based on the integration of SOA-relevant elements, such as business services, service channels, SOA vision, drivers, SLAs, and QoS in the business architecture.
Information systems architecture	EA can evolve on the information systems architecture level only. IS transformation or reproduction is determined based on the integration or lack of integration of SOA elements, such as application services, service descriptions, and SLAs in information systems architecture.
Technology architecture	EA can evolve on the technology architecture level only. Its transformation or reproduction is determined based on the integration or lack of integration of SOA elements such as technology services, services monitoring, messaging, services security, and an enterprise service nus (ESB) with the technology architecture.
EA governance	EA governance can be transformed after introducing SOA by integrating SOA governance into the pre-existing EA governance practices. The case study findings present different outcomes. Dubai Customs integrated its EA and SOA governance and Businesslink did not.
EA methods and tools	EA methods and tools can be transformed or reproduced, dependent on the integration or lack of integration of SOA methods and tools. The case study findings present different outcomes. Dubai Customs integrated its EA and SOA methods and tools and Businesslink did not.
Contribution	EA can evolve on one or more of the five levels (five outcomes). Such findings explicitly build a first empirical classification of EA evolution rather than a black box perspective (e.g., EA either evolves or does not). These evolution outcomes could be explained by examining the generative mechanisms of the two previous phases in a given context.

In summary, from a meta-theoretical perspective, the thesis is one of the first to adopt a critical realist theory in examining a dynamic aspect of EA. Therefore, the thesis provides a stimulating example of how the morphogenetic theory can be used to obtain the big picture overview of a complex phenomenon and its intrinsic details (in this case, EA evolution). The theory has facilitated the description of EA evolution process along its three analytical phases. Further, it has provided a means to explain the evolution outcomes at phase three, through the examination of the conditional generative mechanisms of the first phase, and the action-formation generative mechanisms in the second phase. This thesis provides a fresh new roadmap to explore EA evolution and to pinpoint the wide array of underlying influences shaping its evolution process.

As to the developed model's generalisability, the first and third phases of the model, architectural conditioning and elaboration, are considered generic to EA evolution. In other words, they are expected to be applicable for studying EA evolution due to other emergent business and IT trends. The second phase (T2 architectural interaction T3) focused on SOA introduction in this thesis. Yet, the action-formation generative mechanisms related to SOA introduction are expected to be relevant when similar IT capabilities are examined in relation to EA evolution. For example, "view of SOA" and "SOA governance" would address the view and the governance of the new IT capabilities that may cause EA to evolve. As such these six mechanisms would be labelled as "perception", "perceived benefits", "scope", "design", "governance" and "business/IT collaboration".

10.4 Domain Contributions

The thesis also made substantial contributions to the domain of EA. Such contributions have significant implications for EA professionals.

The model sheds light on the complexity of EA evolution due to emergent business and IT trends. The study's findings explicitly improve awareness of such a complex process as EA evolution. By identifying the combination of generative mechanisms influencing EA evolution, organisations can be more informed about the prevailing aspects influencing their EA evolution practices. The model provides the basis for the comparative case studies detailed in Chapters 6 and 7 and compared in Chapter 8. In doing so, the model enabled the comparative study of EA evolution in two organisations and provided a consistent basis from which to compare the evolution of EA in these cases. The findings of the two cases provide examples for how EA can evolve differently and enrich the underemphasised EA evolution literature.

Furthermore, through the use of the model as an analytical tool, practitioners are empowered to pinpoint the relevant aspects of EA evolution and identify the weak ones in order to effectively manage EA evolution.

First, identifying the impact of the conditional generative mechanisms of EA evolution has a very important practical contribution. The conditional impact of these mechanisms emphasises that EA development and management are not a single activity but instead a continuous process. The previous activities either facilitate the next ones or make them difficult. This conditional influence is excreted through the three identified conditional generative mechanisms. Thus, to better manage EA evolution, practitioners should consider the impact of these generative mechanisms EA framework, EA objectives, and EA maturity) on EA evolution (e.g., by developing evolution-aware EA frameworks and EA meta-models, or by adopting a flexible EA framework that is malleable enough to accommodate new emerging concepts or trends). Further, by explicitly identifying the wider objectives EA addresses, organisations would be motivated to undertake the initiative to effectively develop their EA initially, and to capitalise the required resources to keep EA evolving. Continuous maturation of an organisation's EA is crucial not only because it enables EA evolution due to growing business and IT changes but also because mature EA is a means to take advantage of the opportunities presented by these emerging capabilities.

Second, identifying the impact of the action-formation generative mechanisms on SOA introduction has remarkable implications for practice. It shows that these mechanisms not only have an impact on the action (SOA's introduction), but on EA evolution as well. Thus, to better manage EA evolution, it is recommended that EA practitioners need to sense potential major business or IT activities that have a potential impact on EA in advance. Such a pro-active approach improves EA evolution management (e.g., Dubai Customs).

Third, the explicit classification of EA evolution outcomes (five levels) helps practitioners identify the relevant aspects of EA that need to be transformed (evolved) in response to a new business/IT capabilities based on their needs and circumstances (e.g., Dubai Customs: evolution on five levels, and Businesslink: evolution on one level).

10.5 Limitations

This thesis, similar to any other research efforts, has limitations. First, this study's scope is limited to identifying the contextual generative mechanisms (factors) related to EA. There are potential contextual generative mechanisms beyond EA such as wider organisational factors or sector-related factors that may impact EA evolution.

Second, this thesis does not thoroughly investigate the inter-relationships between the generative mechanisms at each of the three analytical phases. For example, the inter-relationships between the contextual generative mechanisms of the architectural conditioning phase were not thoroughly examined. Due to the hierarchical nature of mechanisms (Hedström & Ylikoski, 2010), lower-level mechanisms explain higher-level mechanisms. This thesis focused on EA evolution (SOA's integration into EA) through the three phases of the developed model. Thus, the lower-level mechanisms analysis (inter-relationships between them in each phase) is unaddressed.

Third, the limited number of cases constrained the generalisation of the findings. Obviously, examining a complex issue (EA evolution) in only two cases impacts the study's generalisability. Yet, Sayer (2000) argues that one or two cases is enough when using intensive (qualitative) research methods. This thesis intensively examined EA evolution using case study method (two cases, each case involved 8-10 inteerviews, several documents, and online materials). The case studies were enlightend by preceding explorative interviews (20 interviews) and a comprehensive literature review.

Fourth, reliability of the coding may be limited because it was conducted by only one researcher. However, this weakness was mitigated by using measures such as having the coding/quotes critiqued by supervisors.

Fifth, it is recognised that, in some cases, there could be multiple actions that cause EA evolution such as the introduction of SOA, capabilities design, or cloud computing at the same time or in an overlapping way. It would be interesting to investigate such cases to identify the impact of concurrent/overlapping actions on EA evolution.

10.6 Future Research

There is considerable work that needs to be done to advance the potential and utility of the developed model for research in the information systems discipline in general and the EA domain in particular. The developed theoretical model provides a solid theoretical basis and a shared language for future research in the EA evolution domain. Viewing EA evolution through this analytical, combined theoretical model will be valuable for future EA evolution research studies.

Steps in this direction might include applying this model to the examination of EA evolution due to other types of emergent business/IT capabilities. There is potential to examine the developed theoretical model in response to other EA evolution triggers to further develop the model. For instance, a study could identify what further variations are necessary if SOA is replaced by another emerging capability such as cloud computing. Recent research suggests that enterprise architects need to examine their current architectures and to consider the viable means and mechanisms in order to skilfully integrate emerging cloud aspects into their architectures (Raj & Periasamy, 2011) because the characteristics of cloud computing require EA frameworks to be redesigned (Khan & Gangavarapu, 2011).

Moreover, future research, taking into account the findings of this thesis, could develop methods that guide and manage EA evolution using the design science approach as a design theory (Gregor, 2006).

Future studies could also address several limitations of this thesis and remaining open questions. For example, future studies could investigate the interaction of the generative mechanisms at each analytical phase (e.g., at conditioning and interaction phases) to comprehend their internal influence. It would also be interesting to examine whether one mechanism at each phase triggers or influences other mechanisms. Another potential avenue for future research is a concurrent longitudinal case study examining EA evolution as it occurs. Moreover, future studies could investigate the wider contextual factors (conditioning phase) beyond EA that may influence EA evolution such as organisational factors.

In summary, in spite of the importance of understanding how EA evolves (in particular after introducing SOA), there is a paucity of empirical studies that address EA evolution. Thus, this thesis develops a theoretical model that describes EA evolution and explains its outcomes using comprehensive literature review, theoretical lens, twenty explorative interviews, and two intensive retrospective case studies. This thesis made a substantial contribution to the understating and further consideration of EA evolution. By doing so, this thesis complements, using a theory lens, EA dynamic dimensions.

Bibliography

Afshar, M. (2007). Moving beyond project-level SOA: how to achieve departmental and enterprise SOA. Retrieved April 12, 2011 from Oracle Open World, http://www.soasystems.com/news/20071113

Aier, S., & Gleichauf, B. (2009). Towards a sophisticated understanding of service design for enterprise architecture. In *Proceedings of the International Conference on Service-Oriented Computing* (pp. 316-326). Berlin, Germany: Springer.

Aier, S., Gleichauf, B., Saat, J., & Winter, R. (2009). Complexity levels of representing dynamics in EA planning. In A. Albani, J. Barjis & J. G. Dietz (Eds.), *Advances in Enterprise Engineering III* (pp. 55-69): Springer Berlin Heidelberg.

Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2011). Where do we find services in enterprise architectures? a comparative approach. In *Proceedings of the 22nd Australasian Conference on Information Systems (ACIS 2011).*

Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013a). A comparative analysis of the integration of SOA elements in widely-used enterprise architecture frameworks. *International Journal of Intelligent Information Technologies (IJIIT), 9*(2), 54-70.

Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013b). A critical realist perspective of enterprise architecture evolution: preliminary findings. In *Proceedings of the 24nd Australasian Conference on Information Systems (ACIS 2013).*

Alwadain, A., Fielt, E., Korthaus, A., & Rosemann, M. (2013c). Service-oriented architecture integration within enterprise architecture: a-priori model. In *Proceedings of the 24nd Australasian Conference on Information Systems (ACIS 2013).*

Alwadain, A., Korthaus, A., Fielt, E., & Rosemann, M. (2010). Integrating SOA into an enterprise architecture – a comparative analysis of alternative approaches. In *Proceedings of the Fifth IFIP International Conference on Research and Practical Issues of Enterprise Information Systems.*

Antikainen, J., & Pekkola, S. (2009). Factors influencing the alignment of SOA development with business objectives. In *Proceedings of the 17th European Conference on Information Systems*.

Archer, M. (1979). *Social origins of educational systems*. London: Sage Publications.

Archer, M. (2011). Morphogenesis : Realism's explanatory framework. In A. Maccarini, E. Morandi & R. Prandini (Eds.), *Sociological realism. Ontological Explorations* (pp. 59-94). Routledge :Oxford: Taylor & Francis.

Archer, M. S. (1982). Morphogenesis versus structuration: On combining structure and action. *The British Journal of Sociology, 33*(4), 455-483.

Archer, M. S. (1995). *Realist social theory: The morphogenetic approach*. Cambridge, UK: Cambridge university press.

Archer, M. S. (2010). Morphogenesis versus structuration: on combining structure and action1. *The British Journal of Sociology, 61*(s1), 225-252.

Arsanjani, A., Zhang, L., Ellis, M., Allam, A., & Channabasavaiah, K. (2007). Design an SOA solution using a reference architecture. Retrieved March 20, 2010 from http://www.ibm.com/developerworks/library/ar-archtemp/

Assmann, M., & Engels, G. (2008). Transition to service-oriented enterprise architecture. In *Proceedings of the 2nd European conference on Software Architecture*. (pp. 346-349).

Banerjee, J., & Aziz, S. (2007). SOA: the missing link between enterprise architecture and solution architecture. Retrieved April 02, 2010 from http://www.infosys.com/offerings/it-services/soa-services/white-papers/documents/soa-link-ea-sola.pdf

Becker, A., Buxmann, P., & Widjaja, T. (2009). Value potential and challenges of service-oriented architectures-a user and vendor perspective. In *Proceedings of the 17th European Conference on Information Systems (ECIS 2009).*

Becker, W. S., & Burke, M. J. (2012). The staff ride: An approach to qualitative data generation and analysis. *Organizational Research Methods, 15*(2), 316-335.

Beekhuyzen, J., Nielsen, S., & Von Hellens, L. (2010). The NVivo looking glass: seeing the data through the analysis. In *Proceedings of the 5th Conference on Qualitative Research in IT*.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly, 11*(3), 369-386.

Berrisford , G., & Lankhorst, M. (2009). Using ArchiMate with TOGAF. Retrieved October 20, 2010 from http://www.via-nova-architectura.org/artikelen/tijdschrift/using-archimate-with-togaf.html

Bhaskar, R. (1975). *A realist theory of science*. Hassocks, England: Harvester Press.

Bhaskar, R. (1998). *The possibility of naturalism: A philosophical critique of the contemporary human sciences*. London: Taylor & Francis Group.

Boyatzis, R. E. (1998). *Transforming qualitative information : thematic analysis and code development*. Thousand Oaks, CA: Sage Publications.

Bradley, R. V., Pratt, R. M. E., Byrd, T. A., Outlay, C. N., & Wynn, J. D. E. (2012). Enterprise architecture, IT effectiveness and the mediating role of IT alignment in US hospitals. *Information Systems Journal, 22*(2), 97-127.

Braun, C., & Winter, R. (2007). Integration of IT service management into enterprise architecture. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1215-1219).

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101.

Brooks, T. (2009). Principles for implementing a service-oriented enterprise architecture. *SOA Magazine*.

Bucher, T., Fischer, R., Kurpjuweit, S., & Winter, R. (2006). Enterprise architecture analysis and application. An exploratory study. In *Proceedings of The EDOC Workshop on Trends in Enterprise Architecture Research*.

Buckl, S., Ernst, A. M., Matthes, F., Ramacher, R., & Schweda, C. M. (2009). Using enterprise architecture management patterns to complement TOGAF. In *Proceedings of the IEEE International Enterprise Distributed Object Computing Conference* (pp. 34-41).

Buckl, S., Ernst, A. M., Matthes, F., & Schweda, C. M. (2009). Visual roadmaps for managed enterprise architecture evolution. In *Proceedings of the 10th ACIS International Conference onSoftware Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (pp. 352-357).

Buckl, S., Matthes, F., Schulz, C., & Schweda, C. M. (2010). Exemplifying a framework for interrelating enterprise architecture concerns. In M.-A. Sicilia, C. Kop & F. Sartori (Eds.), *Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science* (pp. 33-46). Berlin Heidelberg: Springer

Bui, Q. N. (2012). Making connections: A typological theory on enterprise architecture features and organizational outcomes. In *Proceedings of the 18th Americas Conference on Information Systems*.

Butler , J. (2007). The architecture component of the SAE™ reference framework for SOA. Retrieved April 10, 2010 from http://www.cbdiforum.com/secure/interact/2007-03/the_architecture_component.php

Butti, A. (n.d.). Director general's message: welcome to the Dubai Customs website. Retrieved Dec 10, 2012 from http://www.dubaicustoms.gov.ae/en/AboutDubaiCustoms/Pages/DirectorGeneralsMessage.aspx

Bygstad, B., & Munkvold, B. E. (2011). In search of mechanisms. Conducting a critical realist data analysis. In *Proceedings of the 2011 International Conference on Information Systems*.

Campbell, S., & Mohun, V. (2007). *Mastering enterprise SOA with SAP NetWeaver and mySAP ERP*. Indianapolis, IN: Wiley Pub.

Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in Industry, 59*(7), 647-659.

Chen, H., Kazman, R., & Perry, O. (2010). From software architecture analysis to service engineering: An empirical study of methodology development for enterprise SOA implementation. *IEEE Transactions on Services Computing, 3*(2), 145-160.

Chen, H. M. (2008). Towards service engineering: service orientation and business-IT alignment. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences* (pp. 114-123).

Chiasson, M., Germonprez, M., & Mathiassen, L. (2009). Pluralist action research: a review of the information systems literature. *Information Systems Journal, 19*(1), 31-54.

Chwelos, P., Benbasat, I., & Dexter, A. S. (2001). Research report: Empirical test of an EDI adoption model. *Information systems research, 12*(3), 304-321.

Clarke, M., Hall, J. G., & Rapanotti, L. (2013). Enterprise architecture: A snapshot from practice. *International Journal of IT/Business Alignment and Governance (IJITBAG)*, 1-10.

Correia, A., & Silva, M. (2007). Modeling services in information systems architectures. In P. F. Cunha & P. G. Maropoulos (Eds.), *Digital enterprise technology perspectives and future challenges* (pp. 157-164). New York, USA: Springer.

Creswell, J. W. (2003). *Research design : qualitative, quantitative, and mixed methods approaches* (2nd ed.). London: Sage Publications.

Creswell, J. W. (2009). *Research design : qualitative, quantitative, and mixed methods approaches* (3rd ed.). London: Sage Publications.

Cuellar, M. J. (2010). Using realist social theory to explain project outcomes. *International Journal of Information Technology Project Management 1*(4), 38-52.

Danermark, B., Ekström, M., Jakobsen, L., & Karlsson, J. (2002). *Explaining society : critical realism in the social sciences*. London ; New York: Routledge.

Dico, A. S. (2012). Towards whole-of-government EA with TOGAF and SOA. In *Enterprise Architecture for Connected E-Government: Practices and Innovations* (pp. 177-204): IGI Global.

Dobson, P., Jackson, P., & Gengatharen, D. (2011). Examining rural adoption of broadband–critical realist perspectives. In *Proceedings of the 2011 Pacific Asia Conference on Information Systems (PACIS).*

Dobson, P., Jackson, P., & Gengatharen, D. (2013). Explaining broadband adoption in rural Australia: modes of reflexivity and the morphogenetic approach. *MIS Quarterly, 37*(3), 965-991.

Dobson, P., Myles, J., & Jackson, P. (2007). Making the case for critical realism: Examining the implementation of automated performance management systems. *Information Resources Management Journal (IRMJ), 20*(2), 138-152.

Dubai Customs. (2011). Dubai customs sustainibility report 2011. Retrieved Aug 15, 2012 from https://www.globalreporting.org/Pages/FR-Dubai-Customs-2012.aspx

Dubai Customs. (2012a). About dubai customs. Retrieved Dec 15, 2012 from http://qa.dxbcustoms.gov.ae/en/AboutDubaiCustoms/Pages/default.aspx

Dubai Customs. (2012b). Customs development. Retrieved Aug 12, 2012 from http://www.dubaicustoms.gov.ae/en/AboutDubaiCustoms/OrganizationStructure/Pages/CustomsDevelopment.aspx

Dubai Customs. (2012c). Our vision and mission. Retrieved Aug 12, 2012 from http://www.dxbcustoms.gov.ae/Content/About+Customs/OurVisionMissionAndObjectives/

Easton, G. (2010). Critical realism in case study research. *Industrial Marketing Management, 39*(1), 118-128.

Easton, G., & Harrison, D. (1998). Temporally embedded case comparison in industrial network research. In *Proceedings of the 14th Industrial Marketing and Purchasing Group Conference* (pp. 241-264).

Eisenhardt, K. M. (1989). Building theories from case study research. *The Academy of Management Review, 14*(4), 532-550.

Ekstedt, M., Johnson, P., Lindström, Å., Gammelgård, M., Johansson, E., Plazaola, L., . . . Liliesköld, J. (2004). Consistent enterprise software system architecture for the CIO- A utility-cost based approach. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)* (pp. 5-8).

Elder-Vass, D. (2007). For emergence: refining Archer's account of social structure. *Journal for the Theory of Social Behaviour, 37*(1), 25-44.

Engels, G., & Assmann, M. (2008). Service-oriented enterprise architectures: evolution of concepts and methods. In *Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference.*

Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J. P., . . . Willkomm, J. (2008). A method for engineering a true service-oriented architecture. In *Proceedings of the 10th International Conference on Enterprise Information Systems* (pp. 272-281).

Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. New York: Prentice Hall PTR.

Espinosa, J. A., Boh, W. F., & DeLone, W. (2011). The organizational impact of enterprise architecture: a research framework. In *Proceedings of the 44th Hawaii International Conference on System Sciences* (pp. 1-10).

Everware-CBDI Inc. (2009). CBDI-SAE Meta Model for SOA v3. Retrieved Novomber 20, 2010 from http://everware-cbdi.com/mm-v3

Farwick, M., Schweda, C., Breu, R., Voges, K., & Hanschke, I. (2012). On enterprise architecture change events. In S. Aier, M. Ekstedt, F. Matthes, E. Proper & J. Sanz (Eds.), *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation* (pp. 129-145): Springer Berlin Heidelberg.

Federal CIO Council. (2008). A practical guide to federal service oriented architecture (version 1.1). Retrieved November 10, 2010 from http://www.whitehouse.gov/omb/E-Gov/pgfsoa

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods, 5*(1), 1-11.

Fontana, A., & Frey, J. H. (2000). The interview: from structured questions to negotiated text. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research, 2nd edition,* (pp. 645-672). California: Sage Publications.

Foorthuis, R., Hofman, F., Brinkkemper, S., & Bos, R. (2009). Assessing business and IT projects on compliance with enterprise architecture. In *Proceedings of the Second International Workshop on Governance, Risk and Compliance - Applications in Information Systems*.

Franke, U., Hook, D., Konig, J., Lagerstrom, R., Narman, P., Ullberg, J., . . . Ekstedt, M. (2009). EAF2- A framework for categorizing enterprise architecture frameworks. In *Proceedings of the 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (pp. 327-332).

Gable, G. G. (1994). Integrating case study and survey research methods: an example in information systems. *European Journal of Information Systems, 3*(2), 112-126.

Gartner. (2012a). EA practitioners have significant influence on $1.1 trillion in enterprise it spend. Retrieved Jan 5, 2013 from http://www.gartner.com/id=2286216

Gartner. (2012b). ITScore overview for enterprise architecture. Retrieved November 7, 2012 from http://www.gartner.com/id=1958816

Gartner. (2013). Enterprise architecture program key initiative overview. Retrieved July 10, 2013from http://www.gartner.com/id=2515515

Giddens, A. (1984). *The constitution of society: Outline of the theory of structuration*. Berkeley, CA: University of California Press.

Gimenez, M. E. (2007). For structure: A critique of ontological individualism. *Journal of Critical Realism, 2*(2).

Goethals, F. (2003). An overview of enterprise architecture framework deliverables. Retrieved April 6, 2010from http://econweb.econ.kuleuven.ac.be/fetew/pdf_publicaties/0570.pdf

Gosselt, R. (2012). A maturity model based roadmap for implementing TOGAF. In *Proceedings of the 17th Twente Student Conference on IT*.

Gregor, S. (2006). The nature of theory in information systems,. *MIS Quarterly, 30*(3).

Grigoriu, A. (2007). *SOA, BPM, EA, and service oriented enterprise architecture*: BPTrends.

Gu, Q., & Lago, P. (2011). Guiding the selection of service-oriented software engineering methodologies. *Service Oriented Computing and Applications, 5*(4), 203-223.

Gulledge, T., & Deller, G. (2009). Service-oriented concepts: bridging between managers and technologists. *Industrial Management & Data Systems, 109*(1), 5-15.

Gustas, R. (2007). Semantic integration in conceptual modeling of service architectures. *Transactions on Internet Research, 3*(2), 3-7.

Haki, M. K., Legner, C., & Ahlemann, F. (2012). Beyond EA frameworks: towards an understanding of the adoption of enterprise architecture management. In *Proceedings of the European Conference on Information Systems*.

Handfield, R. B., & Melnyk, S. A. (1998). The scientific theory-building process: a primer using the case of TQM. *Journal of Operations Management, 16*(4), 321-339.

Harmon, K. (2005). The "systems" nature of enterprise architecture. In *Proceedings of the IEEE 2005 International Conference on Systems, Man and Cybernetics* (pp. 78-85).

Harris, P. T. (2008). Enterprise architecture and Its role in solving business issues: Case study of the NSW department of lands. In *Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops*.

Hassanzadeh, A., & Namdarian, L. (2010). Developing a framework for service oriented architecture governance maturity. In *Proceedings of the 5th International Symposium on Telecommunications* (pp. 513-520).

Healy, M., & Perry, C. (2000). Comprehensive criteria to judge validity and reliability of qualitative research within the realism paradigm. *Qualitative Market Research: An International Journal, 3*(3), 118-126.

Hedström, P., & Swedberg, R. (1998). *Social mechanisms : an analytical approach to social theory*. Cambridge New York: Cambridge University Press.

Hedström, P., & Ylikoski, P. (2010). Causal mechanisms in the social sciences. *Annual Review of Sociology, 36*, 49-67.

Henfridsson, O., & Bygstad, B. (2013). The generative mechanisms of digital infrastructure evolution. *MIS Quarterly, 37*(3), 896-931.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-106.

Hirschheim, R., Welke, R., & Schwarz, A. (2010). Service-oriented architecture: myths, realities, and a maturity model. *MIS Quarterly Executive, 9*(1), 37-48.

Hojaji, F., & Shirazi, M. R. A. (2010). Developing a more comprehensive and expressive SOA governance framework. In *Proceedings of the 2nd IEEE International Conference on Information Management and Engineering* (pp. 563-567).

Holland, C., & Light, B. (2001). A stage maturity model for enterprise resource planning systems use. *The DATA BASE for Advances in Information Systems, 32*(2), 34-45.

Iacob, M. E., Jonkers, H., Lankhorst, M. M., & Steen, M. W. A. (2007). Service-oriented enterprise modelling and analysis: A case study. *International Journal of Business Process Integration and Management, 2*(1), 26-37.

Ibrahim, M., & Long, G. (2007). Service-oriented architecture and enterprise architecture, part 2: similarities and differences. Retrieved January 10, 2010 from http://www.ibm.com/developerworks/webservices/library/ws-soa-enterprise2/

IEEE. (2000). *IEEE Std 1471-2000 recommended practice for architectural description of software-Intensive systems*.

IFEAD. (2004). The extended enterprise architecture maturity model (E2AMM). Retrieved March 14, 2011 from http://www.enterprise-architecture.info/Images/E2AF/E2AMMv2.PDF

Infosys. (2009). Enterprise architecture expands its role in strategic business transformation, Infosys enterprise architecture survey 2008/2009. Retrieved February 05, 2010 from http://www.infosys.com/offerings/IT-services/architecture-services/ea-survey/Pages/index.aspx

ISO/IEC/IEEE 42010. (2011). *ISO/IEC/IEEE 42010:2011 systems and software engineering -- architecture description*.

Jamshidi, P., Sharifi, M., & Mansour, S. (2008). To establish enterprise service model from enterprise business model. In *Proceedings of the 2008 IEEE International Conference on Services Computing* (pp. 93-100).

Jensen, J. L., & Rodgers, R. (2001). Cumulating the intellectual gold of case study research. *Public Administration Review, 61*(2), 235-246.

Joachim, N. (2011). A Literature review of research on service-oriented architectures (SOA): characteristics, adoption determinants, governance mechanisms, and business impact. In *Proceedings of the 17th Americas Conference on Information Systems*.

Joachim, N., Beimborn, D., Hoberg, P., & Schlosser, F. (2009). Examining the organizational decision to adopt service-oriented architecture (SOA)-development of a research model. In *Proceedings of the 2009 Diffusion Interest Group In Information Technology*.

Joachim, N., Beimborn, D., Schlosser, F., & Weitzel, T. (2011). Does SOA create or require IT/business collaboration? investigating SOA's potential to reduce the gap between IT and business. In *Proceedings of the 2011 International Conference on Information Systems (ICIS)*.

Joachim, N., Beimborn, D., & Weitzel, T. (2013). The influence of SOA governance mechanisms on IT flexibility and service reuse. *The Journal of Strategic Information Systems, 22*(1), 86-101.

Johnson, P., Ekstedt, M., Silva, E., & Plazaola, L. (2004). Using enterprise architecture for cio decision-making: On the importance of theory. In *Proceedings of the Second Annual Conference on Systems Engineering Research*.

Johnston, R., & Smith , S. (2008). How critical realism clarifies validity issues in theory-testing research. In D. Hart & S. Gregor (Eds.), *Information Systems Foundations: The role of design science*. Canberra, Australia.

Jung, C. (2009). Actionable enterprise architecture. In *Proceedings of the 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (pp. 294-299).

Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in Information Systems research: A case study. *MIS Quarterly, 12*(4), 571-586.

Kappelman, L., McGinnis, T., Pettite, A., & Sidorova, A. (2008). Enterprise architecture: charting the territory for academic research. In *Proceedings of the Fourteenth Americas Conference on Information Systems*.

Karimi, J., Gupta, Y. P., & Somers, T. M. (1996). Impact of competitive strategy and information technology maturity on firms' strategic response to globalization. *Journal of Management Information Systems, 12*(4), 55-88.

Khan, K. M., & Gangavarapu, N. M. (2011). Addressing cloud computing in enterprise architecture: issues and challenges. *Cutter IT Journal, 22*(11), 27-33.

Khoshnevis, S., Aliee, F. S., & Jamshidi, P. (2009). Model driven approach to Service oriented Enterprise Architecture. In *Proceedings of the 2009 IEEE Services Computing Conference* (pp. 279-286).

King, N. (2004). Using interviews in qualitative research. In C. Cassell & G. Symon (Eds.), *Essential Guide to Qualitative Methods in Organizational Research*. London: SAGE Publications Ltd.

Kistasamy, C., van der Merwe, A., & De La Harpe, A. (2010). The relationship between service oriented architecture and enterprise architecture. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference* (pp. 129-137).

Kistasamy, C., Van der Merwe, A., & De La Harpe, A. (2012). The role of service oriented architecture as an enabler for enterprise architecture. In *Proceedings of the 18th Americas Conference on Information Systems*

Knippel, R., & Skytte, B. (2007). Service oriented enterprise architecture: An opportunity of evolution. *Journal of Enterprise Architecture, 3*(3), 17-25.

Kohlmann, F., Börner, R., & Alt, R. (2010). A Framework for the design of service maps. In *Proceedings of the Sixteenth Americas Conference on Information System*.

Kokko, T., Antikainen, J., & Systa, T. (2009). Adopting SOA-experiences from nine finnish organizations. In *Proceedings of the 13th European Conference on Software Maintenance and Reengineering*.

Koumaditis, K., Themistocleous, M., & Da Cunha, P. R. (2013). SOA implementation critical success factors in healthcare. *Journal of Enterprise Information Management, 26*(4), 343-362.

Kumar, S., Dakshinamoorthy, V., & Krishnan, M. S. (2007). Does SOA improve the supply chain? an empirical analysis of the impact of SOA adoption on electronic supply chain performance. In *Proceedings of the 40th Hawaii International Conference on System Sciences*.

Kvale, S. (1996). *Interviews : An introduction to qualitative research interviewing*. Thousand Oaks, Calif: SAGE Publications.

Lagerstorm, R. (2010). *Enterprise systems modifiability analysis* Doctoral Thesis. KTH, Royal Institute of Technology, Stockholm, Sweden

Lagerstrom, R., Sommestad, T., Buschle, M., & Ekstedt, M. (2011). Enterprise architecture management's impact on information technology success. In *Proceedings of the 44th Hawaii International Conference on System Sciences*.

Land, M., Proper, E., Waage, M., Cloo, J., & Steghuis, C. (2009). *Enterprise architecture: creating value by informed governance*. Berlin Heidelberg: Springer-Verlag

Lange, M. (2012). *Evaluating the realization of benefits from enterprise architecture management: construction and validation of a theoretical model*: CreateSpace Independent Publishing Platform.

Lankhorst, M. (2005). *Enterprise architecture at work: Modelling, communication and analysis*. Berlin Heidelberg: Springer-Verlag.

Lankhorst, M. M. (2004). Enterprise architecture modelling--the issue of integration. *Advanced Engineering Informatics, 18*(4), 205-216.

Lapalme, J. (2012). Three schools of thought on enterprise architecture. *IT Professional, 14*(6), 37-43.

Laplante, A., Zhang, J., & Voas, J. (2008). What's in a name? Distinguishing between SaaS and SOA. *IT Professional, 10*(3), 46-50.

Lee, J. H., Shim, H.-J., & Kim, K. K. (2010). Critical success factors in SOA implementation: An exploratory study. *Information Systems Management, 27*(2), 123 - 145.

Legner, C., & Heutschi, R. (2007). SOA adoption in practice - findings from early SOA implementations. In *Proceedings of the 15th European conference on Information Systems* (pp. 1643-1654).

Leist, S., & Zellner, G. (2006). Evaluation of current architecture frameworks. In *Proceedings of the 2006 ACM symposium on Applied computing* (pp. 1546-1553).

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA: Sage Publications, Inc.

Lindström, Å., Johnson, P., Johansson, E., Ekstedt, M., & Simonsson, M. (2006). A survey on CIO concerns-do enterprise architecture frameworks support them? *Information Systems Frontiers, 8*(2), 81-90.

Lucke, C., Krell, S., & Lechner, U. (2010). Critical issues in enterprise architecting–a literature review. In *Proceedings of the 2010 Americas Conference on Information Systems*.

Luo, H. (2006). Building EA management standards:the progresses and challenges of the EA management guide project. Retrieved March 10, 2010 from a|EA Standards Committee, aea-dc.org/eamg/Luo-building-EA-Mgmt-Standards-EA2006-9-11.ppt

Luthria, H., & Rabhi, F. (2009). Service oriented computing in practice - an agenda for research into the factors influencing the organizational adoption of service oriented architectures. *Journal of Theoretical and Applied Electronic Commerce Research, 4*(1), 39-56.

Lynham, S. A. (2000). Theory building in the human resource development profession. *Human Resource Development Quarterly, 11*(2), 159-178.

MacLennan, E., & Van Belle, J.-P. (2012). Factors affecting the organizational adoption of service-oriented architecture (SOA). *Information Systems and e-Business Management*, 1-30.

Martin, R., Purao, S., & Robertson, E. (2009). Evolving enterprise architecture. In *Proceedings of the Next Generation Enterprise Architecture ICEIMT Workshop*.

Maxwell, J. A. (2005). *Qualitative research design: An interactive approach*: Sage Publications, Inc.

Mayntz, R. (2004). Mechanisms in the analysis of social macro-phenomena. *Philosophy of the Social Sciences, 34*(2), 237-259.

McKendrick, J. (2010). Top 10 enterprise architecture trends, identified by Forrester. Retrieved March 02, 2011, from http://www.zdnet.com/blog/service-oriented/top-10-enterprise-architecture-trends-identified-by-forrester/5949

Mens, T., Magee, J., & Rumpe, B. (2010). Evolving software architecture descriptions of critical systems. *Computer, 43*(5), 42–48.

Meyer, M., Helfert, M., & O'Brien, C. (2011). An analysis of enterprise architecture maturity frameworks. In *Perspectives in Business Informatics Research* (pp. 167-177): Springer.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis : an expanded sourcebook* (2nd ed.). Thousand Oaks: Sage.

Mingers, J. (2001). Combining IS Research methods: Towards a pluralist methodology. *Information Systems Research, 12*(3), 240-259.

Mingers, J. (2004). Real-izing information systems: critical realism as an underpinning philosophy for information systems. *Information and Organization, 14*(2), 87-103.

Mingers, J., Mutch, A., & Willcocks, L. (2013). Critical realism in Information Systems research. *MIS Quarterly, 37*(3), 795-802.

Morton, P. (2006). Using critical realism to explain strategic information systems planning. *Journal of Information Technology Theory and Application (JITTA), 8*(1), 3.

Mueller, B., Viering, G., Ahlemann, F., & Riempp, G. (2007). Towards understanding the sources of the economic potential of service-oriented architecture: findings from the automotive and banking industry. In *Proceedings of the Fifteenth European Conference on Information Systems*. (pp. 1608-1619,).

Mutch, A. (2002). Actors and networks or agents and structures: towards a realist view of information systems. *Organization, 9*(3), 477-496.

Mutch, A. (2010). Technology, organization, and structure—A morphogenetic approach. *Organization Science, 21*(2), 507-520.

Mutch, A., Delbridge, R., & Ventresca, M. (2006). Situating organizational action: the relational sociology of organizations. *Organization, 13*(5), 607-625.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization, 17*(1), 2-26.

Mykhashchuk, M., Buckl, S., Dierl, T., & Schweda, C. M. (2011). Charting the landscape of enterprise architecture management. In *Proceedings of the 10th International Conference on Wirtschaftsinformatik*.

NASCIO. (2003). NASCIO enterprise architecture maturity model. Retrieved March 20, 2011 from http://www.nascio.org/publications/documents/NASCIO-EAMM.pdf

Njihia, J. M. (2008). *ICT for development reconsidered: a critical realist approach to the strategic context in Kenya's transition to e-governance*. Doctoral Dissertation, University of Warwick.

Noran, O., & Bernus, P. (2009). Service oriented architecture vs. enterprise architecture: competition or synergy? In R. Meersman, Z. Tari & P. Herrero (Eds.), *Proceedings of the On the Move to Meaningful Internet Systems Workshops* (pp. 304-312).

Nurcan, S., & Schmidt, R. (2009). Service oriented enterprise-architecture for enterprise engineering introduction. In *Proceedings of the 13th Enterprise Distributed Object Computing Conference* (pp. 247-253).

O'Brien, L. (2009). A framework for scope, cost and effort estimation for service oriented architecture (SOA) projects. In *Proceedings of the 2009 Australian Software Engineering Conference* (pp. 101-110).

OASIS. (2006). Reference model for service oriented architecture 1.0. Retrieved January 25, 2010 from http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

Oracle. (n.d.). Dubai Customs' web-based, service-oriented architecture streamlines trade transactions. Retrieved Dec 10, 2012 from http://www.oracle.com/us/corporate/customers/customersearch/dubai-customs-1-soa-snapshot-1500953.html

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: research approaches and assumptions. *Information systems research, 2*(1), 1-28.

Papazoglou, M. P. (2003). Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*.

Paras, G., DeBoever, L., & Westbrock, T. (2007). EA practitioners 'cheatsheet': 12 ideas you can use today to improve your ea team's effectiveness. Retrieved Aug 13, 2010 from http://www.eadirections.com/

Parr, A., & Shanks, G. (2000). A Taxonomy of ERP implementation approaches. In *Proceedings of the 33rd Hawaii International Conference on System Sciences* (pp. 7018-7028).

Patton, M. Q. (1990). *Qualitative evaluation and research methods*. Beverly Hills, CA: SAGE Publications.

Patton, M. Q. (2001). *Qualitative research & evaluation methods*. Thousand Oaks, CA: Sage Publications.

Pawson, R., & Tilley, N. (1997). *Realistic evaluation*. London: Sage Publications.

Perko, J. (2008). *IT governance and enterprise architecture as prerequisites for assimilation of service-oriented architecture. PhD Thesis*. Tampere University of Technology.

Perrey, R., & Lycett, M. (2003). Service-oriented architecture. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops*. (pp. 116-120).

Plummer, D., & McCoy, D. (2006). Achieving agility: defining agility in an it context. Retrieved March 10, 2012 from http://www.gartner.com/id=491393

Postina, M., Trefke, J., & Steffens, U. (2010). An EA-approach to develop soa viewpoints. In *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference* (pp. 37-46).

Pulkkinen, M. (2006). Systemic management of architectural decisions in enterprise architecture planning. four dimensions and three abstraction levels. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*.

Raadt, B., & Vliet, H. (2008). Designing the enterprise architecture function. In S. Becker, F. Plasil & R. Reussner (Eds.), *Quality of Software Architectures. Models and Architectures* (pp. 103-118): Springer Berlin Heidelberg.

Raadt, B., & Vliet, H. (2009). Assessing the efficiency of the enterprise architecture function. In E. Proper, F. Harmsen & J. L. G. Dietz (Eds.), *Advances in Enterprise Engineering II* (pp. 63-83). Berlin: Springer-Verlag

Raduescu, C., & Vessey, I. (2009). Methodology in critical realist research: The mediating role of domain specific theory. In *Proceedings of the 15th Americas Conference on Information Systems* (pp. 1-12).

Raj, P., & Periasamy, M. (2011). The cloud challenges for enterprise architects. In Z. Mahmood & R. Hill (Eds.), *Cloud Computing for Enterprise Architectures* (pp. 187-206): Springer London.

Reich, B. H., & Benbasat, I. (1996). Measuring the linkage between business and information technology objectives. *MIS Quarterly., 20*(1), 55-81.

Ren, M., & Lyytinen, K. J. (2008). Building enterprise architecture agility and sustenance with SOA. *Communications of the Association for Information Systems, 22*(4), 75-86.

Ross, J. (2003). Creating a strategic IT architecture competency: Learning in stages. . *MIS Quarterly Executive, 2*(1), 31-43.

Ross, J. W., & Beath, C. M. (2006). Sustainable IT outsourcing success: Let enterprise architecture be your guide. *MIS Quarterly Executive, 5*(4), 181–192.

Roth, S., Hauder, M., Farwick, M., Breu, R., & Matthes, F. (2013). Enterprise architecture documentation: Current practices and future directions. In *Proceedings of the 11th International Conference on Wirtschaftsinformatik*.

Ryan, A., Tähtinen, J., Vanharanta, M., & Mainela, T. (2012). Putting critical realism to work in the study of business relationship processes. *Industrial Marketing Management, 41*(2), 300-311.

Ryan, G. W., & Bernard, H. R. (2000). Data management and analysis methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research*. London: Sage Publication.

Saat, J., Aier, S., & Gleichauf, B. (2009). Assessing the complexity of dynamics in enterprise architecture planning-Lessons from chaos theory. In *Proceedings of the 15th Americas Conference on Information Systems*.

Salmans, B., Kappelman, L., & Pavur, R. (2009). Organization size, IT capabilities, and EA perceptions: Dark clouds on the ERP horizon? In *Proceedings of the Fifteenth American Conference on Information Systems*.

Sampaio, A. (2010). *An approach for creating and managing enterprise blueprints*. Thesis. Retrieved from https://dspace.ist.utl.pt/bitstream/2295/767600/1/

Sanders, D. T., Hamilton, J. A., & MacDonald, R. A. (2008). Supporting a service-oriented architecture. In *Proceedings of the the 2008 Spring simulation multiconference* (pp. 325-334).

Sayer, A. (1992). *Method in social science : a realist approach* (2nd ed.). London: Routledge.

Sayer, A. (2000). *Realism and social science*. London: Sage Publications Limited.

Scheithauer, G., Augustin, S., & Wirtz, G. (2009). Describing services for service ecosystems. In G. Feuerlicht & W. Lamersdorf (Eds.), *The 2008 Service-Oriented Computing Workshops* (pp. 242-255): Springer Berlin / Heidelberg.

Schekkerman, J. (2004). *How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework*. British Columbia: Canada: Trafford Publishing.

Schekkerman, J. (2005). *The economic benefits of enterprise architecture*. Victoria, BC, Canada: Trafford Publishing.

Schelp, J., & Aier, S. (2009). SOA and EA—sustainable contributions for increasing corporate agility. In *Proceedings of the 42nd Hawaii International Conference on System Sciences* (pp. 1-8).

Schmelzer, R. (2006). SOA and the Zachman framework. Retrieved January 28, 2010 from http://www.zapthink.com/2006/06/15/soa-and-the-zachman-framework/

Schmidt, C., & Buxmann, P. (2011). Outcomes and success factors of enterprise IT architecture management: empirical insight from the international financial services industry. *European Journal of Information Systems, 20*(2), 168-185.

Schroth, C. (2007). The Service-oriented enterprise. *Journal of Enterprise Architecture, 3*(4), 73-80.

Schulte, S., Kadner, K., Repp, N., & Steinmetz, R. (2009). Applied service engineering for single services and corresponding service landscapes. In *Proceedings of the15th Americas Conference on Information Systems (AMCIS).*

Seppänen, V. (2008). Interconnections and differences between EA and SOA in government ICT development. In *Proceedings of the 31st Information Systems Research Seminar in Scandinavia*.

Shah, A., & Kalin, P. (2007). SOA adoption models. Retrieved March 15, 2011 from http://soa.sys-con.com/node/393684?page=0,1

Shah, H., & Golder, P. (2011). ADaPPT: Enterprise Architecture Thinking for Information Systems Development. *International Journal of Computer Science Issues, 8*(1).

Shan, T. C., & Hua, W. W. (2006). Service-oriented solution framework for internet banking. *International Journal of Web Services Research, 3*(1), 29-29-48.

Shankararaman, V., & Kazmi, P. (2011, 5-7 Sept. 2011). Unifying EA, BPM and SOA through a synergestic framework. In *Proceedings of the IEEE 13th Conference on Commerce and Enterprise Computing* (pp. 286-293).

Sharma, S. (2013). Using enterprise architecture with SOA helps in achieving business-IT alignment. Retrieved June 15th, 2013 from http://ovum.com/2013/02/19/using-enterprise-architecture-with-soa-helps-in-achieving-business-it-alignment/

Short, J. (2013). Agenda overview for enterprise architecture, 2013. Retrieved April 10, 2013 from www.gartner.com/id=2292615

Simon, D., Fischbach, K., & Schoder, D. (2013). An exploration of enterprise architecture research. *Communications of the Association for Information Systems, 32*(1).

Smith, M. L. (2006). Overcoming theory-practice inconsistencies: Critical realism and information systems research. *Information and Organization, 16*(3), 191-211.

Sousa, P., Lima, J., Sampaio, A., & Pereira, C. (2009). An approach for creating and managing enterprise blueprints: A case for it blueprints. In A. Albani, J. Barjis & J. L. G. Dietz (Eds.), *Advances in Enterprise Engineering III* (pp. 70-84). Berlin Heidelberg: Springer.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal, 31*(3), 590-616.

Steen, M. W. A., Strating, P., Lankhorst, M. M., ter Doest, H., & Iacob, M. E. (2005). Service-oriented enterprise architecture. In Z. Stojanovic & A. Dahanayake (Eds.), *Service-Oriented Software System Engineering: Challenges and Practices*. (pp. 132-154). London, UK: Idea Group Inc.

Steenbergen, M., Berg, M., & Brinkkemper, S. (2007). An instrument for the development of the enterprise architecture practice. In *Proceedings of the 9th International Conference on Enterprise Information Systems*.

Stein, S., Lauer, J., & Ivanov, K. (2008). ARIS method extension for business-driven SOA. *WIRTSCHAFTSINFORMATIK, 50*(6), 436-444.

Street, C., & Ward, K. (2010). Retrospective case study. In G. D. A. Mills, & E. Wiebe (Ed.), *Encyclopedia of case study research* (pp. 825-828): Thousand Oaks, CA: SAGE Publications.

Sweeney, R. (2010). *Achieving service-oriented architecture: applying an enterprise architecture approach*. New Jersey, USA: Wiley & Sons Inc.

Tamm, T., Seddon, P. B., Shanks, G., & Reynolds, P. (2011). How does enterprise architecture add value to organisations? *Communications of the Association for Information Systems, 28*(1), 10.

Tansley, C., & Watson, T. (2000). Strategic exchange in the development of human resource information systems (HRIS). *New Technology, Work and Employment, 15*(2), 108-122.

Tellis, W. (1997). Introduction to case study. *The Qualitative Report, 3*(2).

The Open Group. (2009a). *ArchiMate 1.0 Specification*. Zaltbommel, Netherlands.: Van Haren Publishing.

The Open Group. (2009b). Architectural artifacts. Retrieved October 15, 2010 from http://www.opengroup.org/architecture/togaf9-doc/arch/index.html

The Open Group. (2009c). Architecture Maturity Models. Retrieved 21 March, 2011, from http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap27.html

The Open Group. (2009d). Content metamodel Retrieved October 15, 2010 from http://www.opengroup.org/architecture/togaf9-doc/arch/chap34.html

The Open Group. (2009e). *The Open Group architecture framework TOGAF*. LJ Zaltbommel, the Netherlands: Van Haren Publishing.

The Open Group. (2009f). SOA reference architecture (draft 10). Retrieved March 10, 2010 from https://www.opengroup.org/projects/soa-ref-arch/uploads/40/19713/soa-ra-public-050609.pdf

The Open Group. (2010). Definition of SOA Retrieved January 20, 2010 from http://www.opengroup.org/projects/soa/doc.tpl?gdid=10632

The Open Group. (2012a). ArchiMate® 2.1 Specification. Retrieved 15th April, 2014from http://pubs.opengroup.org/architecture/archimate2-doc/chap02.html#_Toc371945150

The Open Group. (2012b). Definitions. Retrieved March 21, 2010 from http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html

The UK Department of Defence. (2010a). MOD architecture framework (MODAF). Retrieved October 22, 2010 from http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/ModafDetailedGuidance.htm

The UK Department of Defence. (2010b). The MODAF service oriented viewpoint. Retrieved October 24, 2010 from http://www.mod.uk/DefenceInternet/AboutDefence/CorporatePublications/InformationManagement/MODAF/TheServiceOrientedViewpointsov.htm

The US GAO. (2010). A framework for assessing and improving enterprise architecture management (version 2.0). Retrieved October 24, 2010 from http://www.gao.gov/new.items/d10846g.pdf

The US OMB. (2009). OMB EA assessment framework 3.1. Retrieved March 10, 2011 from http://www.whitehouse.gov/omb/e-gov/eaaf

Traverson, B. (2008). A comparison study of viewpoint approaches in service enterprise architecture. . In *Proceedings of the 12th Enterprise Distributed Object Computing Conference* (pp. 444-451).

Trosper, R. L. (2005). Emergence unites ecology and society. *Ecology and Society, 10*(1), 1-14.

Tsang, E. W. K. (2013). Case studies and generalization in information systems research: A critical realist perspective. *The Journal of Strategic Information Systems*(In Press).

Turner, P., Gøtze, J., & Bernus, P. (2010). Architecting the firm – coherency and consistency in managing the enterprise. In P. Bernus, G.

Doumeingts & M. Fox (Eds.), *Enterprise Architecture, Integration and Interoperability* (Vol. 326, pp. 1-10): Springer Boston.

Urbaczewski, L., & Mrdalj, S. (2006). A comparison of enterprise architecture frameworks. *Issues in Information Systems, 7*(2), 18-23.

US Department of Defence. (2009). DoD architecture framework version 2.0. Retrieved October 20, 2010 from http://cio-nii.defense.gov/sites/dodaf20/

van der Raadt, B., Hoorn, J. F., & van Vliet, H. (2005). Alignment and maturity are siblings in architecture assessment. In O. Pastor & J. Cunha (Eds.), *Advanced Information Systems Engineering* (pp. 357-371).

Varnus, J., & Panaich, N. (2009). TOGAF 9 survey results. Retrieved January 29, 2010 from https://www.opengroup.org/conference-live/uploads/40/20081/Mon - am - 3 - Varnus.pdf

Venkatesh, V., Bala, H., Venkatraman, S., & Bates, J. (2007). Enterprise architecture maturity: The story of the veterans health administration. *MIS Quarterly Executive, 6*(2), 79-90.

Viering, G., Legner, C., & Ahlemann, F. (2009). The (lacking) business perspective on SOA–critical themes in SOA research. In *Wirtschaftsinformatik* (pp. 45-54).

Volkoff, O., & Strong, D. M. (2013). Critical realism and affordances: Theorizing it-associated organizational change processes. *MIS Quarterly, 37*(1), 819-834.

Volkoff, O., Strong, D. M., & Elmes, M. B. (2007). Technological embeddedness and organizational change. *Organization Science, 18*(5), 832-848.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly, 26*(2), 13-23.

Weerakkody, V., Janssen, M., & Hjort-Madsen, K. (2007). Integration and enterprise architecture challenges in e-government: A European perspective. *International Journal of Cases on Electronic Commerce, 3*(2), 13-35.

Welke, R., Hirschheim, R., & Schwarz, A. (2011). Service-oriented architecture maturity. *IEEE Computer, 44*(2), 61-67.

Winter, K., Buckl, S., Matthes, F., & Schweda, C. M. (2010). Investigating the state-of-the-art in enterprise architecture management methods in literature and practice. In *Proceedings of the Mediterranean Conference on Information Systems*.

Winter, R., & Fischer, R. (2007). Essential layers, artifacts, and dependencies of enterprise architecture. *Journal of Enterprise Architecture, 3*(2), 7–18.

Wong, C. (2005). A critical realist approach to organizational innovation research. In *Proceedings of the 4th International Critical Management Studies Conference*.

Wynn, D., & Williams, C. K. (2012). Principles for conducting critical realist case study research in information systems. *Management Information Systems Quarterly, 36*(3), 787-810.

Wynn Jr, D. E., & Williams, C. K. (2008). *Critical realm-based explanatory case study research in information systems.* . Paper presented at The International Conference on Information Systems (ICIS).

Yin, R. (2003). *Case study research: Design and methods*: Thousand Oaks, California: Sage Publications, Inc.

Yin, R. (2009). *Case study research: Design and methods*. Thousand Oaks, California: Sage Publications, Inc.

Yoon, T., & Carter, P. (2007). Investigating the antecedents and benefits of SOA implementation: a multi-case study approach. In *Proceedings of the 13th Americas Conference on Information Systems* (pp. 1-11).

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 454-470.

Zawya. (2011). Dubai Customs and Oracle sign agreement. Retrieved Nov 5, 2012 from http://www.zawya.com/story/ZAWYA20110120140238/

Appendix A: Interview Protocol

Introduction

The objective of this study is to understand how SOA is integrated into enterprise architecture, how SOA elements (artefacts) are represented in the enterprise architecture, and why is SOA represented in such way. It also aims to understand how SOA elements alter and interact with the original elements of the enterprise architecture.

This is the standard interview protocol that will be used during the interview sessions. A semi-structured interview is used, and the following topics and questions could therefore be seen as guidelines for the interview. Participants will be asked to be audio recorded. The interview was updated after the preliminary analysis of the first 3 interviews to include points/issues mentioned by participants.

Setting Up

- Introduce the interviewer:
- Introduce the subject of the study
- Explain the confidential agreement
- Sign a written consent form (enclosed) to confirm agreement to participate.

Interviews Questions Divided Into Three Parts

The interview is divided into four parts covering general information, EA, SOA, and their integration. The main questions are presented in the main bullet points. The sub-bullet points are to elaborate on the point if the interviewee doesn't mention it.

Retrieve insights about the organisation, the interviewee

- What is your primary job title?
- How long have you been doing this job?
- What is the industry of the organisation?
- How many people work in the organisation?

Retrieve insights about EA prior and during SOA introduction

- What is the definition of EA?
- What was the used EA framework?
    - EA structure/layers?
    - EA methods?
- When did you start your EA program?

- What were the objectives of adopting EA?
- Who were involved in EA efforts?
- Where was/is EA team located in the organisational structure?
    - How many architects? Structure of the team? SOA team?
- How mature were your EA practices?
    - Documentation, planning, governance, team and resources, business support and EA evaluation
- How often EA and its artefacts are used and by whom?
- Do you use EA models for projects (as input to projects)?

Retrieve insights about SOA practices

- When did the organisation start the SOA?
- Why did the organisation start the SOA?
- What was the view/perception of SOA?
- What were the perceived benefits of SOA?
- What was the scope of SOA adoption?
- What methodology was used for SOA implementation?
- Did you have SOA governance strategy/framework?
- Was your SOA implementation supported by the business?
- Are there any other important aspects related to SOA implementation?

Retrieve insights about SOA and EA integration

- How SOA is integrated within EA?
- How did SOA align with or affect your EA?
    - What were the sub-architectures that are affected?
    - How were they affected?
- What SOA's elements/artefacts did you represent in EA?
    - examples, documents or models that illustrate the integration?

Main contacts at Queensland University of Technology

Ayed Alwadain PhD student Queensland University of Technology T: +61 (0)7 3138 9476 M: +61 (0)423 561759 E a.alwadain@student.qut.edu.au	**Prof. Michael Rosemann** Head of Information Systems Discipline Faculty of Science and Technology Queensland University of Technology P +61 (0)7 3138 9473 M +61 (0)408 735363 E m.rosemann@qut.edu.au

Appendix B: Case Study Protocol

Introduction

The objective of this study is to understand how SOA is integrated into enterprise architecture, how SOA elements (artefacts) are represented in the enterprise architecture, and why is SOA represented in such way. It also aims to understand how SOA elements alter and interact with the original elements of the enterprise architecture.

This is the standard case study protocol that will be used during the case studies. It includes information about the ideal candidates for the interviews, relevant (required) EA and SOA relevant documentations, and a semi-structured interview protocol and

Candidate Participants

- Access to senior executives who have initiated and managed EA and service-orientation project (2-3 CxO executives, one hour interview each).
- Access to managers who were involved in EA and service-orientation (e.g., program manager, project manager, chief enterprise architects, etc) (3-4 participants, one hour interview each).
- Access to architects who were involved in EA and service-orientation (e.g., business, application, solution, infrastructure architects, etc) (2-3 participants from each domain one hour interview each).
- Access to EA and service-orientation related strategies, objectives, roadmaps, meta-models, and methodologies.
- Access to project documentation such as program charter, project plan, project presentations, models, meta-models, memos, and progress reports for analysis.
- If possible, attendance at meetings and workshops related to a current EA and service-orientation project.

Data Collection Procedures

In this study, multiple procedures for data collections are employed such as interviews, observations, documents analysis, and archival analysis.

Publicly available documents, presentation, and the organisation's website are the starting point for the case study. Access to documents related to EA planning, EA framework, EA models, EA meta-model, EA governance, SOA design, SOA reference architecture, SOA governance, SOA roadmap, and SOA planning will be requested.

Interviews

The company's business and technical members who have been involved in the SOA integration into EA, specifically the project leading team, will be approached to be interviewed. Such detailed interviews can provide a close perspective on the actual SOA and EA integration practices and experiences to enrich our understanding and help refining the models built in the previous qualitative interviews phase.

Interviews will target EA managers, business architects, application architects, technical architects, SOA architects, and EA and SOA governance bodies.

Documents

- EA strategies, framework, plans, methods, artefacts, tools, governance models, deliverables, meta-models, KPIs, and so on
- SOA strategies, plans, methods, service models, tools, artefacts, governance models, KPIs, and so on
- SOA and EA related reports, workshops, teams' structure, presentations, deliverables, and so on

Semi-structured Interviews Questions

The interview protocol is an extended version of the interview phase.

Retrieve Insights about the Interviewee

- What is your primary job title?
- How long have you been doing this job?

EA-related Questions Prior to SOA Introduction

- When did you start your EA program?
- Where is/was EA team located in the organisational structure?
    - How many architects?
    - What are their roles?

Present EA Maturity assessment to the interviewee to choose the best statements to describe their EA maturity level prior to SOA implementation (Page 6-9).

- How would you define EA?
- What was the view of EA in the organisation?
- What were the drivers/objectives of it?
- What was the used EA methodology and architectural viewpoints?
- What was the used EA framework?
- Were EA (as-is) models important for SOA?
    - How were they used?
    - What models/viewpoints were relevant for SOA?

- Can you please describe your EA planning process?
- Did you have an EA governance body?
- Did you have well-defined EA financial and staffing resource requirements?
- Do you evaluate/assess your EA models, framework, meta-model?
- Who were involved in EA program?

SOA Related Questions

- When did the organisation start the SOA initiative?
- Where is SOA team located?
    - How many people were in SOA team?
    - What were their roles?

View of SOA

- What was the view of SOA?
    - How would you define it?

SOA scope

- What was the scope of your SOA program?
- Why did choose such scope?

SOA perceived benefits

- Why did you adopt SOA?

SOA design

- How was your SOA designed?
    - SOA reference architecture? SOA roadmap/strategy? Service repository (catalogue)? service classification model? types of services that you have? SOA methodology?

SOA governance

- Can you please describe your SOA governance?
    - How is it linked to other governance practices, e.g. EA?

Business and IT collaboration

- What is the level of the collaboration between business and IT in terms of SOA introduction?
    - How do they collaborate?
    - Is business support important for SOA? how?
    - What were/are the required skills for SOA program?

EA and SOA Integration Aspects

The table below describes EA and SOA integration aspects.

Business architecture	SOA is integrated within business architecture. It accommodates related SOA elements business services, service description, service channels, SOA vision, drivers, service actors, SLAs and SOA vision.
IS architecture	SOA is integrated within information systems architecture. It accommodates relevant SOA elements such as application services, service descriptions, SLAs.
Tech architecture	SOA is integrated within technology architecture. It accommodates SOA elements such as technology services, service interfaces, messages, services monitoring elements, services security elements and physical technology Components (SOA infrastructure; e.g., repository, enterprise service bus (ESB), BPEL executors and registry).
EA governance integration	SOA governance is integrated with EA governance standards, committees and practices.
EA methods and tools integration	SOA methods and tools are integrated with EA methods and tools.

Business architecture

- Is your business architecture service-oriented?
    - How?
- What are SOA/services elements that are represented in the business architecture?
    - E,g., business services, Service channels, contract, service consumers, providers, SOA vision, drivers, , SLA, QoS, and so on
    - Describe them please?
- What is the relationship between SOA elements and other business architecture elements?

Information systems architecture

- Is your Information Systems (Application/Data) architecture service-oriented?
    - How?
- What are SOA/services elements that are represented in the IS (Application/data) architecture?
    a. E,g. application services, data services, contract, consumer..ect
- How these elements are associated / integrated with other EA elements?
- Do you have a meta-model that explains such integration?

Technology architecture

- Is your technology architecture service-oriented?
    - How?
- What are SOA/services elements that are represented in the technology architecture?
    - E,g. technical services, web services, SLA, QoS, ESB?
    - How these elements are associated / integrated with other EA elements?
- Do you have a meta-model describes such elements/ integration?

EA governance

- What changes SOA brings to your existing EA governance?
- How are EA and SOA governance integrated?
    - What is the overlap between them?

  - How are their frameworks, bodies, standards, policies, and lifecycles integrated?
- Do you have documents that describe such integration/alignment?

EA methods and tools

- What changes SOA brings to your EA design and development methods/processes and tools?
- How is SOA integrated with your existing ....
  - design and development methods/processes
  - Guidelines
  - Solution/Project management

EA Maturity Survey

Statements were adapted from (Lagerstrom, et al., 2011; NASCIO, 2003; The Open Group, 2009c).

EA objectives

- **Which one of the following statements best describe your EA vision?**

Strategic EA	In our organisation, EA is used for strategic alignment In our organisation, EA is used for business-IT alignment
Operational EA	EA is used for operational activities EA current state (as-is) documentation is the main focus EA is used for just in time problem solving (no long term planning)
Governance oriented	EA is a legalisation body. EA sets procedures, guidelines to govern architectural practices.
IT oriented	EA covers only the IT domain (IT architecture)

EA benefits

- **What are the realized benefits of EA?**

EA benefits	Strategy Execution
	B-IT Alignment
	Communication
	Enterprise Integration
	Decision Making
	Governance
	Managing Change
	Accountability
	Reduce complexity
	Standardisation
	Others..

EA documentation

- **Which number below best describes your EA documentation practices?**

0	• Architecture processes, artefacts and templates are not documented
1	• Documentation processes are ad hoc and informal
2	• The need for an EA repository for storage and dissemination of the captured EA information has been identified • The organisation is beginning to reuse methods for capturing critical EA information
3	• Templates are used to ensure the capturing of information is consistent • Documentation of business and IT information is consistent
4	• Documentation has become a standard practice • The organisation captures metrics to identify the need for updates to blueprint information
5	• Captured business and technology information is reviewed in conjunction with the monitoring of new technology and business trends

EA planning

- **Which number below represents your EA planning practices?**

0	• No plans for developing Enterprise Architecture are in place
1	• EA activities are informal and unstructured
2	• Organisation has begun to identify EA tasks and resources requirements. • Organisation has decided on a methodology and begun to develop a plan for their EA
3	• EA plans are well-defined, including a structured framework and timeline for developing the EA • EA activities are carried out according to the defined plan
4	• EA plans are reviewed and changes are incorporated to improve the EA Program • The organisation captures metrics to measure the progress against the established EA plans • Goals are being set for the future of the EA Program Plan
5	• Action plans are proactively implemented to increase the effectiveness of the EA Program based on captured metrics.

EA governance

- **Which number below describes your architecture governance?**

0	• No explicit governance of EA.
1	• The need for committees to define the architectural standards and processes has been identified
2	• EA Program has begun to develop clear roles and responsibilities • Governance committees are starting to form
3	• Architecture Governance committees are defined, and have defined roles and responsibilities • Authority of the governance committees is aligned to work together smoothly
4	• Governance roles and responsibilities are reviewed and updated to incorporate changes to the EA Framework • Formal processes for managing variances feed back into architecture.

5	• Governance committees proactively review their activities to improve their processes • Explicit governance of all investments. • A standards and waivers process is used to make governance process improvements.

EA team (committee)

- **Which number below describes your EA team (committee) practices?**

0	• There is no EA team
1	• The organisation has identified a need for capable EA team • EA team efforts are informal and inconsistent
2	• EA Program has begun to develop clear roles and responsibilities • EA team has begun to develop plans for EA educational sessions to increase the awareness of the EA
3	• EA team includes business and IT staff. • Training is provided for members of the EA team
4	• EA awareness training is incorporated into new employee orientation • The organisation captures metrics to measure the effectiveness of the EA team
5	• Metrics are used to proactively identify opportunities for improved EA team and resources • The organisation works with others to share ideas for improvements of EA team and resources

EA business support

- **Which number below describes the business support for your EA?**

0	• No business Support of EA
1	• Senior Management understands the need for EA
2	• The need to create greater awareness of EA has been identified • EA awareness activities are beginning to emerge or be developed
3	• Business and IT stakeholders have a good understanding of the architecture principals and participate in EA processes
4	• Senior management directly involved in the architecture processes. • Senior Management participate in various EA committees
5	• Business and IT work together as contributors to the architecture and its processes • The organisation creates an atmosphere for active involvement and participation in EA Program and activities across the organisation

EA evaluation

- **Which number below describes your EA evaluation and maintenance practices?**

0	• There are no evaluation procedures for EA framework and outcomes
1	• Evaluation processes are ad-hoc and informal
2	• The organisation begins to develop systematic architectural evaluation procedures
3	• There exist defined evaluation processes for EA framework and outcomes
4	• EA framework and outcomes are regularly evaluated • Meetings are held regularly to review modifications to the EA framework and

	outcomes
5	• Feedback on architecture processes and outcomes is used to drive architecture process improvements. • Corrective action plans are put in place when deficiencies in templates and/or procedures are identified

The statements abovementioned *are grouped according to five maturity stages* in the following Table

Level 0 : no program
• Architecture processes, artefacts and templates are not documented • No plans for developing Enterprise Architecture are in place • No explicit governance of EA. • There is no EA team • No business Support of EA
Level 1: informal program
• Documentation processes are ad hoc and informal • EA activities are informal and unstructured • The need for committees to define the architectural standards and processes has been identified • The organisation has identified a need for capable EA team • Senior Management understands the need for EA • Evaluation processes are ad-hoc and informal
Level 2: repeatable program
• The organisation is beginning to reuse methods for capturing critical EA information • The need for an EA repository for storage and dissemination of the captured EA information has been identified • Organisation has begun to identify EA tasks and resources requirements. • Organisation has decided on a methodology and begun to develop a plan for their EA • A need for architecture governance has been identified • EA program has begun to develop clear roles and responsibilities • Governance committees are starting to form • EA team has begun to develop plans for EA educational sessions to increase the awareness of the EA • The need to create greater awareness of EA has been identified • EA awareness activities are beginning to emerge or be developed • The organisation begins to develop systematic architectural evaluation procedures
Level 3: well-defined program
• Templates are used to ensure the capturing of information is consistent • Documentation of business and IT information is consistent • EA plans are well-defined, including a structured framework and timeline for developing the EA • EA activities are carried out according to the defined plan • Architecture Governance committees are defined, and have defined roles and responsibilities • Authority of the governance committees is aligned to work together smoothly • EA team includes business and IT staff. • Training is provided for members of the EA team • Business and IT stakeholders have a good understanding of the architecture principals and participate in EA processes • There exist defined evaluation processes for EA framework and outcomes
Level 4: managed program
• Documentation has become a standard practice • The organisation captures metrics to identify the need for updates to blueprint information • EA plans are reviewed and changes are incorporated to improve the EA Program

• The organisation captures metrics to measure the progress against the established EA plans • Goals are being set for the future of the EA program plan • Governance roles and responsibilities are reviewed and updated to incorporate changes to the EA framework • Formal processes for managing variances feed back into architecture • EA awareness training is incorporated into new employee orientation • The organisation captures metrics to measure the effectiveness of the EA team • Senior management directly involved in the architecture processes. • Senior Management participate in various EA committees • EA framework and outcomes are regularly evaluated • Meetings are held regularly to review modifications to the EA framework and outcomes
Level 5: continuously improving vital program
• Captured business and technology information is reviewed in conjunction with the monitoring of new technology and business trends • Action plans are proactively implemented to increase the effectiveness of the EA Program based on captured metrics • Governance committees proactively review their activities to improve their processes • Explicit governance of all investments. • A standards and waivers process is used to make governance process improvements. • Metrics are used to proactively identify opportunities for improved EA team and resources • The organisation works with others to share ideas for improvements of EA team and resources • Business and IT work together as contributors to the architecture and its processes • The organisation creates an atmosphere for active involvement and participation in EA Program and activities across the organisation • Feedback on architecture processes and outcomes is used to drive architecture process improvements. • Corrective action plans are put in place when deficiencies in templates and/or procedures are identified

Field Notes Templates

Field notes templates are design for reflection purposes during the interviews and after the interviews. The employed templates are

1. Contact summary form
2. Document summary form

Contact Summary Form

Contact (Visit, phone, email) Site: ________________

Details of contact person: Date: ________________

Name: ____________________________

Position: ____________________________

Phone: ____________________________

Email: ____________________________

	questions	Reflection notes
1	Summarise the information that you obtained (or failed to obtain) for each relevant question	
2	Are there any emerging ideas important for the study?	
3	What are the new (or remaining) target questions for next contact?	

Document Summary Form

Document Name: ____________________ Date: ____________________

Accessed: __________________________

	notes
Description of the document	
Importance of the document for the study	
Additional comments	